Enabling Peer to Peer Energy Trading Marketplace

Using Consortium Blockchain Networks

by

Raj Anil Sadaye

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2019 by the
Graduate Supervisory Committee:

Kasim Selcuk Candan, Chair
Dragan Boscovic
Ming Zhao

ARIZONA STATE UNIVERSITY

December 2019

ABSTRACT

Blockchain technology enables peer-to-peer transactions through the elimination of the need for a centralized entity governing consensus. Rather than having a centralized database, the data is distributed across multiple computers which enables crash fault tolerance as well as makes the system difficult to tamper-with due to a distributed consensus algorithm.

In this research, the potential of blockchain technology to manage energy transactions is examined. The energy production landscape is being reshaped by distributed energy resources (DERs) : photo-voltaic panels, electric vehicles, smart appliances, and battery storage. Distributed energy sources such as microgrids, household solar installations, community solar installations, and plug-in hybrid vehicles enable energy consumers to act as providers of energy themselves, hence acting as 'prosumers' of energy.

Blockchain Technology facilitates managing the transactions between involved prosumers using 'Smart Contracts' by tokenizing energy into assets. Better utilization of grid assets lowers costs and also presents the opportunity to buy energy at a reasonable price while staying connected with the utility company. This technology acts as a backbone for 2 models applicable to transactional energy marketplace viz. 'Real Time Energy Marketplace' and 'Energy Futures'. In the first model, the prosumers are given a choice to bid for a price for energy within a stipulated period of time, while the Utility Company acts as an operating entity. In the second model, the marketplace is more liberal, where the utility company is not involved as an operator. The Utility company facilitates infrastructure and manages accounts for all users, but does not endorse or govern transactions related to energy bidding. These smart

contracts are not time bounded and can be suspended by the utility during periods of network instability.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Dragan Boscovic for the continuous work to support my thesis study and related research. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master's study, He has not only given me experience with research, but also helped me expose myself to Industry.

I would like to thank Dr. Kasim Selcuk Candan and Dr. Ming Zhao for their relentless support and guiding me throughout my thesis work.

I would like to thank Microsoft Corporation for providing resources on Microsoft Azure that supported deployment and testing blockchain systems.

I would like to thank Kyle Girardi for teaching me and making me understand more about utility infrastructure and operational details for energy markets.

I would like to thank all members of the Blockchain Research Lab at ASU: Arsh Padda, Manish Vishnoi, Kiran Suresh, Sasa Pesic, Nakul Chawla, Francis Mendoza for their contribution in discussions regarding consortium blockchains. I would like to thank Shammya Saha for helping me understand the power systems infrastructure for an energy marketplace.

Finally, I must express my very profound gratitude to my parents for providing me the unfailing support and continuous encouragement throughout my years of study and through the process of researching. This accomplishment would have not been possible without them.

Thank you.

Raj Anil Sadaye

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

A blockchain is a distributed database of records called blocks. It is a ledger of transactions that have been executed and shared among the participating members. The blocks are linked and secured using cryptography. In a broader sense a blockchain is defined as “a data structure that enables identifying and tracking transactions digitally and sharing this information across a distributed network of computers, creating in a sense a distributed trust network. The distributed ledger technology offered by blockchain provides a transparent layer and secure means for tracking ownership and transfer of assets” (Stroud, 2015).

The members of the blockchain network are called ’nodes’ or ’peers’, hence it is referred to as a peer to peer network. All the peers follow one protocol that takes care of messaging between these peers, and addition and validation of the new blocks formed. Every block consists of the following components:

- Block Header: The block header consists of
    - Block Version
    - Hash of the previous block
    - Merkle root of the transactions
    - Nonce (in public blockchains)
    - Timestamp of blocks
    - Difficulty bits (in public blockchains)
- Transactions: Number of transactions included in a block affect the block size.

A prime business driver for blockchain technology is to achieve greater transparency and substantiate accuracy of transaction data across the digital information ecosystem [34] . Mougayar (2016) identifies the value proposition of blockchains by using the mnemonic *ATOMIC (Assets, Trust, Ownership, Money, Identity and Contracts)*. The development of blockchain applications over the years can be can be categorized into generations:

1. The first generation consists of digital currencies better known as cryptocurrencies.Bitcoin continues to be the most well known application from this generation which was aimed at forming a public blockchain that can record transaction involving digital cash(assets in general)
2. The second generation expanded beyond cryptocurrencies to explore applications that could help with provenance, data security, validation of data, distributed storage etc. This is enabled through 'Smart contracts' which are agreements made between users and the distributed systems about actions to be taken when certain conditions are met.

1.1 Challenges For Building Blockchain Applications

The application of blockchain technology to business environments, beyond digital currencies has not been addressed extensively in a lot of literature. There exist some challenges that will need to be handled so that it becomes technically, economically and legally viable to use blockchain technology in enterprise environments. Some of these challenges are:

- Architecture
- Governance

- Deployment
- Data Privacy
- Scalability

There are 3 types of blockchains that handle these challenges in different ways:

1. **Public Blockchains**: Public Blockchains can be thought of as an open to all system. All nodes part of the public blockchain are anonymous. The nodes that share their computational power to add blocks to the blockchain are called as miners. On addition of a new block the miners receive some kind of reward (referred to as the coinbase transaction). A public blockchain scales up as more nodes are added to the system. The probability of data tampering is very low in a public blockchain due to the Proof of Work(PoW) consensus algorithm. The PoW algorithm involves every node solving a computationally intensive problem to be elected as the leader of a block. To alter a previous block an attacker would have to perform the compounded work from the current block all the way back, which is computationally impossible.

   There are also some drawbacks of using a public blockchain. If a controlling entity owns 51% of nodes in the network, the entity can corrupt the blockchain by gaining majority of the network. Also, miners consume high amounts of computing power which leads to waste of energy, space, money and hardware. Further transaction confirmation takes about 10 minutes, hence it is not instantaneous.
2. **Private Blockchains**: A private blockchain is operated and controlled by a single organization. This organization owns all the nodes, while also controlling the governance of the blockchain. Usually all the nodes of a private blockchain would be present on the premises of an organization. One worrying aspect of

private blockchains is that they let the middleman back in. The users submitting transactions have no control over ownership of the data.

3. **Consortium Blockchains**: A consortium blockchain operates under the leadership of a group of entities, thus enabling collaborative transformations involving multiple organizations [8]. These are also referred sometimes as permissioned blockchains. The participants identities are not anonymous. Each transaction being submitted needs to be signed by the participant key which keep track of provenance within the blockchain. The great thing about consortium blockchains is that participants may not fully trust each other(they might be competitors). However they can adopt a governance model that can help them establish agreements as well as handle disputes in a much quicker way. In such a blockchain, blocks are validated by predefined nodes called as validators. Since there is no competition for being elected the leader of a block, computational power required for generating a block is significantly lowered. This also leads to an increased scalability in terms of the transaction throughput[8].

1.2 Blockchain Technology and the Energy Sector

Energy Sector presents an opportunity to handle user accounts and payments using blockchains. End-users can be connected with the grid using a wholesale energy marketplace implemented using blockchain. Energy generated or consumed can be viewed as an asset on the blockchain. This asset can be tokenized and tracked through it's life cycle using devices connected to the grid. This would enable the consumers to trade and purchase energy directly from the grid rather than from retailers.

Addition of DERs is a demanding task and it raises new challenges in management and operation of electricity systems. These are difficult to predict and depend on weather conditions [9, 2]. Also they require flexible measures such as integration of fast-acting supply, demand response and energy storage services [2, 26].

Blockchains, due to their inherent nature, have the potential to provide a promising solution to control and manage trading energy surplus or flexible demand on a Peer-to-Peer basis. Commodity trading on an autonomous electricity marketplace can be secured and recorded using immutable, transparent and tamper-proof smart contracts [2]. Improved control of decentralised energy systems can also be achieved by adopting local energy marketplaces enabled by Peer to Peer energy trading. This would increase independent energy production and consumption(behind the meter activities), which impacts revenue and tariffs [22]. Blockchains can also assist in achieving integrated flexibility trading platform which might otherwise lead to expensive network upgrades. Often consumers are willing to pay a premium for buying green energy, but currently there is no guarantee about the origin of energy purchased. It is most likely that the energy used by the consumer is sourced by the closest fossil-fuel power plant[11, 31]. Blockchains promise complete transparency on the origins of the energy purchased such as its type, generating unit and exact location produced

Blockchain applications can help secure communications from industrial control systems and other operational technology (OT) protocols (MODBUS, DNP3, BacNet, etc.) by including an advanced crypto-signature that assigns a data signer, authenticity of the data, and time of signing to a data asset. This signature is represented by including the hash of the data in the signature. Combining crypto-graphic signing events and distributed infrastructure may help increase fidelity of data, competition and real-time energy exchange for micro-grids and building to building energy generation and sale[29]. Increased data fidelity afforded by blockchain could also help detect targeted cyber-attacks and increase resiliency of DER grid integration. Current techniques used in energy distribution and buildings-to-grid connections are vulnerable to cyber-attacks[28] . The integration of DERs without appropriate cyber-security measures including trustworthy communications and monitoring could potentially destabilize the power grid and create outages and reliability problems for customers.

Companies in the utility sector are concerned about privacy of data as well as membership of the network. Hence consortium blockchain solutions are an ideal choice for implementing an energy marketplace for 2 main reasons:

1. Data Permissioning through Access Control.
2. Selective Membership to pre-approved parties.

1.3 Hyperledger Fabric: A Consortium Blockchain Solution

Hyperledger Fabric(HLF) is a modular and extensible open-source system for deploying and operating permissioned blockchains[4]. It supports modular consensus protocols. It allows developing distributed applications without relying on a native cryptocurrency. Figure 1 illustrates the organizational architecture of HLF.

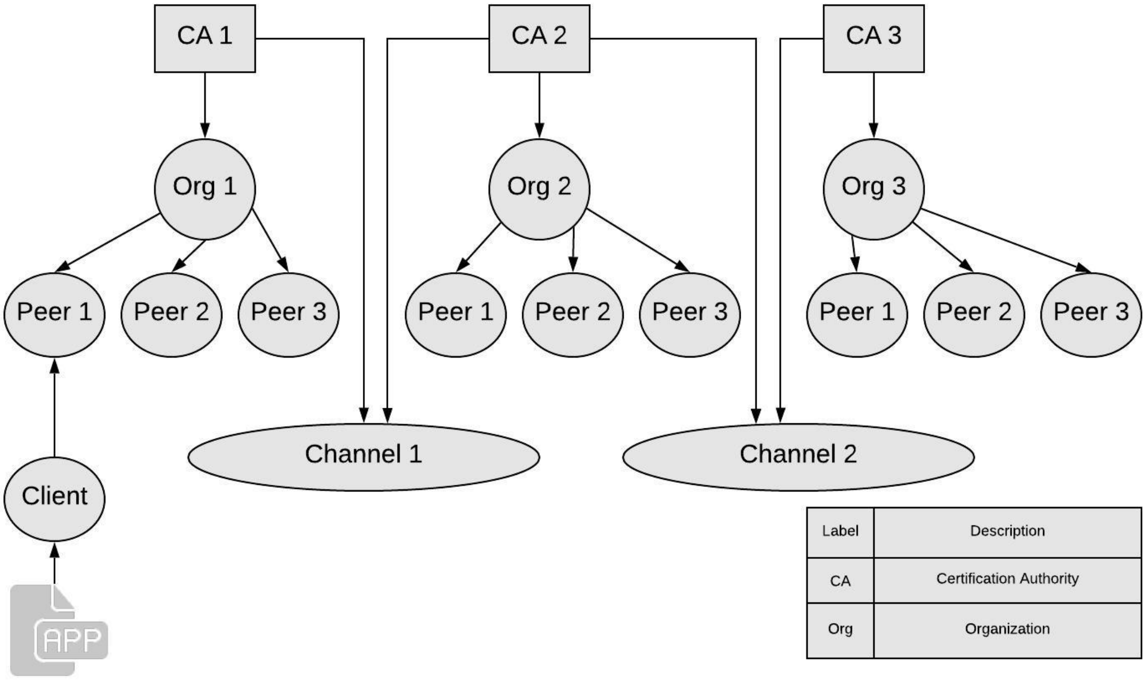

Figure 1. Hyperledger Fabric's Organizational Architecture

- **Certification Authority**: A certification authority(CA) distributes certificates to all participants of the network. These certificates are digitally signed by the CA. Hyperledger Fabric provides a built-in CA component called Fabric CA. It manages the digital identities of participants in the form of X.509 certificates. Each organization hosts a separate node that acts a certification authority [16].
- **Organizations**: An organization is a managed group of members of the blockchain network. Every organization manages their members using a Membership Service Provider(MSP). An organization consists of peers, users and clients [18].
- **Membership Service Provider(MSP)**: An MSP identifies which Root CAs and Intermediate CAs are trusted to define the members of a trust domain. An MSP can also identify specific roles an actor might play(eg. members or admins).

An MSP can be implemented at locally different levels in the blockchain network [18].

- **Peers**: Theses are nodes in the blockchain network that commit transactions as well as maintain the state and copy of the ledger. Peers can also act as endorsing peers(also called endorsers). Every chaincode specifies an endorsement policy that may refer to a set of endorsing peers which are used for validating transactions [19].
- **Clients**: These nodes submit transaction invocation to endorsers and broadcast transaction proposals to the ordering service. These nodes are usually controlled by end users through an application. They must be connected with a peer of its choice [13].
- **Channel**: A channel is a private subnet of communication between two or more specific network members. It's main purpose is to conduct private and confidential transactions. A channel is defined by members(organizations), anchor peers per member, the shared ledger, chaincode application, and the ordering service node. Every transaction is executed on a channel, where each party must be authenticated and authorized to transact on that channel [14].
- **Chaincode**: A smart contract is called chaincode in a hyperledger fabric network. It's a self executing logic that encodes the rules for specific types of network transactions. It is installed and instantiated onto a channel's peers by an authorized member of the network. End-users invoke chaincode through a client-side application that interfaces with a network peer. Chaincode transactions are appended to the shared ledger and modify world state, provided they are validated [20].

- **Ordering Service**: The consensus algorithm in a public blockchain is probabilistic which guarantee ledger consistency, but is still vulnerable to divergent ledgers(sometimes called forks). Hyperledger Fabric relies on deterministic consensus algorithms which avoids forks. For this solution, the ordering service is implemented using Apache Kafka. Kafka is a Crash Fault tolerant implementation of a messaging queue that uses a 'leader and follower' node configuration. Transactions are replicated from the leader node to the follower nodes. If a leader goes down, one of the followers goes on to become the leader, ensuring fault tolerance.

1.4 Endorsement Policies in HLF

Every chaincode has an endorsement policy which specifies the set of peers on a channel that must execute chaincode and endorse the execution results in order for the transaction to be considered valid. These endorsement policies define the organizations (through their peers) who must "endorse" (i.e., approve of) the execution of a proposal[15]. Endorsement policies can be constructed using a combination of *expressions* and *principals*. A transaction submitted must satisfy the endorsement policy before being signed as valid by endorsing peers

Expressions can be : *AND*, *OR*, *Outof*.

Principals can be: *Org.admin*, *Org.member*, *Org.client*, *Org.peer*.

For example,

- *AND('Org1.member', 'Org2.member', 'Org3.member')* requests one signature from each of the three principals.

- *OR('Org1.member', 'Org2.member')* requests one signature from either one of the two principals.

1.5 Transaction Flow in HLF

Figure 2 depicts the basic workflow of transaction endorsement in HLF. This workflow is divided into four phases which can be described as follows:

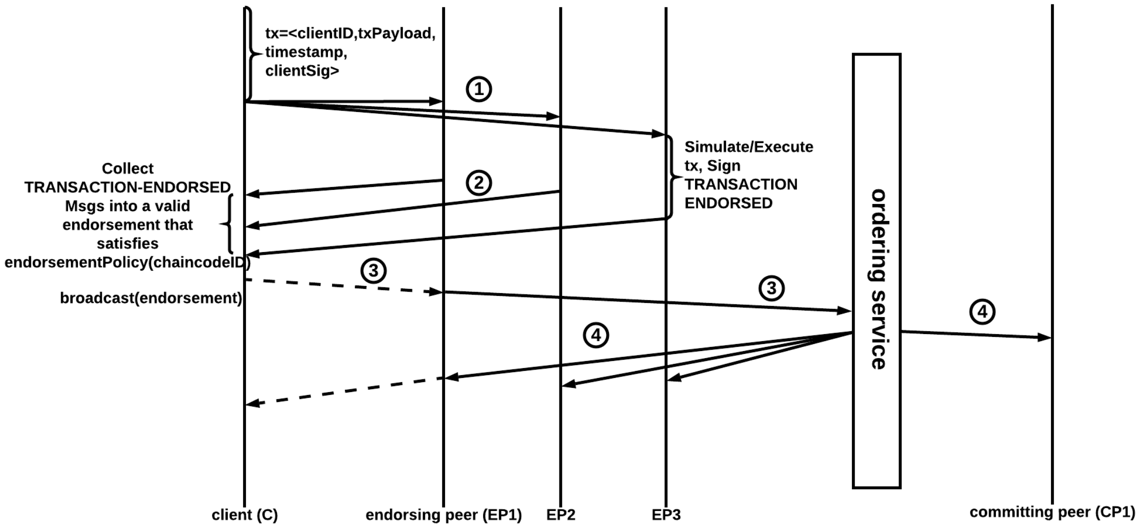

Figure 2. Transaction Flow in Hyperledger Fabric

1. **Transaction Proposal:** A client application that wishes to execute some operation on the blockchain, submits a *transaction proposal* that consists of clientID, transaction payload, timestamp and client Signature. This proposal is sent to one or more endorsing peers. In this phase no changes are made to the state of distributed ledger.
2. **Simulate and Sign:** The endorsing peers accept this proposal and simulate the transaction with the current version of the ledger. Every peer maintains

a separate copy of the ledger which are synchronized together. At the end of this simulation, a Read/Write set would be generated indicating the updated versions of keys whose value would have changed. The endorsing peer signs this Read/Write set with its keys and sends back the *endorsement response* to the application. Hence this endorsement response contains cryptographic information about the endorsing peer, the transaction and the Read/Write set of the transaction. At this phase no changes are made to the state of distributed ledger.

3. **Invocation request:** The application collects all the endorsement responses received from the peers, and packs them into an *invocation request* which is sent to the ordering service. Ordering service verifies all the cryptographic material and verifies whether all the endorsement responses satisfy the *endorsement policy*. Every chaincode on a channel has a particular *endorsement policy*. If the endorsement policy is invalid after taking into account the endorsement responses from the invocation request, the *invocation request* will be rejected and the transaction would make no changes to state of the distributed ledger. However that transaction will still be stored on the blockchain. This helps with achieving provenance, since in the future we can look back at the details of this transaction. We can tackle security issues by auditing this stored record.

   Also, the ordering service verifies the Read/Write sets within the *endorsement responses* in the *invocation request*. All the Read/Write sets must match. If they do not match, the ordering service treats it as if a peer is out of sync or has been compromised, and the transaction will not be committed onto the ledger. If the policy is valid and all the Read/Write sets match, the orderer sends the data to all the peers that are part of the channel.

4. **Commit Phase:** Every peer applies the Read/Write set to the state of the distributed ledger. There might be multiple client applications submitting invocation requests at a time. The ordering service handles the ordering of these invocation requests according to timestamp.

1.6 Handling The Double Spending Problem using HLF

The endorsement based model described in section 1.4 handles the *double spending problem*. The *double-spending problem* could be described briefly as follows:

For any system that comprises of digital assets, an asset can be copied, and it is not possible to distinguish the copy from the original. *Double spend* means that a participant of the system tries to use the original and the copied asset at different places. In order to solve this problem, we need to make sure that the asset can be used/spent only once. HLF prevent double spending problem through its endorsement process and key versioning.

When the transactions are simulated on the endorsing peers, the Read/Write set generated contains the keys that have been updated are part of the Read/Write set. Every key has a version which is increased by one every time it is updated. The Read/Write set captures the changes made to the version of every key.

Suppose a key k has current version 1 and after the simulation its Read/Write set indicates its version would update to 2. While the simulation is taking place, some other transaction updates this key in parallel, and changes its version permanently to 2. When the invocation request for the transaction is received by the ordering service, it would discover that the version of the key has already been changed while the simulation was running. Hence the simulation is no longer valid since it was executed based on the previous value of the key. This transaction would be stored on the blockchain but it will not update the state of the distributed ledger.

Hence checking the current version of the key in the ledger against the version in the Read/Write set of an invocation request before committing the transaction helps us in solving the *double spending problem*.

1.7 Block Formation In Hyperledger Fabric

The first block of an HLF blockchain is called the *genesis block*. The contents of the genesis block are different from those in a public blockchain. Inside the genesis block, the public keys of all the entities of the network are stored. This implies that the configuration of the HLF blockchain network is part of the blockchain itself. Hence it is impossible for some adversary to modify this configuration. Only a user with pre-defined privileges can update the configuration.

Along with the public keys of entities, the *genesis block* also store the configuration of all the channels in the blockchain network. The next blocks added to the blockchain contain transactions submitted by the client application. The parameters that affect block size and block arrival time are configured in the genesis block. These parameters are set depending on the frequency of incoming transactions as well as the transaction size. The ordering service packages the transaction invocation requests into blocks using the following parameters:

- **MaxMessageCount:** This parameter defines a limit of transactions that can be included into a block. A block will never have more than *MaxMessageCount* transaction inside it.
- **AbsoluteMaxBytes:** This parameter defines the upper limit for the size of a block.

- **PreferredMaxBytes:** This parameter affects the block arrival time. If there arises a scenario that we can form a block under *Preferred max bytes*, then a block would be cut prematurely, and transactions larger than this size would appear in their own block. For example, lets assume *PreferredMaxBytes* is set as 4 MB. When transactions are being packaged, if the ordering service notices that the next batch of messages which would form the next transaction is larger than 4MB, the block would be preempted and the next transaction gets a separate block.

1.8 Why Choose Consortium Blockchains For Building Energy Trading Marketplace?

Blockchain technology presents security and optimization benefits in its application to energy infrastructure. It can provide enhanced integrity of energy data by supporting multi-factor verification through a distributed ledger. In the current scenario, the utility company has control as well as ownership pf all energy related data which is stored and maintained in centralized servers maintained by the utility company. There is no way for a users to audit or validate the data which is stored. Moreover, the utility companies also are incapable of providing such a service using the current technology that is being used.

With the inclusion of distributed energy resources, there are multiple parties who are involved in negotiations for energy. One way to deal with this problem is to find a third party that all actors trust to provide the service of audit and verification of data. This would however introduce it's own set of problems:

- The process would be expensive.

- It would be very difficult to find a third party that all the actors trust.
- The third party introduced would be susceptible to human errors. There from within the organization alter the data, which the actors involved would not be able to detect.

A consortium blockchain has the ability to solve all these problems. A consortium blockchain relies on the following pillars:

- Cryptographic verification
- Updating after data validation.
- Enforcement of business rules(through smart contracts and endorsement policies).

If all parties join a consortium blockchain, all of them would have the same information(enforced by smart contract). In case the utility tries to alter data, all other parties would be able to view that the Read-Write sets won't match. Hence the data change won't be valid and won't be applied on the ledger. If an external adversary attempts to alter all the data at the same instant, the process of committing data to the ledger would fail cryptographic verification(since the membership is permissioned). Hence all parties within a consortium blockchains trust the utility, but the operations on the ledger are always accompanied with verification and validation measures. Hence rather than being a "trustless" blockchain, a consortium blockchain follows the principal "trust, but verify".

1.9 Hypothesis

With Hyperledger Fabric as an underlying technology it is possible to build an open marketplace for energy trading which can operate in parallel with the existing system. Two models for transaction flow are proposed and implemented:

- Real-Time Energy Transactions
- Energy Futures

Using a common underlying architecture, both these models are tested and their performance is compared based on latency, throughput and resource utilization as the metrics. The tests involved varying the characteristics of the hyperledger network to find the best performing configuration. Further this configuration is used to compare the performance of both models in terms of response times for smart contracts.

1.10 Organization of Thesis

The content of the thesis is organized as follows:

- Chapter 2 focuses on the Background and Related Work.
- Chapter 3 describes the design and implementation of the two models using hyperledger fabric.
- Chapter 4 contains the Experimental Details. It contains information for experimental setup, data set, evaluation criteria and performance of the system.
- Chapter 5 concludes the thesis and describes the direction for future work.

Chapter 2

RELATED WORK AND THEORY

2.1 Single Market-Clearing Price In Electricity Markets

Typically energy related product in electricity markets are traded through a daily(or day-ahead) or short-term auction process where the pricing rule is set to a uniform price. Even though there exist several offers and bids for various prices at any given hour, all energy is sold at one single price, known as market-clearing price(MCP) [35]. An auction finds the generators with the lowest production costs to meet consumer demand in electricity market.

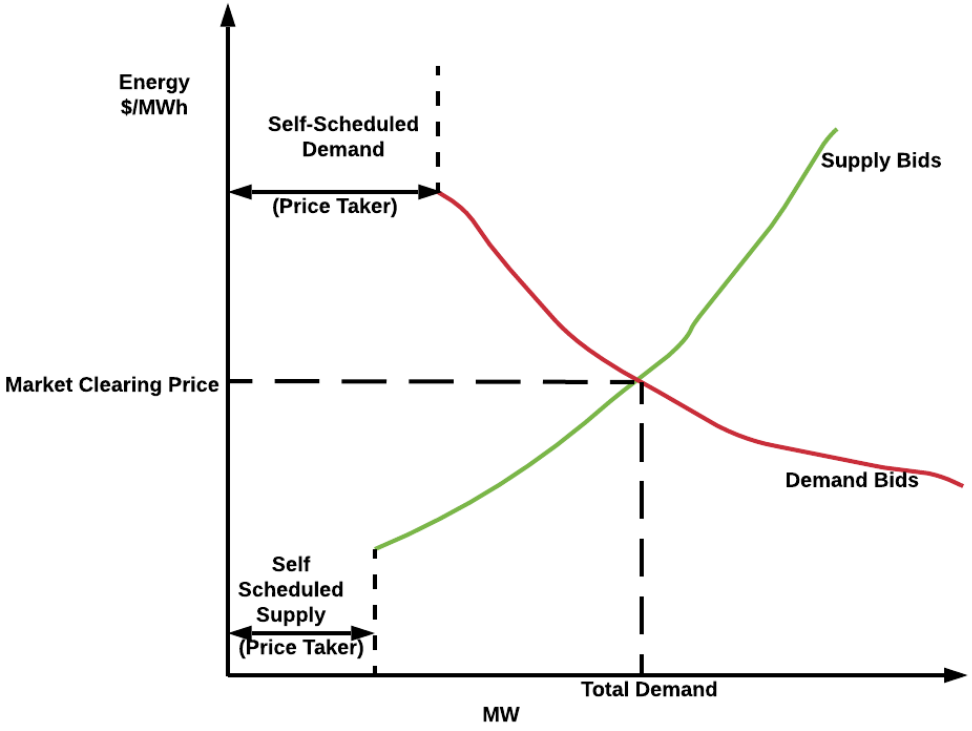

Figure 3. Graphical Representation of the MCP Auction

For example, suppose *generator A* has low production cost of \$35/MWh and *generator B* has production cost of \$90/MWh. During off-peak hours, when demand is lower, generator A can suffice the demand and hence the MCP would be around \$35/MWh. During on-peak hours, both the generators would be needed to suffice the demand, hence the MCP would be around \$90/MWh. The difference between the MCP on-peak and the production cost of *generator A* enables the capital incentive.

The auction used in electricity markets is often called "offer-based economic dispatch" and is used to choose generators with the lowest offers to meet demand. This auction is highly centralized. If this centralized process was abandoned, it would require a coordinated system that would achieve the goal of using the lowest production cost generation to meet demand. Also, by making transparent prices available based on offers, effective market monitoring can also be carried out and generation investment decisions can be better informed.

2.2 Transactive Energy Concept

There is a need for transformation in the current power distribution system due to penetration of flexible loads and intermittent generation resources [12]. To properly integrate these new players into the energy marketplace it is important to build a system to manage interactions between these autonomous prosumers. Transactive energy systems can provide a solution for this. The Gridwise Architecture council defines transactive energy as an approach that assigns value to facilitate dynamic balancing between independent power producers(IPP).[7].

In transactional energy systems operational decisions are made by exchanging value based information. This information is captured while executing transaction involving prosumers. Rules of these transactions are designed to align the behaviour of prosumers with the supply-demand balance.

When we talk about supply-demand balance we have to look at the concept of demand response. Demand response can be defined as changes in electric usage by end-user customers from their normal consumption patterns in response to changes in the price of electricity over time or due to incentive payments designed to lower electricity usage at the time of high wholesale market prices.[1] This is the way in which high demand is dealt with.

The transactive energy approach offers a way for prosumers to more closely match and balance energy supply and energy demand. If energy providers and users can agree on the value of the electricity at a certain point in time and place, the prosumers can each make a decision if they want to proceed with that transaction at that given price.[30] Smart contracts on a blockchain have the potential to digitally facilitate, verify and enforce the negotiations taking place on the transactive grid.

The transactive energy approach offers key benefits to consumers:

1. Better utilization of grid assets can lower costs, especially during high demand hours. Opportunity to buy energy at a reasonable price while staying connected with the utility provider during peak hours and peak usage seasons.
2. Greater reliability and resilience.
3. Increased choice and information will give consumers greater control over personal energy use.
4. Increased use of renewable energy resources gives individual consumers the satisfaction of contributing to larger, societal goals.

Benefits for utility companies are:

1. Offset energy demand during on peak hours by using renewable energy resources leading to reduction in cost of buying additional fuel.
2. Harnessing the power of renewables while being compensated for infrastructure

Benefits for prosumers are:

1. Opportunity to get returns on surplus energy
2. Ability to transact with energy while choosing price

2.3 Existing Solutions

2.3.1 Power Ledger

The Power Ledger Platform is a trustless, transparent and interoperable energy trading platform that supports energy applications, with a frictionless energy trading token, *Sparkz* [25]. The platform enables trading units of electricity (kWh) by way of pre-purchased tokens. The platform follows a dual token ecosystem (POWR and Sparkz).

The platform supported an application for P2P electricity trading marketplace between prosumers and local consumers. POWR serves as the fuel of the ecosystem. POWR tokens can be bought on exchanges. POWR can be converted to Sparkz, the marketplace's native currency, which can be traded on the company's private blockchain [3]. Due to the dual-token ecosystem, the applications supported on the platform support two concurrent models. Figure 4 shows the Retail as well as P2P

Direct model. The Retail model supports working with existing market structures. The P2P Direct model supports moving towards a deregulated market structures.

The system consists of three layers. The high transaction volume of P2P energy trading is managed using a public Ethereum blockchain, where the ecosystem interfaces with third party exchanges [25]. The public Ethereum blockchain makes up the first layer of the system. The second layer is the Power Ledger core which consists of public smart contracts. The public smart contracts enable exchange of POWR and Sparkz tokens through a Smart Bond contract for Application Hosts. The third layer is a consortium blockchain called *EcoChain.* EcoChain is a private Proof of State(PoS), low-power blockchain that is used for energy data collection and settlement.

Power Ledger's P2P pilot project agreed a pricing scheme of 20 cents/kWh of energy purchased through the platform. Electricity charges went through a 75/25 split between prosumers and utility company.

Although Power Ledger is marketed as a P2P application, for any community wishing to integrate the Power Ledger Platform, it will need the agreement of local utility company who own the local infrastructure. This would create a problem in expanding the platform, since energy companies differ across the world and so do regulations related to energy sector.

2.3.2 Brooklyn Microgrid

The Brooklyn Microgrid is P2P energy trading platform run by Transactive Grid, that aims to develop sustainable energy network powered by community residents' rooftop solar installations. Transactive Grid is a partnership between LO3 Energy, Consensys, Siemens and Centrica [3]. The goal was to enable local energy producers to automatically conduct transactions with local energy consumers in near-real time.

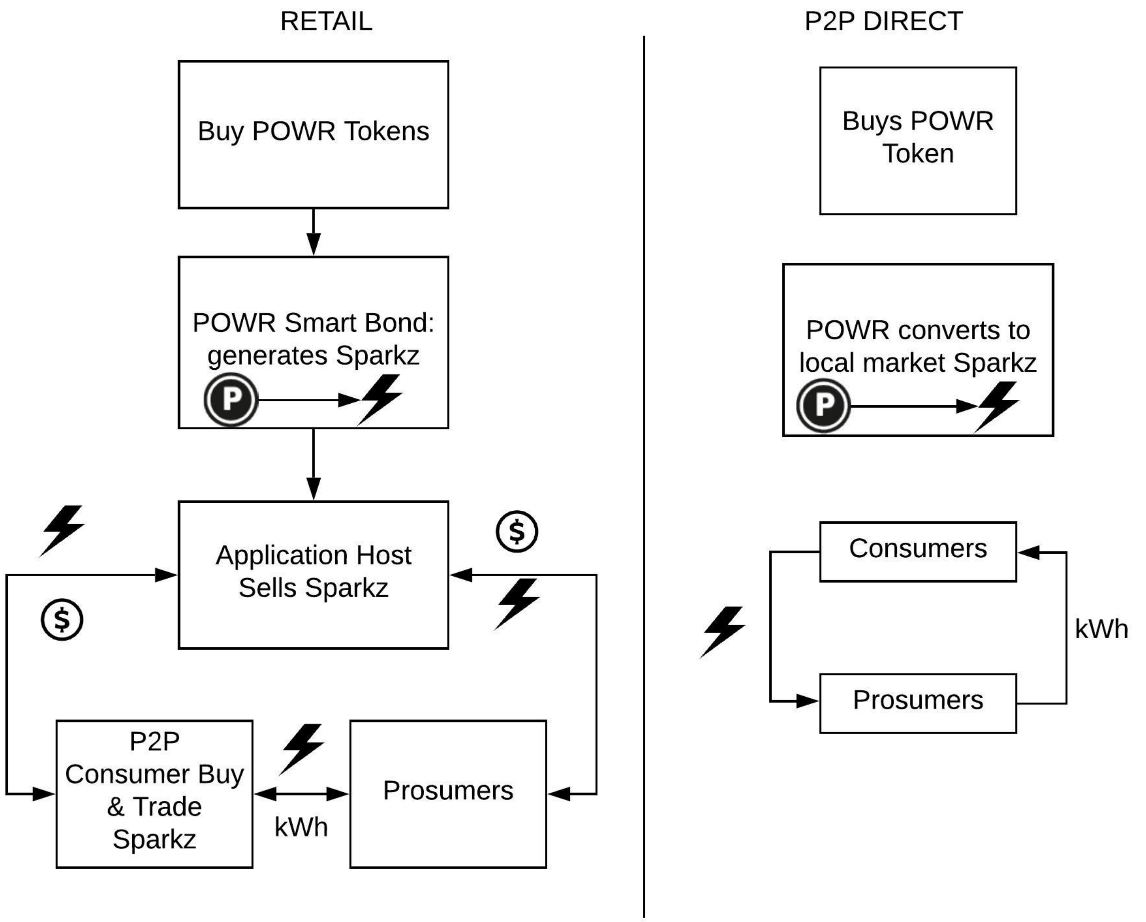

Figure 4. Dual Token Ecosystem Used by the Power Ledger Platform

The project was piloted in April 2016 and the trial run included 5 prosumers and 5 neighbouring consumers.

It is based on a private blockchain using the Tendermint protocol [5]. Transactive energy smart meters are coupled with analog meters which handle transactions with tokens on the blockchain [27]. The Smart Meters consist of Transactive Grid Element Generation 2(TAG-e G2). TAG-e G2 is a small compute unit attached to the smart electricity meter. This unit is connected to the blockchain marketplace via Ethernet or WiFi. The aim of the project was to implement a double auction market mechanism, which clears energy at discrete time intervals. However the trial run did not actively use the market mechanism and the electricity price was pre-determined and fixed at the traditional price of energy in Brooklyn.

The participants could virtually trade electricity, they were not physically connected to microgrid. Most of the participants simply continued to use the main grid. When two participants trade electricity, one participant feeds excess solar power back to the distribution grid, and the other participant consumes electricity from the grid. However, the utility has a monopoly over electricity sales, and the participants exchanges renewable energy certificates.

A takeaway from this project was the fact that a private blockchain protocol can successfully implement and operate a microgrid energy market [5]. However this project does not consider inclusion of participants who own energy storage equipment, electric vehicles. Also, the design focused on keeping energy trading independent of the utility company, which would be a monumental task, if this was deployed in the real world.

Chapter 3

DESIGN AND METHODOLOGY

3.1 Electric Grid As A Platform

The future for utility companies, would be fundamentally operating like a platform, enabling a large amount of transactions, as well as facilitating products and services across a network. This also opens up revenue streams for each of the services being offered, through transaction fees. A sudden shift to a totally deregulated system is not possible, due to the physical infrastructure barriers of the electric system.

3.1.1 Real-Time Energy Transactions

Enabling real-time energy transactions involves introducing a list of business rules in the system. An important aspect of such a system is deciding on the market-clearing price (MCP) for energy. Even though there may be several offers and bids at various prices for any given period of time, all energy is sold at one single price, known as the market clearing price [15, 30].This system is governed by certain rules that control the submission and approval of transactions.

The first rule among them is the related to time intervals. A cycle repeats after a certain time interval(T) which decides the behaviour of the system. This time interval(T) is further divided into bidding time(b) and closing time(T-b). During bidding time energy trading blockchain accept transactions from users, allowing them to bid for energy. During closing time, the energy trading blockchain stops

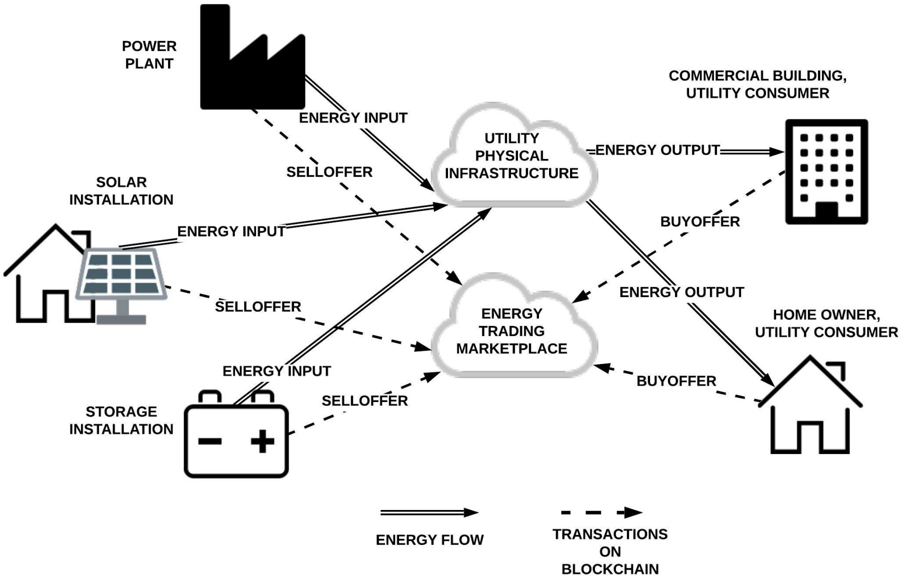

Figure 5. Energy and Transaction Flow on Energy Trading Marketplace

accepting transaction and calculates the MCP which would be used over the next time interval(T). Market Clearing Price is calculated based on the equation (3.1) [32].

$$MarketClearingPrice = \frac{\sum_i^N price_i * amount_i}{\sum_i^N amount_i} \tag{3.1}$$

Where,

- $price_i$: Price of the i^{th} buyoffer
- $amount_i$: Amount of the i^{th} buyoffer
- N: Total number of buyoffers

Assets on the distributed ledger can be divided into three types:

1. **Buyoffer**: All actors who wish to buy energy, submit transactions that lead to the buyoffer being recorded on the blockchain. Actors are allowed to submit

buyoffers within *b*. The buyoffer consist of the following attributes being recorded on the blockchain.

- Owner: Identifier for the actor submitting the transaction.
- Username: Username for the actor submitting the transaction.
- Timestamp: The time when the transaction was submitted on the application side. The timestamp is broken down into date, hour, and minute by the smart contract.
- Offer Type: An integer which is set to 1 for a buy offer.
- Amount: Amount of Energy (KWH) that the actor wishes to buy.
- Price: Price of Energy (/KWH) that the actor proposes.
- Status: Reflects the status of buy offers submitted to the energy trading marketplace.

2. **Selloffer**: Photovoltaic installation owners as well as energy storage owners can submit selloffers which will be recorded on the blockchain. The selloffer consists of the following attributes on the blockchain.

    - Owner: Identifier for the actor submitting the transaction.
    - Username: Username for the actor submitting the transaction.
    - Timestamp: The time when the transaction was submitted on the application side. The timestamp is broken down into date, hour, and minute by the smart contract.
    - Offer Type: An integer which is set to -1 for a buy offer.
    - Amount: Amount of Energy (KWH) that the actor wishes to sell.
    - Price: Price of Energy (/KWH) that the actor proposes.
    - Status: Reflects the status of buy offers submitted to the energy trading marketplace.

3. **Account**: The balance of energy assets traded by actors is tracked by their accounts. Actors can have a separate account on every channel that they are part of. These multiple accounts can be aggregated on the application end to show a net balance for the user. An account consists of the following attributes:
   - Owner: Username for the actor submitting the transaction.
   - ebalance: Amount of energy locked by smart contracts based on buy and sell offers that have been accepted. *T*.
   - tbalance: A token on the energy blockchain is an attribute included to record transfer of energy. Whenever the MCP is decided, the actors trade tokens between each other. The buyer transfers tokens to the seller and energy is transferred from seller to buyer.The price of a token can be pegged against fiat currency or against any other stable coin. In this implementation we have kept the value of 1 token = 10 cents. Hence if 10 kWh of energy is offered at 10 cents per hour, the buyer will receive 10 tokens.

The consumer account has an additional attribute called 'consumed energy' which represents the energy consumed by the consumer. This is updated by the meters at the consumer site via requests sent to the energy trading application.

The battery account has also has an additional attribute called 'stored energy' which represent the energy stored that can be used for locking into sell offers and also providing energy to the grid. This is updated by the inverters at the storage site via requests sent to the energy trading application.

One important aspect of this system is the provision of endorsement policies. In a traditional centralized system, the utility company will have total control over

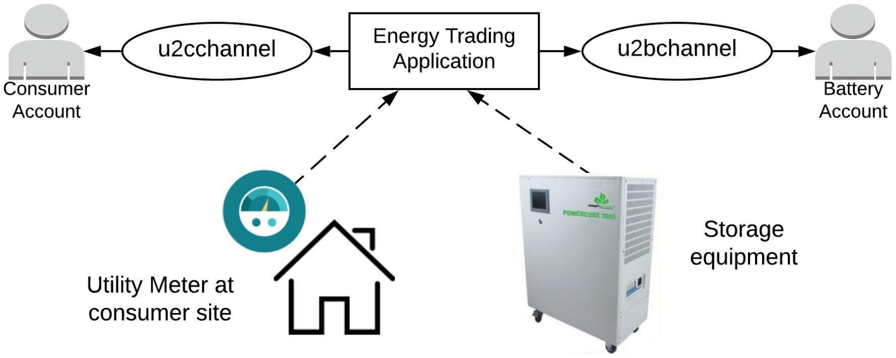

Figure 6. Physical Equipment Updating Account Information on Channels

confirmation of transaction as well as ownership of data. In the blockchain based system an endorsement policy decides whether a transaction is approved or not. For a smart contract an endorsement policy decides the number of approval signatures it should receive before a transaction is committed to the ledger. Equation (3.2) is an example of an endorsement policy. This policy implies that the smart contract should be installed on Utility, Solar as well as Battery peers and until all members sign the transaction, it will not be committed to the ledger. If a malicious actor tries to modify the ledger, trying to take it to an inconsistent state, the Read-Write set would create a conflict and the transaction won't make changes to the state of ledger.

$$AND(UtilityMSP.member, SolarMSP.member, BatteryMSP.member) \tag{3.2}$$

Endorsement policies are highly flexible. To enable peer to peer exchanges between actors we might want to keep some data isolated from the utility company. In such a situation, the exchange would be over two channels. On the first channel, the buyers

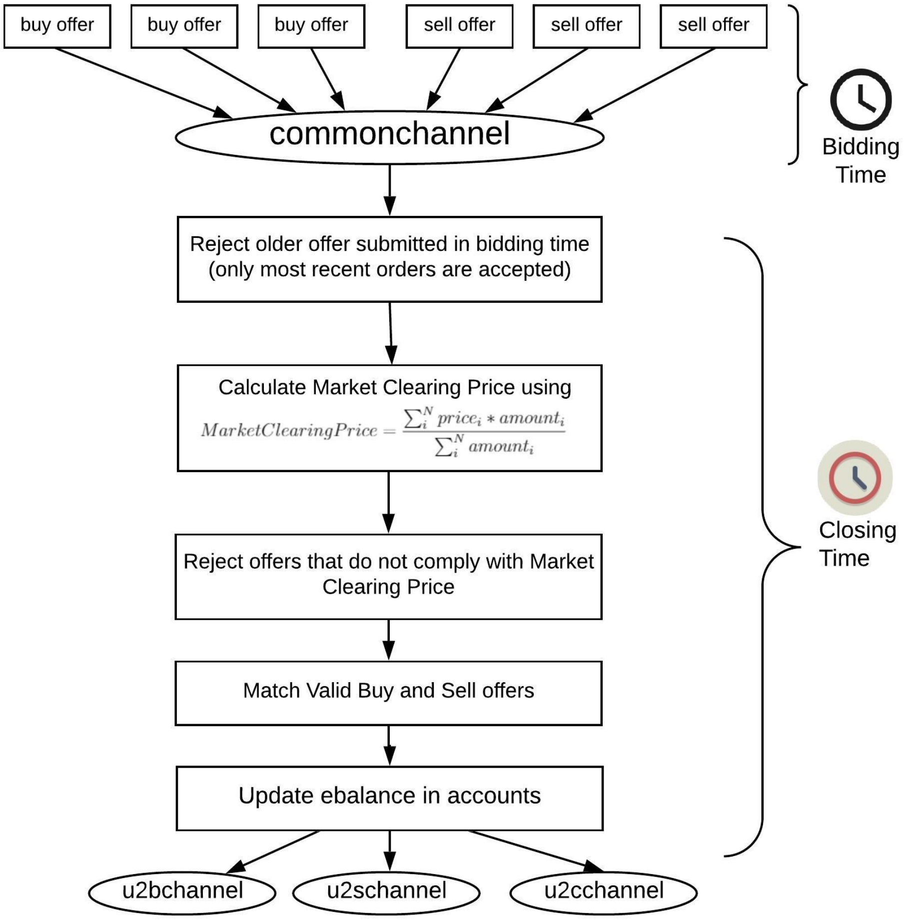

Figure 7. Flowchart Describing Flow Of Data For Real-Time Energy Transactions

and sellers can transact using offers. On the second channel any relevant information would be conveyed to the utility company.

The endorsement policies for smart contract deployed on all channels for Real-Time Energy Transactions are as follows:

- commonchannel: *AND(UtilityMSP.member, SolarMSP.member, BatteryMSP.member, ConsumerMSP.member)*
- u2schannel: *AND(UtilityMSP.member, SolarMSP.member)*
- u2bchannel: *AND(UtilityMSP.member, BatteryMSP.member)*

- u2cchannel: *AND(UtilityMSP.member, ConsumerMSP.member)*

Figure 7 illustrates the flow of transactions for this model. The flow can be described as follows :

1. During bidding time all users can submit buy and sell offers on the commonchannel. The commonchannel facilitates the energy trading marketplace.
2. At the end of bidding time, only the most recent orders made by the users are accepted into the system. All the stale offers are rejected.
3. Further using the remaining offers, the market-clearing price is decided by taking into account the demand for energy using equation 3.1.
4. Any offers that do not comply with the Market Clearing price are rejected using these conditions:
   - Sell offers with price *greater than* Market-Clearing Price are rejected.
   - Buy offers with price *lesser than* Market-Clearing Price are rejected.
5. All the remaining valid buy and sell offers are matched against each other.
6. The amount of energy locked in the accepted offers is updated on the accounts of the corresponding users. These transactions take place on the individual channels that the organizations have with the Utility Company.

Figure 8 shows all the status transitions for buy and sell offers submitted on the energy trading marketplace. An offer initially appears as 'submitted'. If the offer is a stale order it is 'rejected'. When the Market Clearing Price is calculated, all the offers not compliant with MCP are updated to the 'rejected' status. When matched the order status would be 'fulfilled' else the status would be 'not fulfilled'. Once the offer amount has been updated to user account, the offer enters the 'locked' status.

The offer is 'suspended' if there's an instability condition. Once all energy locked is transferred status is updated to 'completed'.

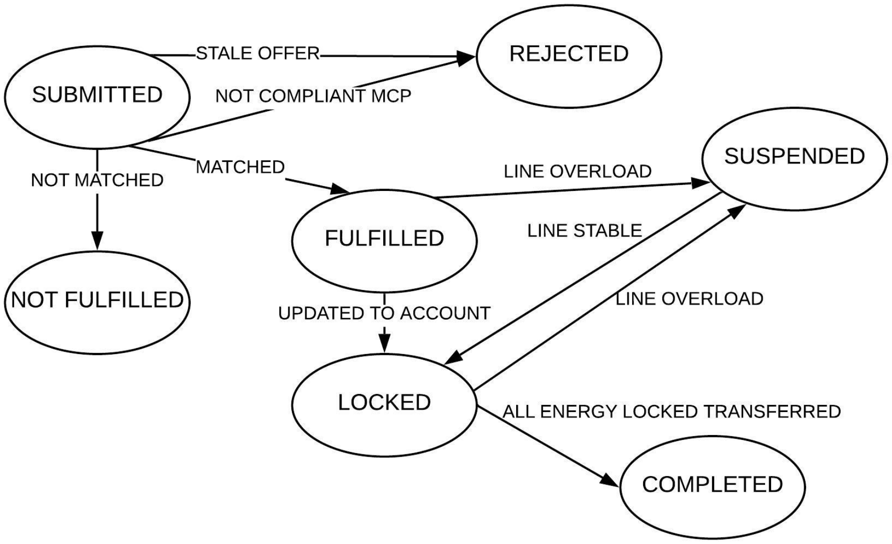

Figure 8. Status Transition Diagram for Real-Time Energy Transactions

3.1.2 Energy Futures

This approach aims to create an open marketplace for bidders of energy by allowing offer contracts to exist over an extended period of time. A futures contract is a financial contract that obligates the parties to transact an asset at a pre-determined future date and price. The buyer must purchase or the seller must sell the underlying asset at the set price, regardless of the current price.

Usually in a derivatives market(where futures are usually traded) the underlying assets include physical commodities or other financial instruments. For enabling an open energy marketplace, the underlying asset is the kilowatt-hours of energy specified

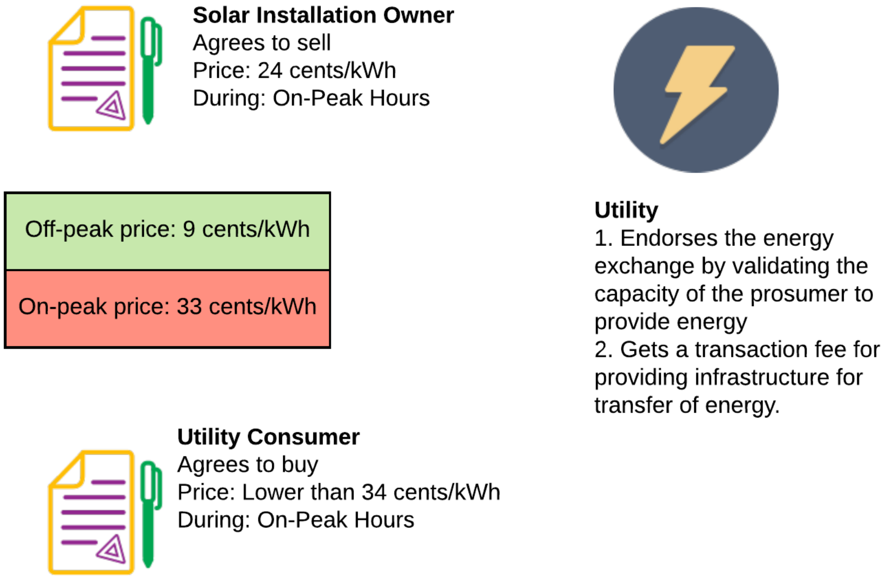

Figure 9. A scenario for the energy futures contract

in a contract. In such a marketplace buyers and sellers can directly agree on terms regarding the price of energy. One key point to consider is the provision of physical infrastructure for transfer of energy. Although prosumers can sell their energy most of them do not have the capability of providing a direct line to the buyer. Hence the utility acts as a provider of infrastructure rather than a complete operator. The utility facilitates the transfer of energy from between the two trading parties and gets compensated for that service using a small transaction fee. Figure 9 outlines a scenario for the energy futures contract trading marketplace.

This approach does not follow the same transaction flow or the business rules used by the first model. An Energy futures contract is similar to a wholesale bilateral contract between two parties trading energy. The formal definition for a bilateral contract is an agreement between two parties to exchange electric power under a set of specified conditions such as amount of electricity, time of delivery, duration and price. [23, 10]. Similar to a bilateral contract can be either physical or financial.

- **Physical Contract**: Power transacted bilaterally must be self-generated and self consumed on specified network buses.[24][6]
- **Financial Contract**:Power transacted is based on net metering. This contract guarantees the difference between the contract price and utility price and has no direct physical transmission implications.[21]

In the previous model, even though prosumers were involved in the decision of the market clearing price, the utility was still in a dominant position in terms of the endorsement for smart contracts on the commonchannel. Energy Futures contracts aim at introduction of liberalization into the energy bidding and matching process using the following ways:

- **Elimination of the Market Clearing Price Parameter**: Rather than trading energy at a common price for all actors involved in the system, the price for energy is decided by matching the buy and sell offers bilaterally on the commonchannel.
- **Utility as a non-dominant actor on the commonchannel**: The utility company is not involved in the endorsement of transactions on the commonchannel. The company however is still involved with maintaining user accounts on the one-on-one channels. Hence the utility company can intermit the enforcement of a contract if it is not favorable to add energy to the grid based on line constraints.
- **Duration decision taken by the user**: Energy Futures Contracts are extended over time. Hence users can decide on the duration over which the contract is valid.

- **Option to choose the time of delivery**: Users have an option to provide a time of delivery, which is an interval during the day during which the smart contract is enforced.

The definition for buy and sell offers on the distributed ledger have a slight difference. Three additional attributes are used for buy and sell offers in order to extend them over time:

1. **Start** Hour: The start of time of delivery
2. **End Hour**: The end of time of delivery
3. **Duration**: The timestamp till which the smart contract is valid.

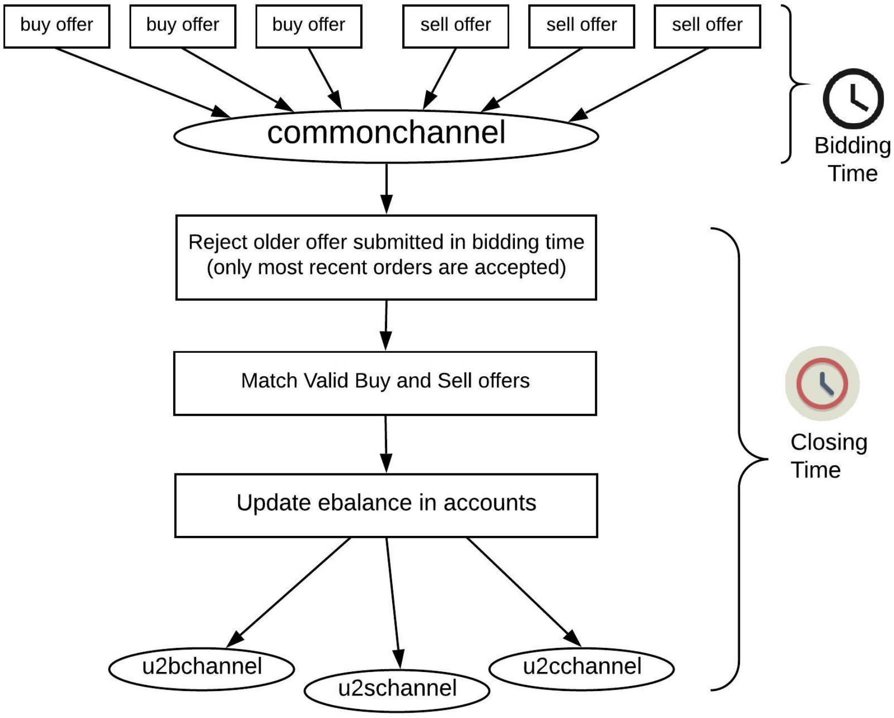

Figure 10. Transaction flow for energy futures

The transaction flow for energy futures is illustrated by figure 10. Similar to the first approach all the buy and sell offers are submitted on the commonchannel by the participants of the network during bidding time. During closing time, stale offers submitted by actors are rejected(Only the most recent offer is used for matching). All the valid (non-stale) offers are matched. The offers which are not fulfilled are retained for further rounds of matching. The amount for the orders is updated in the user accounts.

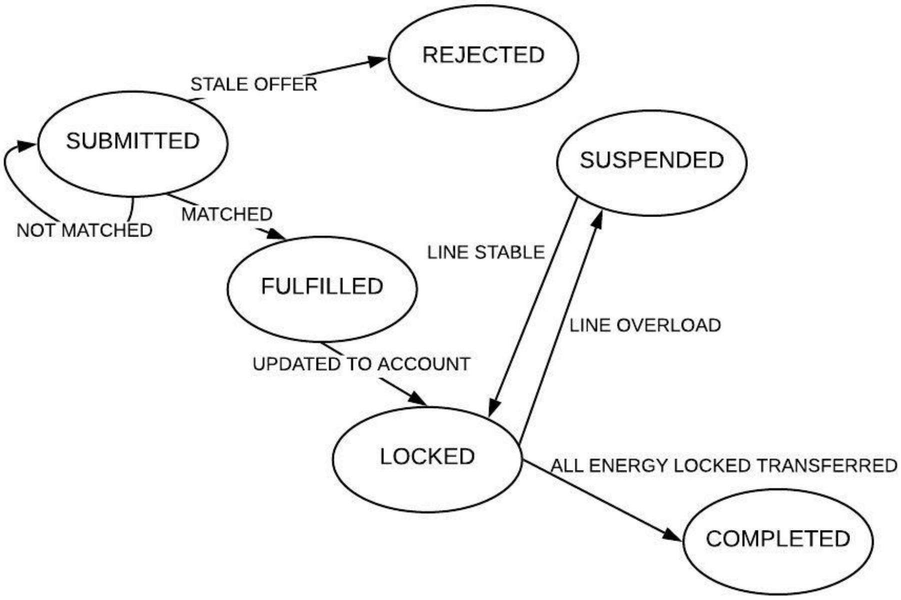

Figure 11. Status Transition Diagram for Energy Futures

Figure 11 shows the status transition diagram for Energy Futures Marketplace. Initially all the offers submitted to the marketplace have 'submitted' status. During closing time, status for stale offers is updated to 'rejected'. For the offers that are matched the status is updated to 'fulfilled'. All other offers are retained and have 'submitted' status. All fulfilled offer amounts are updated in the user accounts and the status changes to 'locked'. The offer is 'suspended' if there's an instability condition. Once all energy locked is transferred status is updated to 'completed'.

For every offer locked, the system updates *tbalance* and *ebalance* during time of delivery. After all the locked energy is supplied/consumed, offer status is updated to completed. The transaction flow for locked offers is shown in figure 12

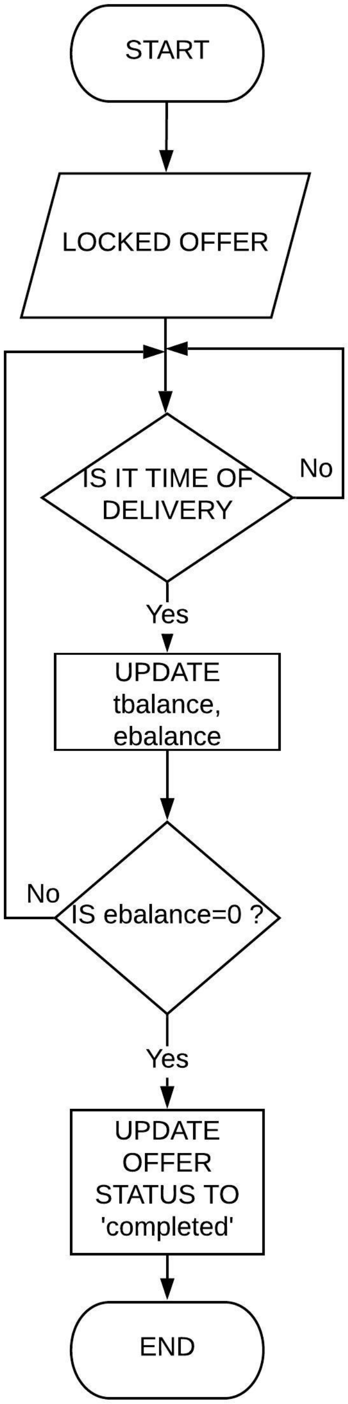

Figure 12. Offer Transaction Flow Post Locking

Chapter 4

EXPERIMENTS AND RESULTS

4.1 Experimental Setup

4.1.1 Description of Blockchain Infrastructure

Hyperledger Fabric blockchain setup for the P2P energy trading application spans over four functional units. These functional units are called organisations when referenced in the context of the hyperledger fabric application:

1. Utility Company - Controls Generation, Transmission and Distribution.
2. Solar - Photo Voltaic Energy Generators/ Operators.
3. Battery - Energy Storage Operators.
4. Consumer - Private and Corporate Energy Consumers.

One thing to remember, a participant in the real world could assume multiple of these functional roles, but they transactions and smart contracts would be bifurcated from the blockchain perspective. Each functional unit owns a portion of the infrastructure. Figure 13 shows the hyperledger fabric infrastructure consisting of the four organizations described earlier. . The labels in the figure correspond to the numbering in the description provided below. The infrastructure consists of the following components:

1. **Peers**: Peers are a fundamental element of the network because they host ledgers and smart contracts. For redundancy, resilience and reliability, every organisation

hosts five peers. These peers maintain copies of the shared distributed ledgers that the peers are part of. Peers within an organization are labeled from 0 to 4. Lets say we want to reference the first peer within utility organization, we can reference it using the following notation: *peer0.utility*.

2. **Client Peers**: Client Peers host the REST Application Programming Interface, that allows user apps to interact with the smart contracts. A client application talks to a client peer over REST or GRPC interface and submits transactions and queries to the peer.
3. **Certification Authority (CA)**: Each organisation hosts an instance of a certification authority. Every organization hosts a certificate authority(CA) which provides features such as:
    - Registration of identities
    - Issuing Enrollment Certificates(ECerts)
    - Certificate renewal and revocation
4. **CouchDB**: For a hyperledger fabric blockchain network, the current state of the ledger represents the latest values for all keys included in the chain transaction logs. This is commonly referred to as *World State*. Chaincode invocations execute transactions against the current state data. To make chaincode interactions efficient, the latest values of all keys are stored in state database, which is essentially an indexed view of the blockchain's transaction log. The state database is recovered(or generated) upon peer startup automatically. LevelDB is default database embedded in the peer process and stores chaincode data as key-value pairs. CouchDB is an optional alternative database that is provides additional query support for chaincode data modelling. In this implementation

CouchDB is hosted for every organization, which allows running rich queries on the data stored on the distributed ledger. [17]

5. **Ordering Service**: The ordering service is implemented using Apache Kafka. It consists of a Zookeeper cluster consisting of three Zookeeper nodes. On top of the Zookeeper cluster is a cluster of Kafka brokers. These Kafka brokers replicate transactions that are set in order according to timestamps by three orderer nodes.
6. **Channels**: Four channels are used to enable transactions on the energy marketplace.

Every peer has a client peer connected with it to support applications for users.

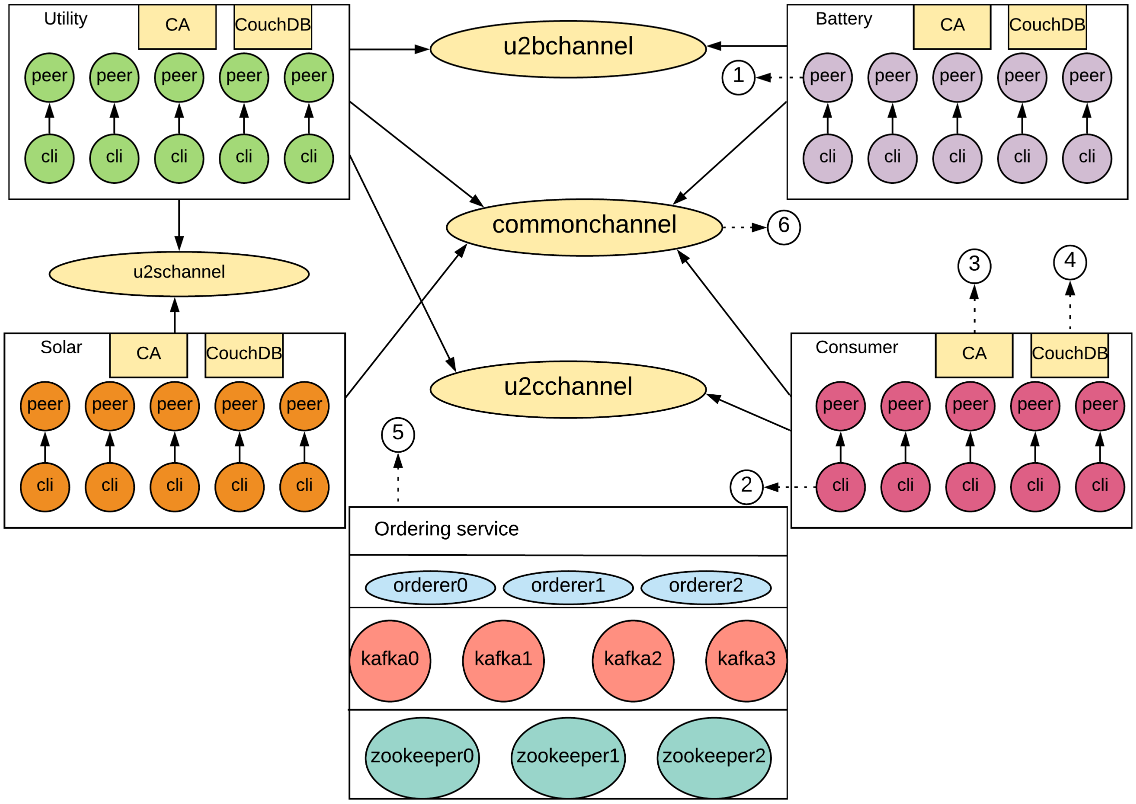

Figure 13. Hyperledger Fabric Infrastructure for Energy Marketplace

Table 1. Organisation Membership, Channel membership and Smart contract details for all organisations and member peers

Organisation	**Peers joining**	**Client Peers joining**	**Channels joined**	**Smart Contracts stored on peers**
Utility	peer0.utility, peer1.utility, peer2.utility, peer3.utility, peer4.utility	cliutility0, cliutility1, cliutility2, cliutility3, cliutility4	commonchannel, u2schannel u2cchannel, u2bchannel	buyoffer, selloffer, matchoffer, solaraccount, batteryaccount, consumeraccount
Solar	peer0.solar, peer1.solar, peer2.solar, peer3.solar, peer4.solar	clisolar0, clisolar1, clisolar2, clisolar3, clisolar4	commonchannel, u2schannel	buyoffer, selloffer, matchoffer, solaraccount
Battery	peer0.battery, peer1.battery, peer2.battery, peer3.battery, peer4.battery	clibattery0, clibattery1, clibattery2, clibattery3, clibattery4	commonchannel, u2bchannel	buyoffer, selloffer, matchoffer, batteryaccount
Consumer	peer0.consumer, peer1.consumer, peer2.consumer, peer3.consumer, peer4.consumer	cliconsumer0, cliconsumer1, cliconsumer2, cliconsumer3, cliconsumer4	commonchannel, u2cchannel	buyoffer, selloffer, matchoffer, consumeraccount

Table 1 describes the membership details for all components of the blockchain infrastructure. It also indicates smart contract storage information.

For 'Real Time Energy Transactions' model, the following endorsement policies for the four channels:

- **commonchannel**: *AND(Utility.member, Solar.member, Battery.member, Consumer.member)*
- **u2schannel**: *AND(Utility.member, Solar.member)*
- **u2bchannel**: *AND(Utility.member, Battery.member)*
- **u2cchannel**: *AND(Utility.member, Consumer.member)*

For 'Energy Futures' model, the following endorsement policies for the four channels:

- **commonchannel**: *AND(Solar.member, Battery.member, Consumer.member)*
- **u2schannel**: *AND(Utility.member, Solar.member)*
- **u2bchannel**: *AND(Utility.member, Battery.member)*
- **u2cchannel**: *AND(Utility.member, Consumer.member)*

For users of Solar, Battery and Consumer Organization, account asset would be maintained on u2schannel, u2bchannel and u2cchannel respectively. All the buyoffers and selloffers would be submitted to the commonchannel. The commonchannel also hosts the matchoffer smart contract.

Hyperledger Fabric NodeJs SDK was used to build the blockchain network. Every committing peer, client peer and ordering peers were deployed using docker containers.

4.1.2 Description of Power Systems

To test the blockchain system for performance of smart contract we needed to model a section of the power grid by taking into consideration the various actors. We use a test feeder designed by IEEE Power and Energy Society (PES) [33]. The IEEE 34 Node Bus Test Feeder was originally created in 1992 and approved by the DSA Subcommittee during the 2000 PES Summer Meeting. The system was designed to evaluate and benchmark algorithms in solving unbalanced three-phase radial systems. This feeder represents a reduced order model of an actual distribution circuit. Figure 14 illustrated the 34-bus feeder.

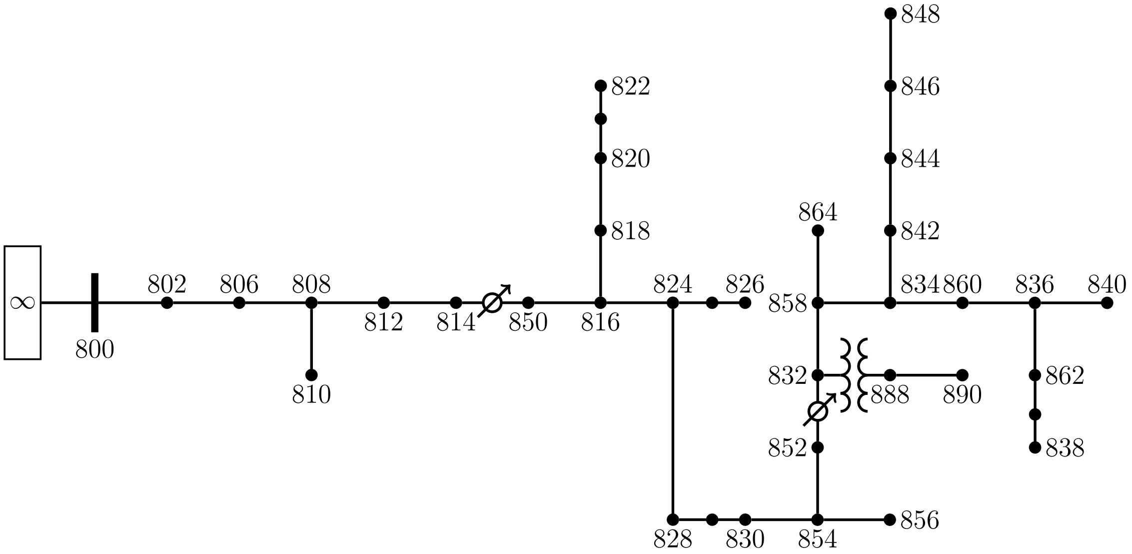

Figure 14. IEEE 34 node feeder

Table 2. Description of storage devices on the IEEE 34 Node Bus(labelled in green)

Node	**Type**	**Capacity(kWh)**
848	Storage/Battery	2000
858	Storage/Battery	1500

The 34-bus feeder is an actual feeder located in Arizona, with a nominal voltage of 24.9 kV. It is characterized by long and lightly loaded, two in-line regulators, an inline transformer for short 4.16 kV section, unbalanced loading and shunt capacitors.

For testing the blockchain network, the nodes on the bus were programmed to submit data from the Phoenix Deer Valley and Scottsdale region. Figure 15 shows the labelled diagram used for experiments. Two storage devices with storage capacity of 2000 kWh and 1500 kWh are places on nodes 848 and 858 respectively. Solar inverters are located at bus 808, 816, 824, 860 with capacity 300, 400, 500, 600 kVA respectively.

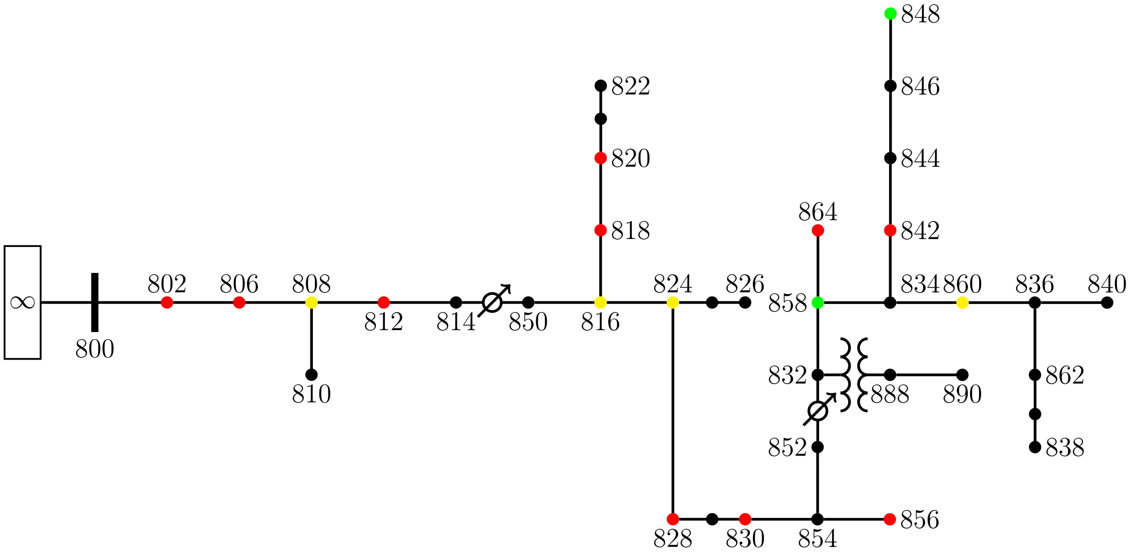

Figure 15. IEEE 34 node feeder with DER placement (yellow nodes represents nodes with solar, green nodes represent nodes with storage, red nodes represent load nodes)

Table 3. Description of load nodes on the IEEE 34 Node Bus(labelled in red)

Node No.	**Type**	**Location**
802	Full Service Restaurant	Phoenix Deer Valley
806	Large Office	Phoenix Deer Valley
812	Hospital	Scottsdale
818	SuperMarket	Scottsdale
820	Midrise Apartment	Phoenix Deer Valley
828	Small Hotel	Scottsdale
830	Large Office	Phoenix Deer Valley
842	Stand Alone Retail	Scottsdale
856	Large Hotel	Scottsdale
864	Warehouse	Phoenix Deer Valley

Table 4. Description of solar inverter nodes on the IEEE 34 Node Bus(labelled in yellow)

Node	**Type**	**Capacity(kVA)**
808	Solar Inverter	300
816	Solar Inverter	400
824	Solar Inverter	500
860	Solar Inverter	600

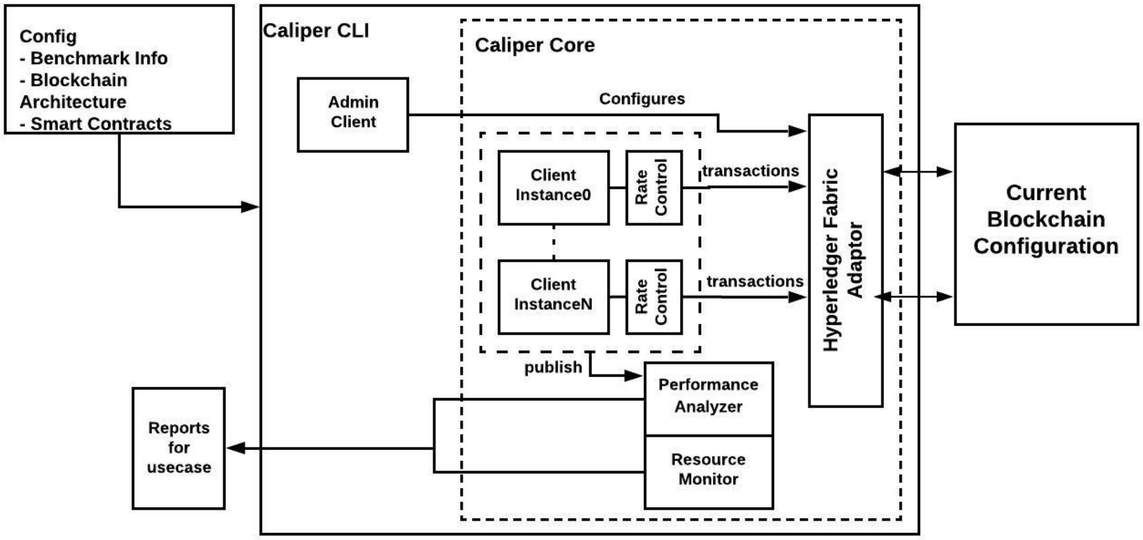

Figure 16. Architecture Diagram For Hyperledger Caliper

The feeder described is used to submit transactions to the system under test and evaluate its performance under varying transaction loads.

4.1.3 Description of Benchmarking Technology: Hyperledger Caliper

Hyperledger Caliper is an open-source blockchain performance framework, which allows users to test different blockchain solutions with predefined use cases and configurations. It can be used for comparative performance studies across different blockchain technologies, performance testing for smart contracts and discovering resource constraints for test loads. Figure 16 shows the architecture for Hyperledger Caliper.

The configuration includes information about the benchmark to be run, the blockchain architecture to be tested and the code for the smart contracts. These are fed to the interface, which creates two types of clients. The Admin client configures the system under test. This client is used to create channels, allow peers to join channels and deploy chaincode. Multiple worker clients are also created which drive the test load. These clients are driven using a rate controller, which impacts the rate at which transactions are submitted. The responses from these transactions are published to a performance analyzer. Resource monitor keeps track of the memory usage, disk I/Os and CPU usage. All of these statistics are then compiled into a report.

4.2 Description of System Under Test Configuration

Table 5 contains all the parameter values that remain constant across all tests. Table 6 indicates the number of peers that join all the channels. The first phase of experiments involves performance testing the blockchain network using the following configurable parameters:

Table 5. Global Parameters for the System Under Test

Parameter Name	**Value**
Max Block Size	98 MB
Number of Peers	20
Number of Client Peers	20
Number of Zookeeper Nodes	3
Number of Kafka Brokers	4
Number of Orderer Nodes	3
System Memory (Total)	236 GB
OS	Ubuntu 16.04.3 LTS

Table 6. Channel Peer Membership Information

Channel Name	**Number of Peers joining**
commonchannel	20
u2schannel	10
u2bchannel	10
u2cchannel	10

- **MaxMessageCount**: Maximum number of Transactions allowed in a block

- **Preferred Max Bytes**: This is the size of the blocks that are replicated across the network.
- **Send Rate**: Rate at which transactions are send as input to the blockchain network during the experiment. Every experiment has multiple rounds which with increasing send rates and performance metrics are recorded and reported for each round.

The performance metrics for the first set of experiments can be described as follows:

- **Max Latency**: Worst Case Transaction Response Time (measured in seconds).
- **Min Latency**: Best Case Transaction Response Time (measured in seconds).
- **Avg Latency**: Mean Transaction Response Time (measured in seconds).
- **Throughput**: Transactions Per Second processed by the system.
- **Successful Transactions**: Percentage of Transactions that were successful.
- **Failed Transaction**: Percentage of Transactions that were unsuccessful.

Table 7 represents the block sizes used for performance testing. For result discussion purposes the these are divided into 2 classes, *Small Block sizes*(512 kB, 1 MB, 2 MB) and *Large Block Sizes*(4 MB, 8 MB, 16 MB). Table 8 indicates the Message Batch Sizes used for performance testing. As shown in the table, a batch size of 500 is only used when testing *Large Block Sizes.* Sections 4.3 and 4.4 contain experimental details, results and discussion for testing the underlying blockchain infrastructure using various block sizes. The experimental setup described in section 4.1 is used to feed transactions to the blockchain network in all the experiments. Section 4.5 compares *Real Time Energy Transactions* model to *Energy Futures* model and discusses the results for smart contract response time for offer matching and settlement.

Table 7. List of Blocksizes Tested

Blocksizes Tested
512 KB
1 MB
2 MB
4 MB
8 MB
16 MB

Table 8. List of Message Batch Tested

Message batch sizes Tested
50
100
200
250
500 (For Block Sizes 4MB, 8 MB and 16 MB)

4.3 Performance Testing Results Using Small Block Sizes

4.3.1 Testing 512 KB Blocks

4.3.1.1 Adding Assets To The Blockchain

For these set of experiments there are three rounds for different send rates which can be described as follows:

- **Round 0** : 20 tps
- **Round 1** : 25 tps
- **Round 2** : 30 tps

Table 9 contains the reports for adding assets using 512 kB blocks.

Table 9. Results for adding assets on the blockchain using 512 kB blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	-	-	-	0	0	100
	round 1	-	-	-	0	0	100
	round 2	-	-	-	0	0	100
100	round 0	3.14	0.94	2.07	14.9	100	0
	round 1	21.78	15.42	18.56	4.6	100	0
	round 2	3.54	1.51	2.46	20	100	0
200	round 0	5.26	0.97	2.35	18.8	100	0
	round 1	23.79	3.58	13.17	19.6	100	0
	round 2	3.81	1.58	2.66	27.6	100	0
250	round 0	6.95	1.01	2.71	18.9	100	0
	round 1	24.36	3.31	13.55	19.1	100	0
	round 2	3.82	1.52	2.62	26.6	100	0

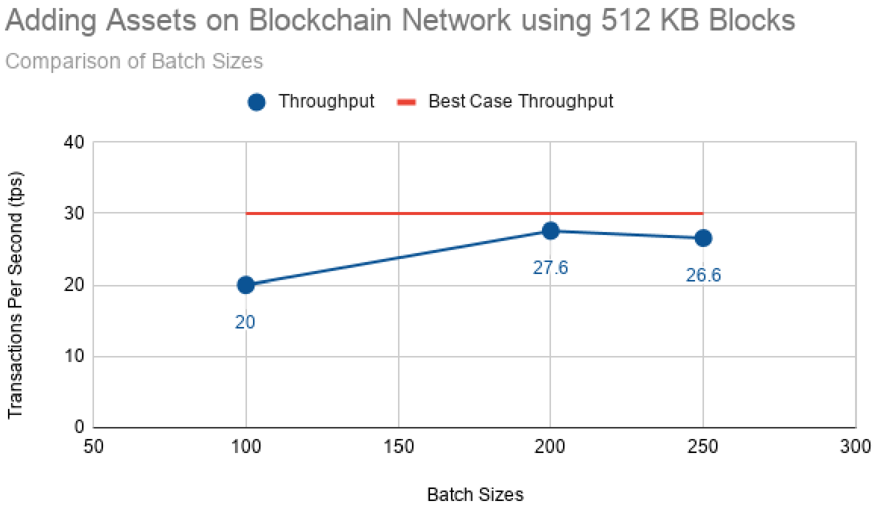

Figure 17. Throughput plot for various message batch sizes tested for 512 KB Blocks

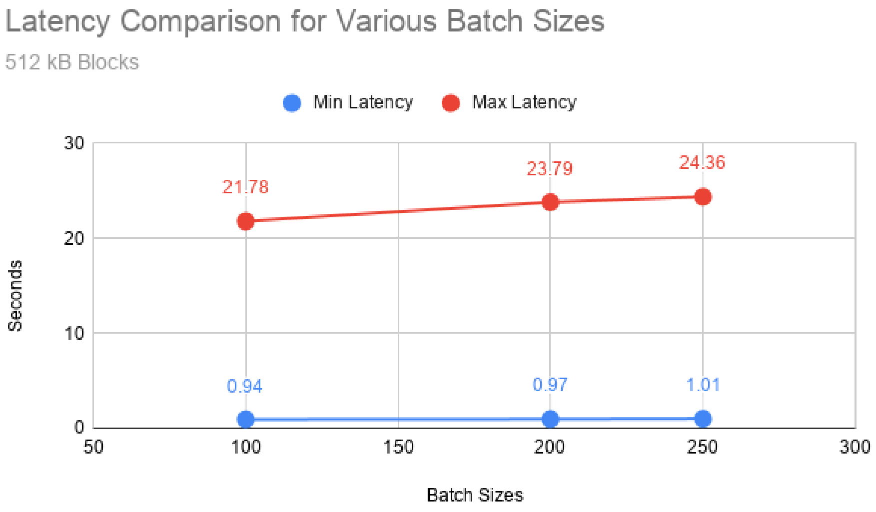

Figure 18. Latency Comparison plot for various message batch sizes tested for 512 KB Blocks

4.3.1.2 Querying The Ledger

For these set of experiments there are five rounds for different send rates which can be described as follows:

- **Round 0** : 25 tps
- **Round 1** : 50 tps
- **Round 2** : 100 tps
- **Round 3** : 200 tps
- **Round 4** : 250 tps

Table 10 presents the results for this set of experiments.

Table 10. Results For Querying The Ledger Containing 512 kB Blocks

Batch Size	Round No.	Max Latency	Min Latency	Avg Latency	Throughput	Success %	Failed %
50	round 0	2.1	0.12	1.11	25	100	0
	round 1	1.21	0.19	0.69	49.8	100	0
	round 2	0.78	0.17	0.5	98.6	100	0
	round 3	5.72	0.42	3.47	132.5	100	0
	round 4	5.4	0.62	3.65	146.1	100	0
100	round 0	2.35	0.17	1.23	24.8	100	0
	round 1	2.31	0.27	1.3	49.5	100	0
	round 2	1.41	0.25	0.87	97.8	100	0
	round 3	4.12	0.56	2.4	147.1	100	0
	round 4	4.74	1.05	3.18	153.5	100	0
200	round 0	13.52	0.18	3.17	24.9	100	0
	round 1	2.51	0.28	1.36	47.9	100	0
	round 2	2.62	0.42	1.56	96.1	100	0
	round 3	3.49	0.82	1.87	162.4	100	0
	round 4	4.05	1.16	2.78	132.2	100	0
250	round 0	2.37	0.16	1.22	24.9	100	0
	round 1	2.46	0.3	1.39	47.6	100	0
	round 2	2.69	0.53	1.63	95.2	100	0
	round 3	3.35	1.06	2.07	162.9	100	0
	round 4	4.21	1.77	2.81	169.4	100	0

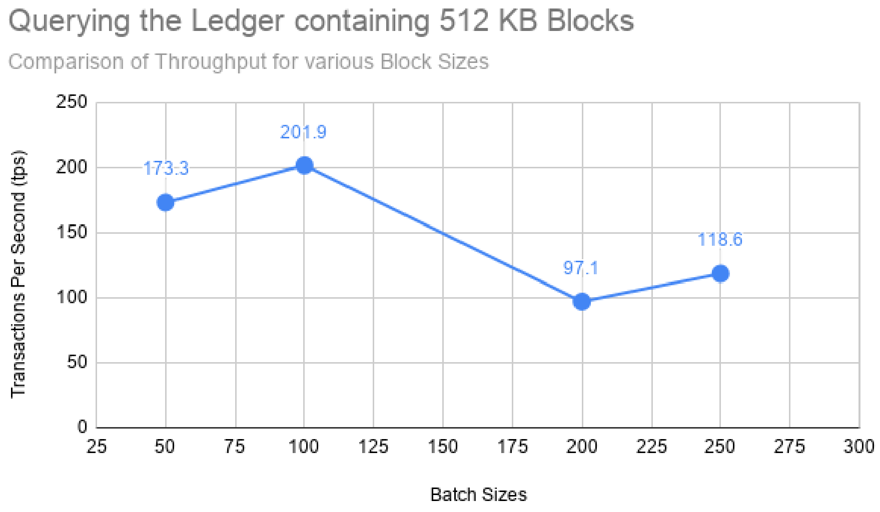

Figure 19. Throughput plot for various message batch sizes tested for querying 512 KB Blocks

4.3.1.3 Results Discussion

- The main intention behind testing *small block sizes*(512 kB, 1 MB, 2 MB) is to observe the performance of the blockchain system with blocks that can be replicated faster across the network of peers. Figure 17 represents the plot for throughput for adding assets onto the blockchain network while varying batch sizes. Figure 18 is the comparison plot for minimum and maximum latency for transactions involving adding assets onto the blockchain. Figure 19 is the throughput plot for processing queries for a ledger containing 512 kB blocks with varying message batch sizes.
- Adding assets to the blockchain involves writing data as well as replicating it on all peers of the blockchain. Querying the blockchain involves reading data from the blockchain. Adding assets is a relatively expensive operation compared to querying the ledger.
- For the blockchain configured with 512 kB blocks, while adding assets, all rounds fail when using a message batch size lower than 50. A low message batch size is not able to support expensive operations on the blockchain. However such a blockchain system is able to support query operations producing satisfactory throughput(173.2 tps).
- The operation of adding assets also has high failure rate when the send rate is higher than 30 tps.
- For both adding and querying operations, there is a fall in performance observed as the message batch size is increased. This is also accompanied by an increase in the maximum latency for processing transactions. The main reason for this is that a 512 kB block will not be able to hold high number of transactions

and as batches keep piling up, there is a delay in processing further incoming transactions.

- Considering all the factors, viz, high failure rate and high latency for expensive operations, 512 kB Blocks aren't ideal for supporting a production grade system.

4.3.2 Testing 1 MB Blocks

4.3.2.1 Adding Assets To The Blockchain

For this set of experiments four rounds were used with varying send rates which can be described as follows:

- **Round 0**: 20 tps
- **Round 1**: 25 tps
- **Round 2**: 30 tps
- **Round 3**: 50 tps

Table 11 describes the results for various rounds for adding assets on the blockchain network using 1 MB Blocks.

Table 11. Results For Adding Assets On The Blockchain Using 1MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	3.48	1.06	2.25	18.8	100	0
	round 1	3.33	1.22	2.27	23.7	100	0
	round 2	5.9	1.22	2.39	23.9	100	0
	round 3	5.77	1.34	3.19	39.9	100	0
100	round 0	4.62	0.89	2.18	19	100	0
	round 1	23.25	3.78	13.28	19.5	100	0
	round 2	3.77	1.51	2.59	27.7	100	0
	round 3	7.68	3.02	4.79	36.7	100	0
200	round 0	4.43	1.08	2.5	18.6	100	0
	round 1	23.23	2.34	12.81	19.3	100	0
	round 2	3.65	1.45	2.58	26.6	100	0
	round 3	8.62	2.87	5.37	33.1	100	0
250	round 0	4.35	0.93	2.32	19.2	100	0
	round 1	23.29	4.47	13.82	19.6	100	0
	round 2	3.76	1.5	2.6	27.7	100	0
	round 3	8.28	3.21	5.74	33.6	100	0

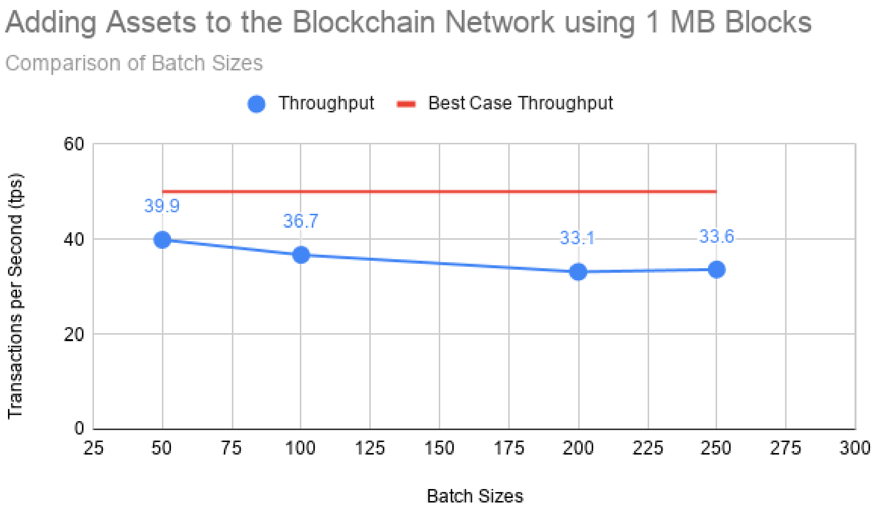

Figure 20. Throughput plot for various message batch sizes tested for 1 MB Blocks

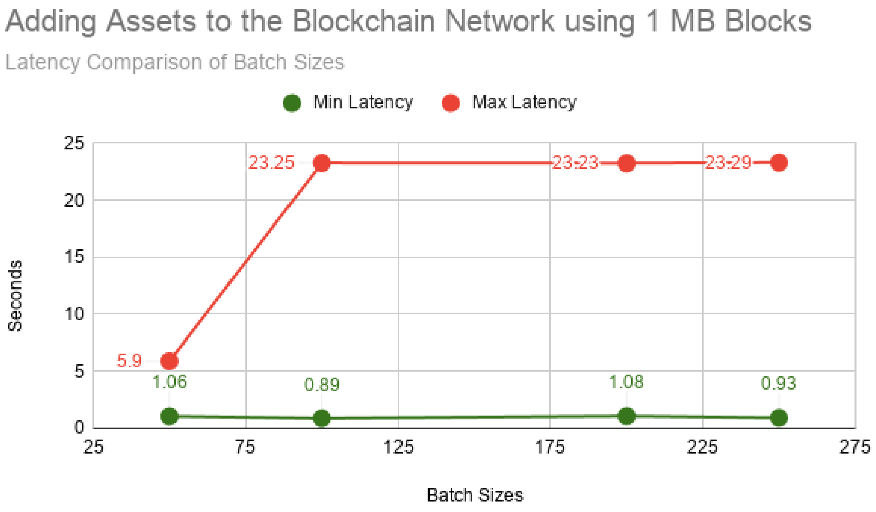

Figure 21. Latency Comparison for various message batch sizes tested for 1 MB Blocks

4.3.2.2 Querying The Ledger

For this set of experiments five rounds were used with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 250 tps

Table 12 describes the experiment results for querying assets containing 1 MB Blocks.

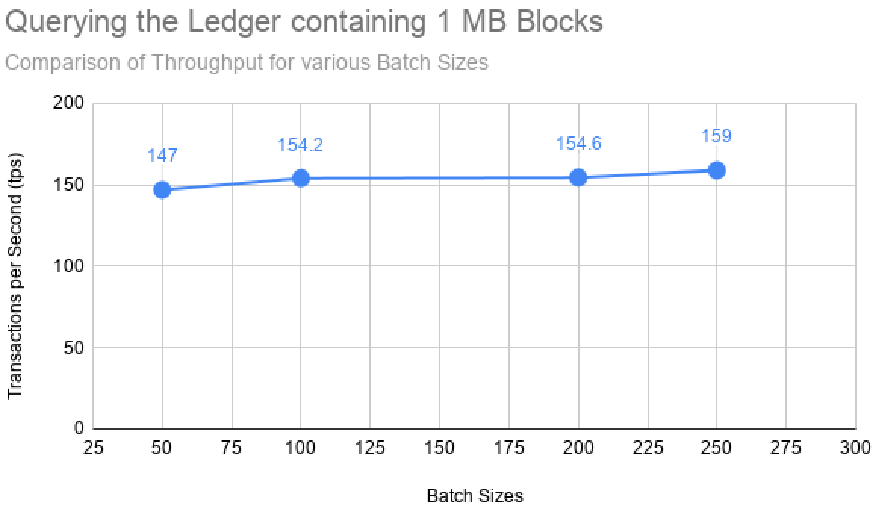

Figure 22. Throughput plot for various message batch sizes tested for 1 MB Blocks

Table 12. Results For Querying The Ledger Containing 1 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	2.1	0.13	1.11	25	100	0
	round 1	1.16	0.16	0.66	49.7	100	0
	round 2	0.79	0.16	0.5	98.7	100	0
	round 3	4.99	0.43	2.83	147	100	0
	round 4	3.91	0.7	1.92	44	25	75
100	round 0	2.36	0.15	1.22	24.2	100	0
	round 1	2.26	0.21	1.26	49.6	100	0
	round 2	1.39	0.22	0.86	97.5	100	0
	round 3	3.5	0.43	1.95	154.2	100	0
	round 4	4.55	0.99	3.12	151.6	100	0
200	round 0	2.48	0.17	1.26	23.9	100	0
	round 1	2.48	0.27	1.38	47.6	100	0
	round 2	2.35	0.32	1.3	81.6	100	0
	round 3	3.33	0.89	2.14	153.5	100	0
	round 4	4.68	1.4	3.07	154.6	100	0
250	round 0	2.35	0.16	1.22	25	100	0
	round 1	2.52	0.32	1.39	47.7	100	0
	round 2	2.45	0.25	1.25	97.2	100	0
	round 3	2.97	0.77	1.81	151.5	100	0
	round 4	4.04	1.38	2.7	159	100	0

4.3.2.3 Results Discussion

- Configuring the blockchain with 1 MB Blocks allows more transactions to be stored in a block. Figure 20 shows the plot for throughput when adding assets onto the blockchain network for various batch sizes. Figure 21 shows the comparison of minimum latency vs maximum latency for adding assets onto the blockchain using 1 MB Blocks for varying batch sizes. Figure 22 represents the throughput plot for querying the ledger containing 1 MB Blocks for various batch sizes.

- For this system configuration the throughput results for addition of assets are similar to that of 512 kB blocks. The throughput of the system decreases as we increase the message batch size. However the failure rate in this system configuration is very low.
- For processing queries, the throughput increases slightly as the message batch size is increased. However the max throughput achieved for querying using 1 MB Blocks (159 tps) is lower than that achieved using 512 kB Blocks (201.9 tps).
- With increasing message batch sizes, the maximum latency for processing transactions remains constant (approx 23.25 s), except for a message batch size of 50. This again points to the fact that a larger block size would be able to allow more transactions, and hence reduce the latency.

4.3.3 Testing 2 MB Blocks

4.3.3.1 Adding Assets To The Blockchain Network

For this set of experiments five rounds were used with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

Table 13 reports the results for the experiments involving adding assets to the blockchain network configured to use 2 MB Blocks.

Table 13. Results For Adding Assets To Blockchain Network using 2 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	24.23	1.17	8.57	23.8	100	0
	round 1	6.12	1.67	3.72	45.4	100	0
	round 2	12.13	1.97	8.23	57.3	100	0
	round 3	13.26	1.94	10.43	62.9	100	0
	round 4	11.36	2.9	8.94	26.4	40	60
100	round 0	23.8	1.36	8.76	24.1	100	0
	round 1	9.55	2.14	5.32	42.1	100	0
	round 2	13.68	3.76	9.17	58.2	100	0
	round 3	12.45	5.76	10.48	64.6	100	0
	round 4	14.35	5.03	12.04	64.2	100	0
200	round 0	26.06	1.27	5.99	24.5	100	0
	round 1	14.75	2.87	8.52	42.6	100	0
	round 2	25.57	7.6	17.69	57.2	100	0
	round 3	27.05	8.5	20.54	67.1	100	0
	round 4	27.41	8.85	22.9	68.3	100	0
250	round 0	26.21	1.28	6.05	24.5	100	0
	round 1	15.25	2.58	9.22	42.4	100	0
	round 2	26.47	6.46	17.88	55.8	100	0
	round 3	26.92	7.58	18.23	52.6	77.55	22.45
	round 4	27.05	8.55	21.21	59.1	86.35	13.65

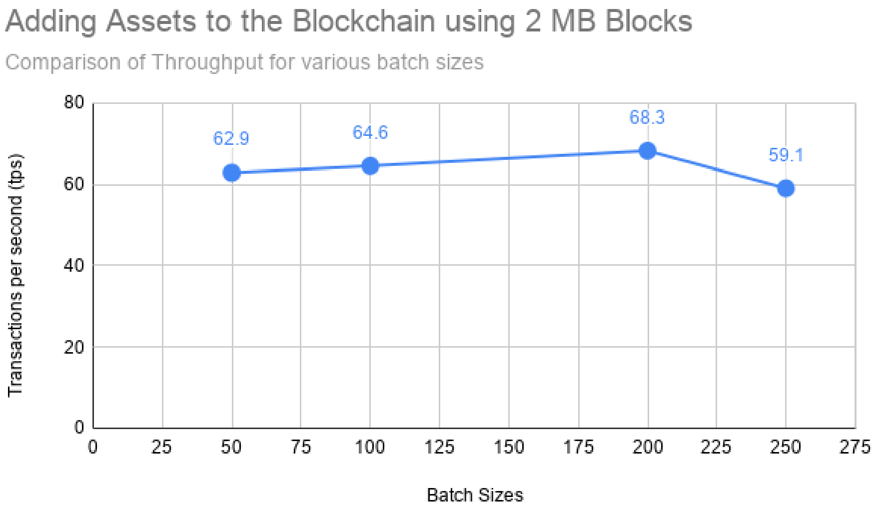

Figure 23. Throughput plot for various message batch sizes tested for 2 MB Blocks

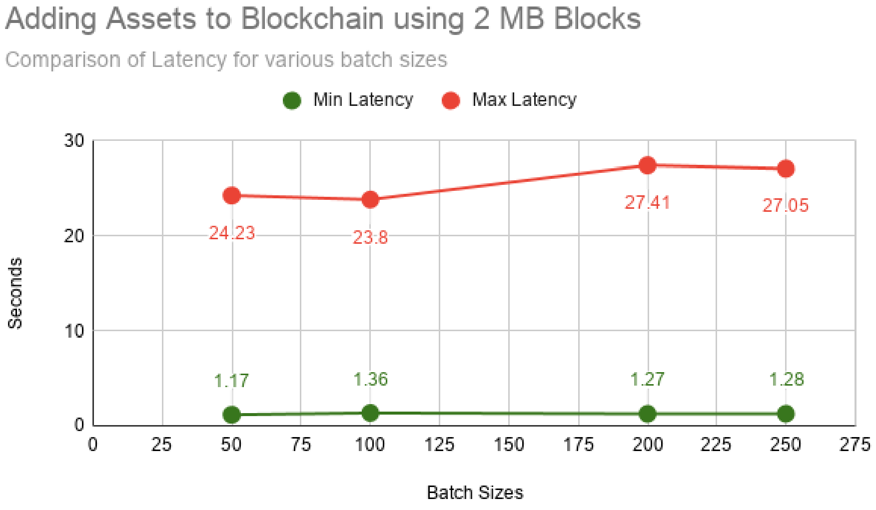

Figure 24. Latency Comparison for various message batch sizes tested for 2 MB Blocks

4.3.3.2 Querying the Ledger

For this set of experiments five rounds were with varying send rates were used which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 250 tps

Table 14 contains the results for experiments involving querying the ledger containing 2 MB blocks.

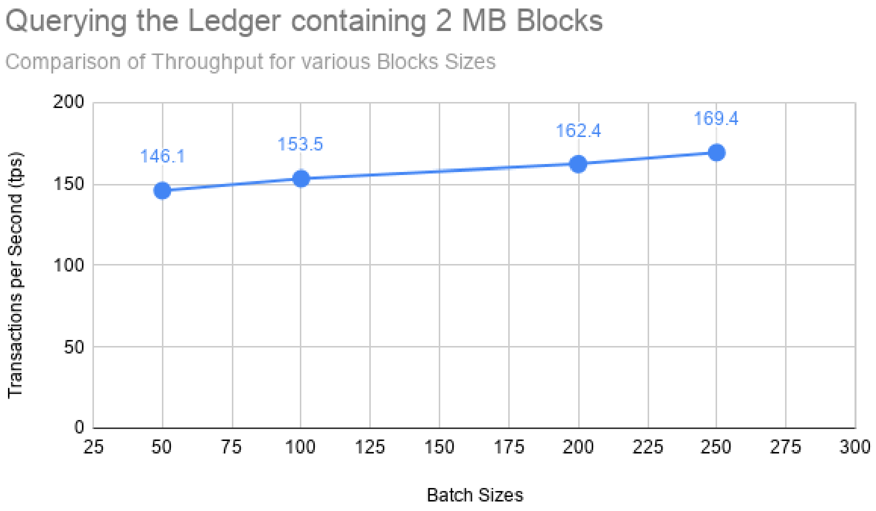

Figure 25. Throughput plot for various message batch sizes tested for 2 MB Blocks

Table 14. Results For Querying Ledger Containing 2 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	2.1	0.12	1.11	25	100	0
	round 1	1.21	0.19	0.69	49.8	100	0
	round 2	0.78	0.17	0.5	98.6	100	0
	round 3	5.72	0.42	3.47	132.5	100	0
	round 4	5.4	0.62	3.65	146.1	100	0
100	round 0	2.35	0.17	1.23	24.8	100	0
	round 1	2.31	0.27	1.3	49.5	100	0
	round 2	1.41	0.25	0.87	97.8	100	0
	round 3	4.12	0.56	2.4	147.1	100	0
	round 4	4.74	1.05	3.18	153.5	100	0
200	round 0	13.52	0.18	3.17	24.9	100	0
	round 1	2.51	0.28	1.36	47.9	100	0
	round 2	2.62	0.42	1.56	96.1	100	0
	round 3	3.49	0.82	1.87	162.4	100	0
	round 4	4.05	1.16	2.78	132.2	100	0
250	round 0	2.37	0.16	1.22	24.9	100	0
	round 1	2.46	0.3	1.39	47.6	100	0
	round 2	2.69	0.53	1.63	95.2	100	0
	round 3	3.35	1.06	2.07	162.9	100	0
	round 4	4.21	1.77	2.81	169.4	100	0

4.3.3.3 Results Discussion

- The last rounds of experiments with *small block sizes* involved testing the blockchain configured with 2 MB Blocks. Figure 23 shows the throughput plot for adding blocks onto the blockchain network using 2 MB blocks for varying message batch sizes. Figure 24 shows the comparison of minimum latency vs maximum latency for adding assets using different batch sizes. Figure 25 represents the plot for throughput for querying the ledger containing 2 MB Blocks for various message batch sizes.

- For adding assets, we observe that the throughput increases steadily until there is a slight fall for message batch size of 250. The reason for this fall is that a 2 MB block is overwhelmed and is not able to process transactions involving expensive operations due to limitations. This can also be justified by a failure rate of 22.45 % and 13.65 % in the last 2 rounds of experiments for batch size 250.
- The throughput of the system for handling queries also increases steadily for an increase in message batch sizes. The system achieves a maximum throughput of 169.4 tps which is higher than the one achieved using 1 MB Blocks.
- Most public blockchains make use of block sizes lesser than or equal to 1 MB. The Bitcoin blockchain uses 1 MB blocks currently and is able to achieve around 4.2 transactions per second due to its design constraints. The Ethereum blockchain, although not constrained by block-size constraints, can still reach about 15 transactions per seconds. The current system under test is able to beat most public blockchains in terms of performance with similar block sizes.

4.4 Performance Testing Results Using Large Block Sizes

4.4.1 Testing 4 MB Blocks

4.4.1.1 Adding Assets To The Blockchain Network

For this set of experiments five rounds are performed, with varying send rate which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

Table 15 contains the results pertaining to experiments for adding assets to the blockchain network using 4 MB Blocks.

Table 15. Results For Adding Assets To Blockchain Using 4 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	25.74	1.18	8.76	24	100	0
	round 1	6.94	1.51	3.84	44.1	100	0
	round 2	11.85	1.38	7.94	61.7	100	0
	round 3	13.27	2.32	10.06	65.3	100	0
	round 4	11.61	3.18	8.63	19.5	30	70
100	round 0	26.52	1.3	9.3	24.2	100	0
	round 1	7.54	2.69	4.8	42.7	100	0
	round 2	14.3	3.16	9.45	56.7	100	0
	round 3	14.43	4.69	10.87	63.4	100	0
	round 4	14.26	4.67	11.73	65.2	100	0
200	round 0	26.45	1.26	5.97	24.5	100	0
	round 1	13.05	2.78	7.87	44.7	100	0
	round 2	25.6	7.83	16.55	59	100	0
	round 3	28.89	7.35	21.55	63	100	0
	round 4	26.88	9.58	22.95	68.8	100	0
250	round 0	27.94	1.26	6.55	24.5	100	0
	round 1	14.59	2.74	8.81	42	100	0
	round 2	26.43	6.49	17.62	54.2	100	0
	round 3	27.22	7.65	19.78	58.6	87.95	12.05
	round 4	28.3	10.78	24.05	66.4	100	0
500	round 0	25.93	1.29	5.85	24.6	100	0
	round 1	16.28	2.06	8.86	40.6	100	0
	round 2	28.05	5.74	19.91	51.2	100	0
	round 3	30.01	12.88	22.79	43.5	79.2	20.8
	round 4	29.52	12.63	20.59	34.4	58.6	41.4

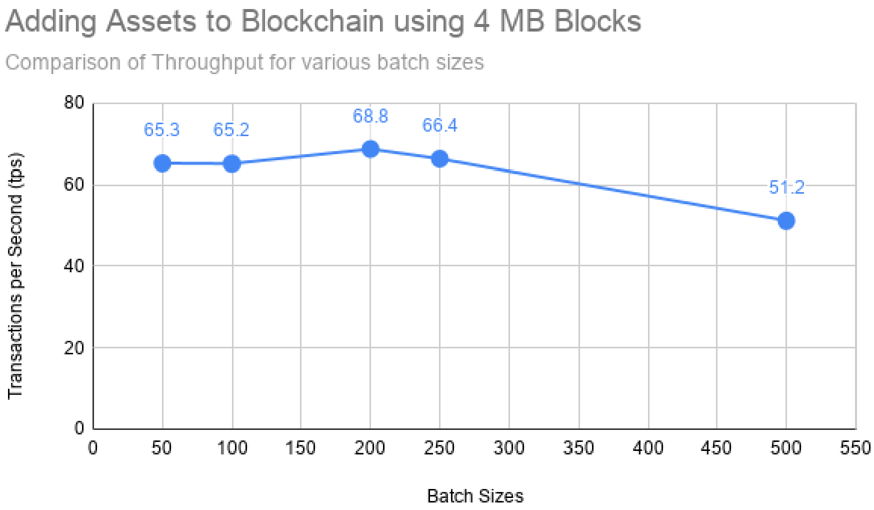

Figure 26. Throughput plot for various message batch sizes tested for 1 MB Blocks

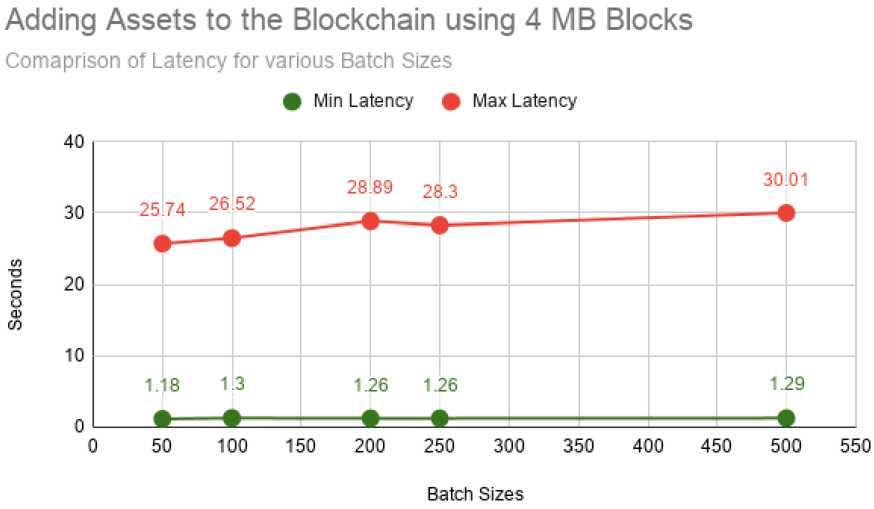

Figure 27. Latency Comparison for various message batch sizes tested for 4 MB Blocks

4.4.1.2 Querying the Ledger

For this set of experiments queries are sent over five rounds with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

Table 16 describes the results for various batch sizes for querying the ledger containing 4 MB Blocks.

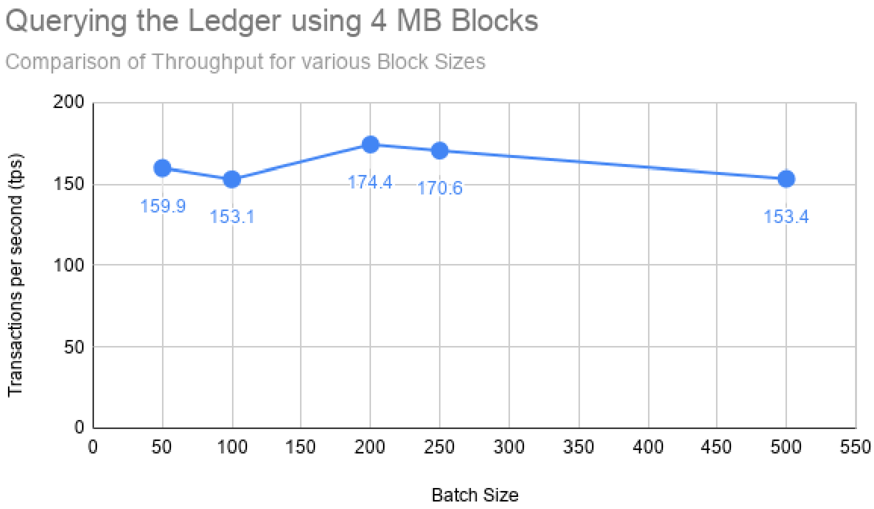

Figure 28. Throughput plot for various message batch sizes tested for 4 MB Blocks

Table 16. Results for Querying The Ledger Containing 4 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	2.07	0.13	1.1	25	100	0
	round 1	1.21	0.17	0.68	49.9	100	0
	round 2	0.8	0.19	0.51	99.2	100	0
	round 3	10.04	0.53	5.94	148	100	0
	round 4	9.08	1.26	7.28	159.9	88.45	11.55
100	round 0	19.66	0.17	3.43	24.4	100	0
	round 1	2.3	0.28	1.28	49.7	100	0
	round 2	1.47	0.25	0.89	98.9	100	0
	round 3	9.32	0.77	5.17	149.1	100	0
	round 4	9.97	6.84	8.32	153.1	91.85	8.15
200	round 0	2.37	0.16	1.23	24.7	100	0
	round 1	2.52	0.26	1.37	47.6	100	0
	round 2	2.61	0.41	1.57	98.1	100	0
	round 3	4.54	1	2.3	174.4	100	0
	round 4	8.64	5.68	7.18	123	68.85	31.15
250	round 0	2.4	0.17	1.23	24.9	100	0
	round 1	2.55	0.29	1.37	47.6	100	0
	round 2	2.7	0.57	1.63	97.3	100	0
	round 3	5.17	1.18	3.01	170.6	100	0
	round 4	9.41	5.89	7.96	166.3	100	0
500	round 0	15.16	0.17	2.5	25	100	0
	round 1	2.48	0.29	1.3	48.1	100	0
	round 2	2.94	0.59	1.69	94.8	100	0
	round 3	5.72	1.74	3.58	153.4	100	0
	round 4	9.39	5.17	7.34	141.9	76.5	23.5

4.4.2 Results Discussion

- Hyperledger Fabric, being a permissioned blockchain, lets us configure and test the system for *large block sizes* which are not used in production by most public blockchains.

- With *large block sizes*, there is exists trade-off of throughput vs latency. If the blocks are not replicated on all peers within the right amount of time, it leads to commit errors which cause the corresponding transactions in the block to fail.
- Figure 26 represents the plot for adding assets onto the blockchain using 4 MB Blocks for various batch sizes. Figure 27 shows the plot for comparison of minimum latency vs maximum latency for adding assets to the blockchain using 4 MB Blocks. Figure 16 represents the plot for handling queries for a ledger containing 4 MB Blocks for various message batch sizes. For *large block sizes*, an additional message batch size of 500 is used to test the performance of the blockchain.
- The throughput for adding assets onto the blockchain stays steady (approx 66.4 tps) for most batch sizes, with the exception of batch size of 500 where it dips to 51.2 tps. This dip is due to the block size not being adequate to serve a batch of 500 transactions arriving at a high rate.
- A similar drop is observed in throughput for processing queries when the batch size is increased to 500 (throughput drops from 170.6 tps to 153.4 tps).
- Latter rounds involving batch sizes of 250 and 500 also have transaction failure rates of 12.5 % and 41.4 % which is also evident by the high latency statistics.

4.4.3 Testing 8 MB Blocks

4.4.3.1 Adding Assets To The Blockchain Network

For this set of experiments transactions are sent over five rounds with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

Table 17 contains the results for experiments involving adding assets to the blockchain network using 8 MB Blocks.

Table 17. Results For Adding Assets to the Blockchain Network using 8 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	27.61	1.27	10.08	24.2	100	0
	round 1	6.92	1.39	4.29	45.5	100	0
	round 2	11.66	2.02	8.2	60.3	100	0
	round 3	12.11	2.39	9.79	66	100	0
	round 4	14.72	3.52	12.43	63.3	100	0
100	round 0	27.02	1.36	9.15	22.3	100	0
	round 1	8.82	2.31	5.47	43.1	100	0
	round 2	14.67	3.17	9.26	55.3	100	0
	round 3	14.09	3.52	10.79	64.6	100	0
	round 4	13.9	5.07	11.76	67.5	100	0
200	round 0	7.13	1.32	2.59	24.5	100	0
	round 1	14.63	2.18	8.15	43.8	100	0
	round 2	26.5	6.06	17.38	59.6	100	0
	round 3	27.25	7.43	19.98	67	100	0
	round 4	23.98	7.71	19.24	64.9	82.9	17.1
250	round 0	28.04	1.19	6.33	24.4	100	0
	round 1	14.48	2.88	9.11	42.5	100	0
	round 2	22.83	5.39	16.56	61.8	100	0
	round 3	29.41	9.09	22.31	62.1	100	0
	round 4	27.46	10.4	22.76	67.6	100	0
500	round 0	25.86	1.24	5.93	24.4	100	0
	round 1	20.96	2.92	10.38	44.8	100	0
	round 2	27.47	6.09	18.64	54.6	100	0
	round 3	30	8.47	22.45	59.1	98.3	1.7
	round 4	24.92	11.82	21.52	74.1	100	0

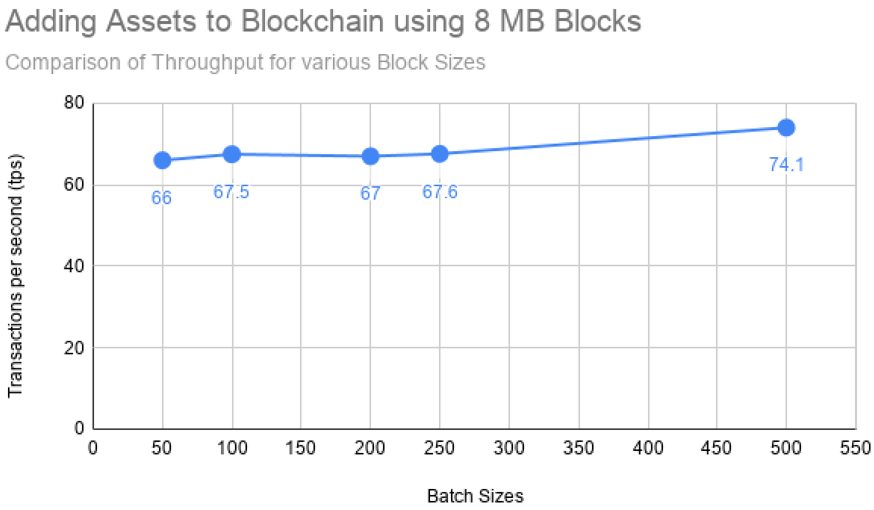

Figure 29. Throughput plot for various message batch sizes tested for 8 MB Blocks

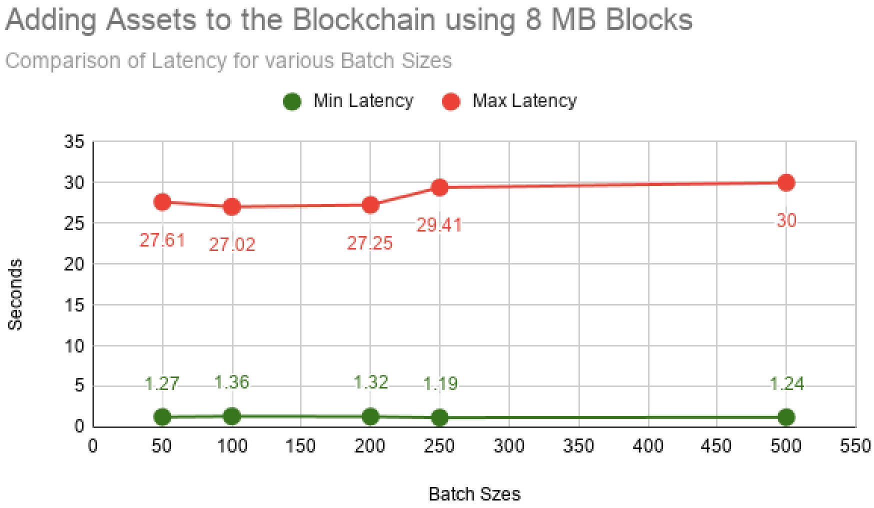

Figure 30. Latency Comparison for various message batch sizes tested for 8 MB Blocks

4.4.3.2 Querying The Ledger

For this set of experiments transactions are sent over five rounds with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

Table 16 describes the results for various batch sizes for querying the ledger containing 8 MB Blocks.

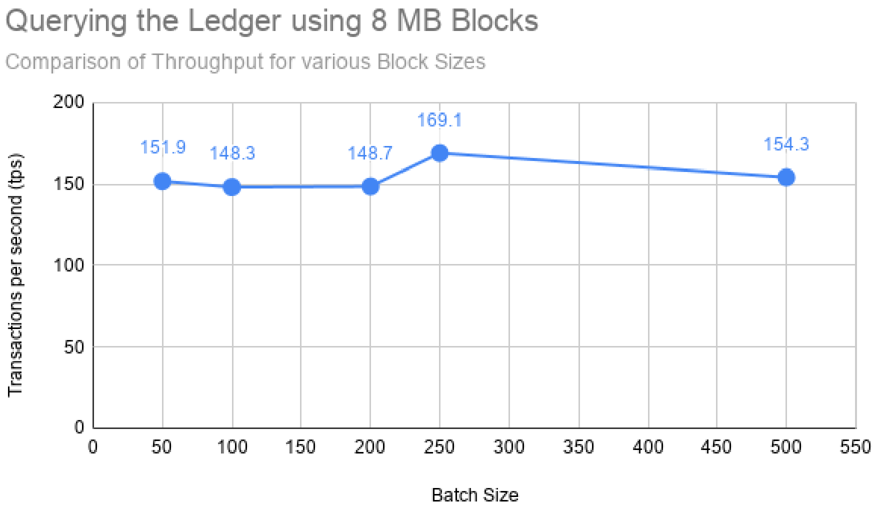

Figure 31. Throughput plot for various message batch sizes tested for 8 MB Blocks

Table 18. Results For Querying The Ledger Containing 8 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	2.11	0.15	1.11	25	100	0
	round 1	1.2	0.17	0.68	49.9	100	0
	round 2	0.77	0.2	0.5	99.2	100	0
	round 3	10.12	0.63	5.94	151.9	100	0
	round 4	9.87	4.42	8.37	123.1	71.25	28.75
100	round 0	2.39	0.17	1.25	25	100	0
	round 1	2.29	0.25	1.27	49.8	100	0
	round 2	1.45	0.24	0.88	98.9	100	0
	round 3	9.47	0.59	5.36	148.3	100	0
	round 4	9.61	3.79	8.25	46.9	25	75
200	round 0	2.37	0.14	1.23	24.4	100	0
	round 1	2.48	0.29	1.37	47.8	100	0
	round 2	2.64	0.39	1.6	98.2	100	0
	round 3	5.76	0.94	2.9	148.7	100	0
	round 4	9.22	5.36	7.58	101.7	53.05	46.95
250	round 0	2.35	0.15	1.22	24.9	100	0
	round 1	2.56	0.29	1.38	47.8	100	0
	round 2	2.94	0.59	1.69	95.1	100	0
	round 3	5.85	1.17	3.08	169.1	100	0
	round 4	9.62	6.29	8.03	145.1	85.35	14.65
500	round 0	2.35	0.17	1.24	24.4	100	0
	round 1	2.54	0.31	1.38	47.6	100	0
	round 2	2.86	0.58	1.64	96.3	100	0
	round 3	5.43	1.45	3.35	154.3	100	0
	round 4	11.29	8.21	9.98	146.8	100	0

4.4.3.3 Results Discussion

- One concern with configuring the fabric blockchain with 8 MB Blocks is the possibility for large number of commit errors when dealing with heavy loads, which was observed in the set of experiments with 4 MB blocks.
- Figure 29 shows the plot for throughput for adding assets to the blockchain using 8 MB Blocks for various block sizes. Figure 30 represents the plot latency

comparison for various batch sizes. Figure 31 represents the throughput plot for querying the ledger containing 8 MB Blocks using various block sizes.

- This system is able to handle heavy incoming loads successfully in most rounds for adding assets on the blockchain. One striking feature is that the system performs the best when using a message batch size of 500(throughput = 74.1 tps).
- Looking also at the query handling results for 8 MB Blocks, there is a similar trend. High throughput is observed when using higher message batch sizes. We observe high failure rates in final rounds for lower batch sizes, which also leads to drop in throughput. A reason for this is under-utilization of the capacity of the blockchain. 8 MB Blocks can easily support high number of incoming transactions, even those that involve expensive operations. However being restricted by message batch sizes, it leads to transaction piling up which leads to them failing later.
- Contrary to expectations the blockchain when configured with 8 MB Blocks performs better under heavy load conditions.

4.4.4 Testing 16 MB Blocks

4.4.4.1 Adding Assets To The Blockchain Network

For this set of experiments transactions are sent over five rounds with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

Table 19. Results for Adding Assets On The Blockchain Network Using 16 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	24.89	1.25	8.89	23.5	100	0
	round 1	6.76	2.24	4.03	44.8	100	0
	round 2	12.65	1.73	8.48	61	100	0
	round 3	13.62	2.23	10.41	63.7	100	0
	round 4	14.78	2.79	11.41	63.7	100	0
100	round 0	25.18	1.31	8.78	24.2	100	0
	round 1	9.76	2.83	5.4	41.8	100	0
	round 2	12.97	3.15	8.98	56.5	100	0
	round 3	14.36	4.13	10.64	61.2	100	0
	round 4	14.87	3.82	12.07	62.5	100	0
200	round 0	29.82	0.9	8.08	24.5	99.5	0.5
	round 1	10.71	1.77	6.62	45	100	0
	round 2	26.94	6.72	18.58	57.3	100	0
	round 3	29.16	6.51	21.46	62.9	100	0
	round 4	27.23	7.95	21.86	69.1	100	0
250	round 0	29.97	1.26	6.95	24.4	100	0
	round 1	17.62	2.81	9.6	42.8	100	0
	round 2	25.88	5.7	16.97	61.7	100	0
	round 3	29.22	10.46	22.17	62.2	100	0
	round 4	27.35	11.7	22.32	69	100	0
500	round 0	27.82	1.14	6.58	24.5	100	0
	round 1	17.23	2.89	9.45	41.8	100	0
	round 2	26.7	6.28	17.92	54.2	100	0
	round 3	30.01	14.77	21.65	28.5	56.95	43.05
	round 4	28.3	15.3	22.32	59	88.3	11.7

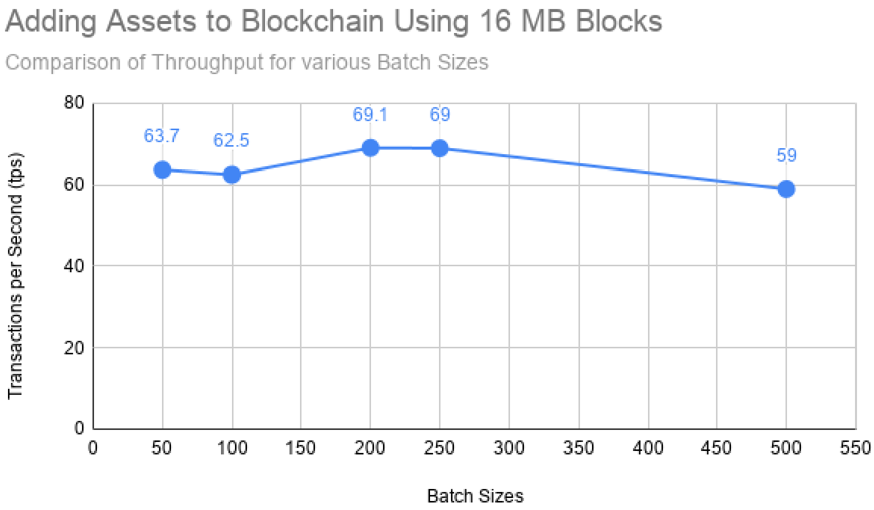

Figure 32. Throughput plot for various message batch sizes tested for 16 MB Blocks

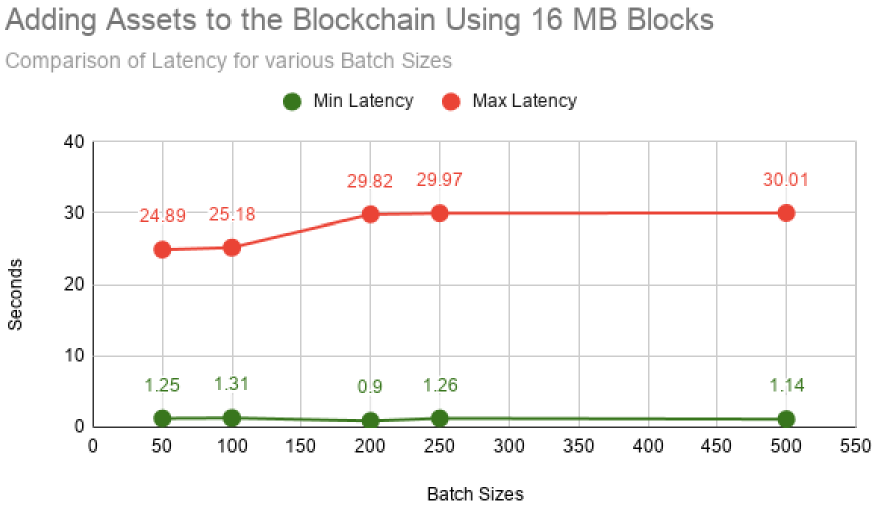

Figure 33. Latency Comparison for various message batch sizes tested for 16 MB Blocks

4.4.4.2 Querying The Ledger

For this set of experiments transactions are sent over five rounds with varying send rates which can be described as follows:

- **Round 0**: 25 tps
- **Round 1**: 50 tps
- **Round 2**: 100 tps
- **Round 3**: 200 tps
- **Round 4**: 500 tps

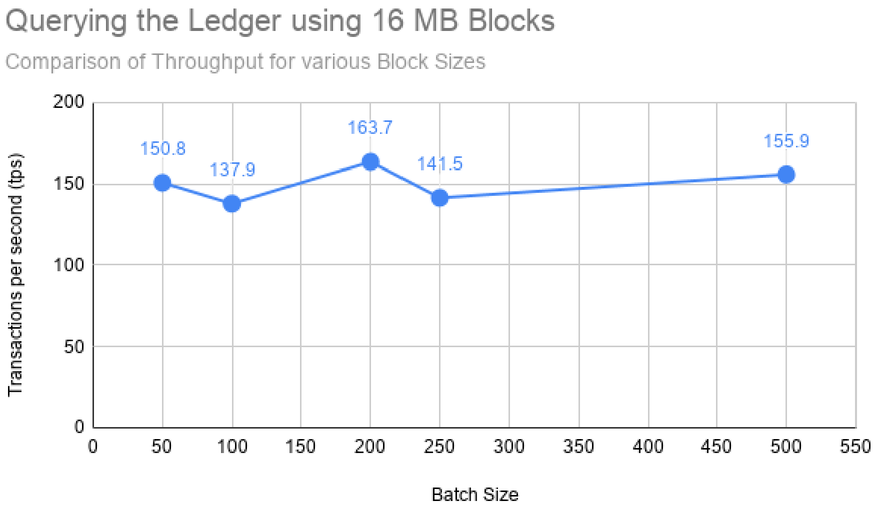

Figure 34. Throughput plot for various message batch sizes tested for 16 MB Blocks

Table 20. Results For Querying The Ledger Containing 16 MB Blocks

Batch Size	**Round No.**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput**	**Success %**	**Failed %**
50	round 0	2.34	0.16	1.23	24.4	100	0
	round 1	2.51	0.31	1.39	49.5	100	0
	round 2	3.17	0.59	1.74	93.7	100	0
	round 3	5.82	1.89	3.58	150.8	100	0
	round 4	9.66	5.47	8.24	129.2	77.55	22.45
100	round 0	16.92	0.18	2.82	24.5	100	0
	round 1	2.26	0.24	1.27	49.8	100	0
	round 2	1.44	0.26	0.88	98.9	100	0
	round 3	7.91	0.7	4.32	133.4	91.95	8.05
	round 4	9.61	2.95	7.79	137.9	82.3	17.7
200	round 0	17.4	0.17	2.93	24.4	100	0
	round 1	2.54	0.29	1.38	47.6	100	0
	round 2	2.66	0.42	1.6	98	100	0
	round 3	5.71	1.17	3.47	163.7	100	0
	round 4	9.8	6.71	8.47	148.3	90.35	9.65
250	round 0	2.39	0.17	1.25	24.5	100	0
	round 1	2.48	0.31	1.37	47.5	100	0
	round 2	3	0.62	1.71	94.5	100	0
	round 3	5.86	1.15	3.05	141.5	90.45	9.55
	round 4	9.36	5.96	7.87	117.8	68	32
500	round 0	2.36	0.16	1.22	24.4	100	0
	round 1	2.52	0.3	1.39	47.5	100	0
	round 2	2.99	0.59	1.68	95.7	100	0
	round 3	5.69	1.52	3.38	155.9	100	0
	round 4	8.56	8.56	7.24	106.4	52.85	47.15

4.4.4.3 Results Discussion

- The concern expressed while experimenting with 8 MB blocks becomes real while testing 16 MB blocks. Failures occur due to large block sizes and commit errors pertaining to replication of blocks on peers of the blockchain.
- This discussion is supported by Figure 32, Figure 33 and Figure 38.

- For adding assets into, the blockchain, the maximum throughput achieved is 69.1 transaction per second. However at higher message batch sizes and while dealing with high incoming load conditions, high failure rates are observed due to commit errors.
- The maximum latency observed is consistently high for most batch sizes.
- A large block size also affects the throughput for query handling adversely. The maximum throughput achieved is 163.7 tps. Although this could be considered satisfactory, one worrying factor is high failure rate under high incoming transaction load conditions. Unlike the blockchain with 8 MB blocks, using 16 MB blocks hampers the operating efficiency of the blockchain for both adding and querying operations.

4.5 Comparison of Models

Comparing *small block sizes* to *large block sizes*, we observe that *large block sizes* perform better while handling heavy workloads. Considering both kinds of workloads together, 8 MB blocks with a message batch size of 500, achieves the best throughput overall.

This blockchain configuration acts as common ground for comparing the *Real Time Energy Transactions* model to the *Energy Futures* model. Using the experimental setup described in Section 4.1 we can compare both systems on basis of settlement time for offers submitted.

This experiment runs for 100 rounds where each round involves:

1. The nodes from the IEEE 34-node bus submitting energy usage, generation and storage data to the blockchain network using hourly data from National Renewable Energy Library(NREL) data set for Arizona.
2. Based on the available energy in the user account, offers are submitted to the blockchain network (bidding time).
3. After submitting offers from all members the offer matching smart contract is called by any one peer part of the commonchannel.
4. The response time for matching offers and updating the accounts for users is recorded for both the models.

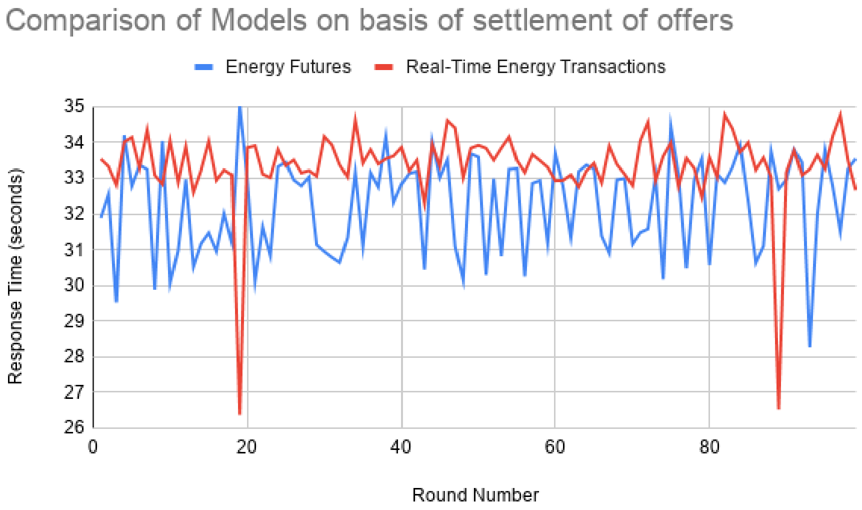

Figure 35. Comparison plot for both models

Figure 35 shows the comparison of smart contract response time over 100 rounds for *Real Time Energy Transactions* vs *Energy Futures*. The *Energy Futures* model performs better in 78 rounds.

One reason for better performance is the amount of offer rejections the transaction flow has to deal with. In the *Energy futures* model, if a matching offer is not found, the offer is not rejected right away, it is retained for the next round of offer matching. However in the *Real Time Energy Transactions* model, the offer status is changed to rejected On average the *Real Time Energy Transactions* makes more updates on the commonchannel due to the characteristic of it's transaction flow. increases the updates that have to be made to the commonchannel.

The major trade-off that accompanies the transaction flow of the *Energy Futures* model is the transaction load at the start of each round. Due to retention of offers, in some rounds, this model has larger amount of offers to examine and match than the *Real Time Energy Transactions* model.

Chapter 5

CONCLUSION

5.1 Performance Summary

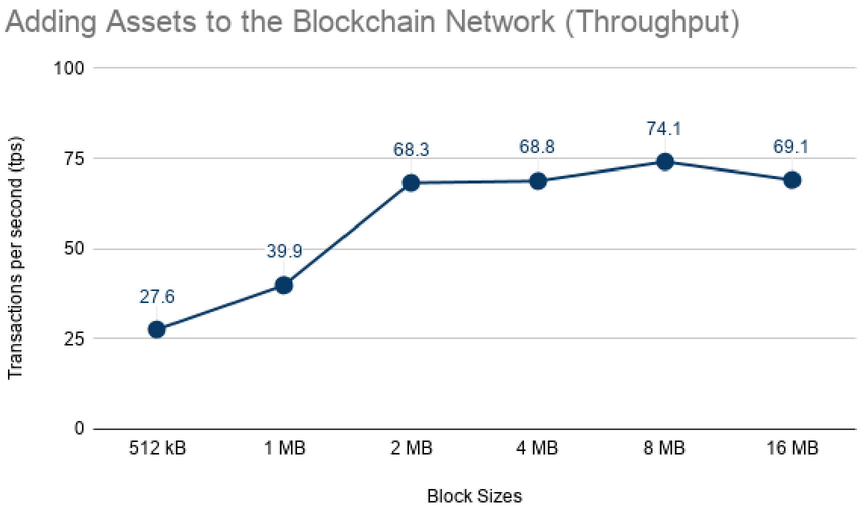

Figure 36. Overall throughput comparison for all block sizes for adding assets to the blockchain

Summarizing all performance tests for adding assets onto the blockchain, we observe that throughput keeps increasing as we increase block size. Peak throughput is achieved when using 8 MB blocks are used with a message batch size of 500. We infer that a large block size with sufficient allowed batch size, is able to perform better processing a load of expensive operations.

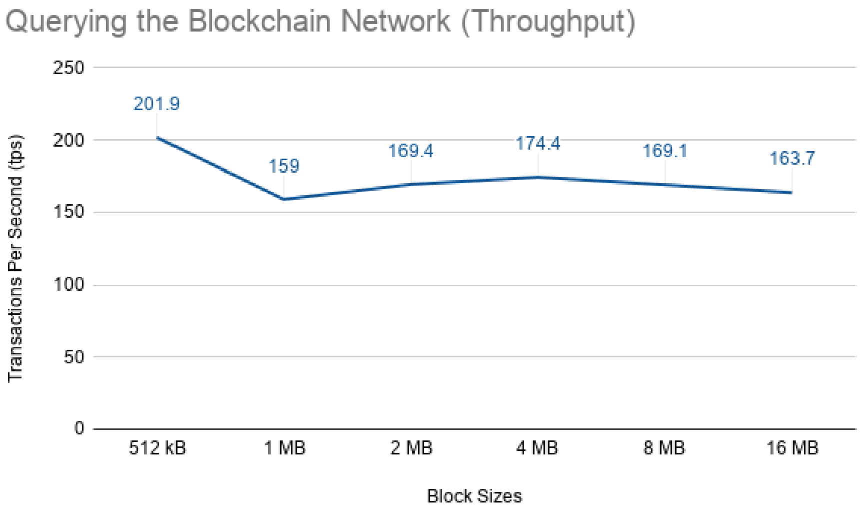

Figure 37. Overall throughput comparison for all block sizes for querying the blockchain

For query based test workloads, although 512 kB blocks exhibit the highest throughput, the high failure rate for adding assets makes it not suitable for production grade deployments. Peak throughput is achieved while using 4 MB blocks with a message batch size of 200.

Comparing *small block sizes* to *large block sizes*, we observe that *large block sizes* perform better while handling heavy workloads. Considering both kinds of workloads together, 8 MB blocks with a message batch size of 500, achieves the best throughput overall. This also opens up a discussion that this blockchain system performs better than most public blockchains in terms of throughput while using significantly larger block sizes. The bitcoin blockchain operates at an average throughput of 3.8 transactions per second using 1 MB Blocks. The Ethereum Blockchain operates at a throughput of 20 tps on average while using block sizes smaller than 40 kB.

5.2 *Real Time Energy Transactions* vs *Energy Futures*

From the discussion in section 4.5 we can infer that a distributed implementation of a variable price energy trading model(*Energy Futures*) can outperform a fixed price energy trading model(*Real Time Energy Transactions*). The *Energy Futures* model can be operated in parallel with the traditional energy billing system. Additional charges/discounts can be applied to the energy bills for users based on data recorded in user accounts. For a particular user, there is full transparency of data on u2cchannel, u2bchannel and u2schannel.

A parallel energy marketplace with bilateral contracts at user level can be operated using the *Energy Futures* model by viewing energy as a digital asset and utilizing offers and accounts to trade the digital asset. To achieve high throughput as well as low failure rate in such a system, the HLF blockchain can be configured to use 8 MB blocks with a message batch size of 500.

5.3 Future Work

One limitation of using a permissioned blockchain is that the digital asset cannot be shared outside its ecosystem. The asset loses meaning outside its ecosystem. One way to utilize the asset outside its ecosystem is to store the proof of ownership of that asset on a public blockchain.

An *OP_RETURN* transaction allows data to be embedded into the payload of the transaction. This data is encoded when a raw transaction is formed. Using the transaction id received after confirmation, the encoded payload can be decoded. This would establish proof of ownership of that asset for any outside party without the hassle of being a member of the permissioned blockchain

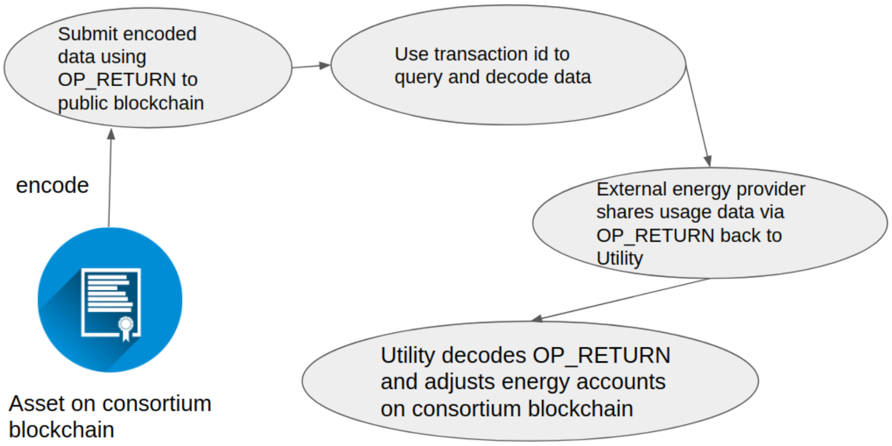

Figure 38. Sharing an Asset Outside Its Ecosystem

Although this process is accompanied with trade-offs governing transaction confirmation, since it is delayed in public blockchains. The data to be shared needs to be encrypted so that only select actors can decrypt it, since any member of the public blockchain has the ability to view the data present.

REFERENCES

[1] *2010 assessment of demand response and advanced metering staff report. Federal Energy Regulatory Commission, Washington, DC, USA, 2010.* https://www.ferc.gov/legal/staff-reports/2010-dr-report.pdf. 2010.

[2] Merlinda Andoni et al. "Blockchain technology in the energy sector: A systematic review of challenges and opportunities". In: *Renewable and Sustainable Energy Reviews* 100 (2019), pp. 143–174.

[3] Merlinda Andoni et al. "Blockchain technology in the energy sector: A systematic review of challenges and opportunities". In: *Renewable and Sustainable Energy Reviews* 100 (2019), pp. 143–174.

[4] Elli Androulaki et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains". In: *Proceedings of the Thirteenth EuroSys Conference*. ACM. 2018, p. 30.

[5] Ethan Buchman. "Tendermint: Byzantine fault tolerance in the age of blockchains". PhD thesis. 2016.

[6] LR Clarke. "New electricity trading arrangements in England and Wales". In: *IEEE/PES Transmission and Distribution Conference and Exhibition*. Vol. 2. IEEE. 2002, pp. 1470–1472.

[7] TGA Council. "Gridwise transactive energy framework version 1.0". In: *The GridWise Architecture Council, Tech. Rep.* (2015).

[8] Omar Dib et al. "Consortium blockchains: Overview, applications and challenges". In: *International Journal On Advances in Telecommunications* 11.1&2 (2018).

[9] Cherrelle Eid et al. "Managing electric flexibility from Distributed Energy Resources: A review of incentives for market design". In: *Renewable and Sustainable Energy Reviews* 64 (2016), pp. 237–247.

[10] Sameh El Khatib and Francisco D Galiana. "Negotiating bilateral contracts in electricity markets". In: *IEEE Transactions on Power Systems* 22.2 (2007), pp. 553–562.

[11] *Energy Saving Trust, Buying green electricity.* http://www.energysavingtrust.org.uk/home-energy-efficiency/switching-utilities/buying-green-electricity. 2017.

[12] Adam Hahn et al. “Smart contract-based campus demonstration of decentralized transactive energy auctions”. In: *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE. 2017, pp. 1–5.

[13] *Hyperledger Fabric Architecture.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/arch-deep-dive.html.

[14] *Hyperledger Fabric Channels.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/channels.html.

[15] *Hyperledger Fabric Endorsement Policies.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/endorsement-policies.html.

[16] *Hyperledger Fabric Identity.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/identity/identity.html.

[17] *Hyperledger Fabric Identity.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger.html.

[18] *Hyperledger Fabric Membership.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/membership/membership.html.

[19] *Hyperledger Fabric Peers.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html.

[20] *Hyperledger Fabric Smart Contracts.* https://hyperledger-fabric.readthedocs.io/en/release-1.4/smartcontract.html.

[21] Marija Ilic, Francisco Galiana, and Lester Fink. *Power systems restructuring: engineering and economics.* Springer Science & Business Media, 2013.

[22] *Indigo Advisory Group. Blockchain in energy and utilities use cases, vendor activity,market analysis.* https://www.indigoadvisorygroup.com/blockchain https://www.indigoadvisorygroup.com/blockchain. 2008.

[23] Ivana Kockar. “Combined pool/bilateral operation in electricity markets.” In: (2004).

[24] Ivana Kockar and Francisco D Galiana. “Combined pool/bilateral dispatch. II. Curtailment of firm and nonfirm contracts”. In: *IEEE Transactions on Power Systems* 17.4 (2002), pp. 1184–1190.

[25] Power Ledger. “Power Ledger white paper”. In: *Power Ledger Pty Ltd, Australia, White paper* (2017).

[26] Xing Luo et al. “Overview of current development in electrical energy storage technologies and the application potential in power system operation”. In: *Applied energy* 137 (2015), pp. 511–536.

[27] Esther Mengelkamp et al. “Designing microgrid energy markets: A case study: The Brooklyn Microgrid”. In: *Applied Energy* 210 (2018), pp. 870–880.

[28] Michael Mylrea. “Smart energy-internet-of-things opportunities require smart treatment of legal, privacy and cybersecurity challenges”. In: *The Journal of World Energy Law & Business* 10.2 (2017), pp. 147–158.

[29] Michael Mylrea and Sri Nikhil Gupta Gourisetti. “Blockchain: A path to grid modernization and cyber resiliency”. In: (2017), pp. 1–5.

[30] Cuong Nguyen. “Transactive Energy: An Overview”. In: (2017).

[31] *Pearce F. The great green electricity con.* https://www.theguardian.com/environment/2008/oct/23/energy. 2017.

[32] Manisa Pipattanasomporn, Murat Kuzlu, and Saifur Rahman. “A Blockchain-based Platform for Exchange of Solar Energy: Laboratory-scale Implementation”. In: *2018 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE)*. IEEE. 2018, pp. 1–9.

[33] KP Schneider et al. “Analytic considerations and design basis for the IEEE distribution test feeders”. In: *IEEE Transactions on Power Systems* 33.3 (2017), pp. 3181–3188.

[34] Karim Sultan, Umar Ruhi, and Rubina Lakhani. “Conceptualizing Blockchains: Characteristics & Applications”. In: *arXiv preprint arXiv:1806.03693* (2018).

[35] Ross Baldick - University of Texas at Austin. *Single Clearing Price in Electricity Markets.* http://www.cramton.umd.edu/papers2005-2009/baldick-single-price-auction.pdf. 2009.