Blockchain-based Micropayment System for Secured Cashless Small Payments

by

Morshed Siam
17101253
Shovon Mandal
18101142
Mahmuda Akter Keya
18101414
Swarojani Dhar
18101145
H.M. Irfan Tahsin
18101515

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
September 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Morshed Siam.

Morshed Siam
17101253

Shovon Mandal

Shovon Mandal
18101142

Mahmuda Akter Keya

Mahmuda Akter Keya
18101414

Swarojani Dhar

Swarojani Dhar
18101145

H.M Irfan Tahsin

H.M. Irfan Tahsin
18101515

Approval

The thesis/project titled "Blockchain-based Micropayment System for Secured Cashless Small Payments" submitted by

1. Morshed Siam(17101253)
2. Shovon Mandal(18101142)
3. Mahmuda Akter Keya (18101414)
4. Swarojani Dhar (18101145)
5. H.M. Irfan Tahsin (18101515)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 26 September, 2021.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The Internet of Things (IoT) is driving a tremendous digitization tsunami right now. IoT devices create huge amounts of data in the internet of everything era. Many *IoT* data services use distributed ledger technology like blockchains or *IOTA*. The *IOTA Foundation* has completely has entirely redesign distributed ledger technology to enable for secure cash and data exchange. It was intended to enable fee-free micro-transactions in the growing IoT device network. It provides a scalable network growth approach as well as transaction confirmations to enable smart device micro-transactions. The quicker the network gets using the IOTA Tangle, the more transactions are verified. IOTA currently has a maximum transaction rate of roughly 7 transactions per second (TPS). The community network reached a maximum of 600 confirmed Transactions Per Second in May 2020. (CTPS). We propose a methodology for integrating Tangle into IoT blockchains in this article, with Tangle serving as the backbone for all IoT devices. We will use two distinct types of interfaces: web and NFC, and we will provide a solution to the message overhead problem caused by message flooding. Also, we would want to present a cost-cutting strategy that drastically cuts transaction time and storage for modest payments. The results clearly reflect the efficacy and efficiency of our framework.

Keywords: IOTA; Blockchain; IoT; NFC- Near Field Communication; Q-learning; IOTA 2.0 DevNET

Acknowledgement

Firstly, With the gifts of Almighty, we completed this report on-”Blockchain-based Micropayment System for Secured Cashless Small Payments”. This was an extraordinary learning experience from the earliest starting point.

Secondly, We want to express our heartiest thanks to our Supervisor Dr. Md. Golam Rabiul Alam for his appreciation. We are really honored that we have the chance to work with him. All through this research we have learned and experienced numerous significant things which will help us in the future.

Finally, we would like to specific our appreciation towards our guardians for their proceeded supplications, support, and bolster without which it would not have been conceivable for us to reach this point.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ϵ	Epsilon
υ	Upsilon
$A3C$	Asynchronous Advantage Actor Critic
$CTPS$	Confirmed Transactions Per Second
DLT	Distributed Ledger Technology
IoT	Internet of Things
NFC	Near Field Communication
TPS	Transactions Per Second

Chapter 1

Introduction

1.1 Background

Prepaid cards or mobile phones are the payment methods of choice in the majority of micropayment systems now in operation. However, if IoT devices can make automated payments, customers may save a lot of the hassle that comes with sophisticated transaction procedures. Furthermore, mobile payment systems frequently rely on third-party financial institutions. However, such dependency might lead to security issues. Personal property, for example, might be jeopardized if these institutions' systems are breached. As a result, we present iota tangle as a blockchain-based micropayment system. IOTA is a free and open data and value transmission protocol (and network)[8]. The IOTA Tangle is a cutting-edge distributed ledger technology (DLT) that was created exclusively for the Internet of Things (IoT). Data and value can be transmitted individually using the IOTA protocol. This is one of the most significant distinctions from other Blockchains. One of its main applications is as a data and value exchange operating system in the upcoming internet. The protocol stands apart from other distributed ledgers because it is built on a directed acyclic graph. The usage of IOTA is anticipated to make transactions and operations involving items with sensors easier. Furthermore, the time it takes to confirm a value transaction is between 10 and 12 seconds. Consider giving money to someone and having it arrive 10 seconds later. You also won't have to pay any fees.

In this work, we create a Block-chain based micropayment mechanism to incorporate Tangle with IoT blockchains[6]. As a control station, we'd want to deploy a consortium blockchain, with Tangle serving as the backbone of the networked IoT devices. The combination of blockchain and iota tangle can enhance the Internet of Things by offering a trustworthy sharing service with traceable data. Data sources may be recognized at any moment, and data is immutable throughout time, enhancing security. This integration would be a game changer in scenarios when IoT data has to be securely shared across a large number of people. However, there are a number of research problems and open concerns that must be addressed before these two technologies can be used together effortlessly, and this study area is still in its early stages. Improvements that this integration may provide, for example, include (but are not limited to):

1.1.1 Decentralization and Scalability

It will also assist to avoid situations where a few powerful firms have complete control over the processing and storage of a large number of people's data. Other advantages of decentralizing the design include increased fault tolerance and system scalability, according to reference. It would help to break down IoT silos while also helping to improve IoT scalability.

1.1.2 Identity

Using a shared blockchain system, participants may identify each device. The data that is supplied and fed into the system is immutable, and it identifies the data that was provided by a device in a unique way. Furthermore, blockchain can provide reliable distributed authentication and authorization of devices for IoT applications**Reyna**. This would be a significant advancement for the Internet of Things and its participants.

1.1.3 Reliability

In blockchain, IoT data may be immutable and dispersed across time [3]. Participants in the system may check the data's validity and ensure that they haven't been tampered with. Furthermore, the technology allows for the traceability and accountability of sensor data. The crucial feature of the blockchain to bring in the IoT is reliability.

1.1.4 Security

Information and communications can be safeguarded by storing them as blockchain transactions [3]. Device message exchanges can be treated as transactions on the blockchain, with smart contracts validating them, encrypting interactions between devices. With the usage of blockchain, current secure standard-protocols used in the IoT may be improved. [3].

1.2 Problem Statement

1.2.1 Message Overhead

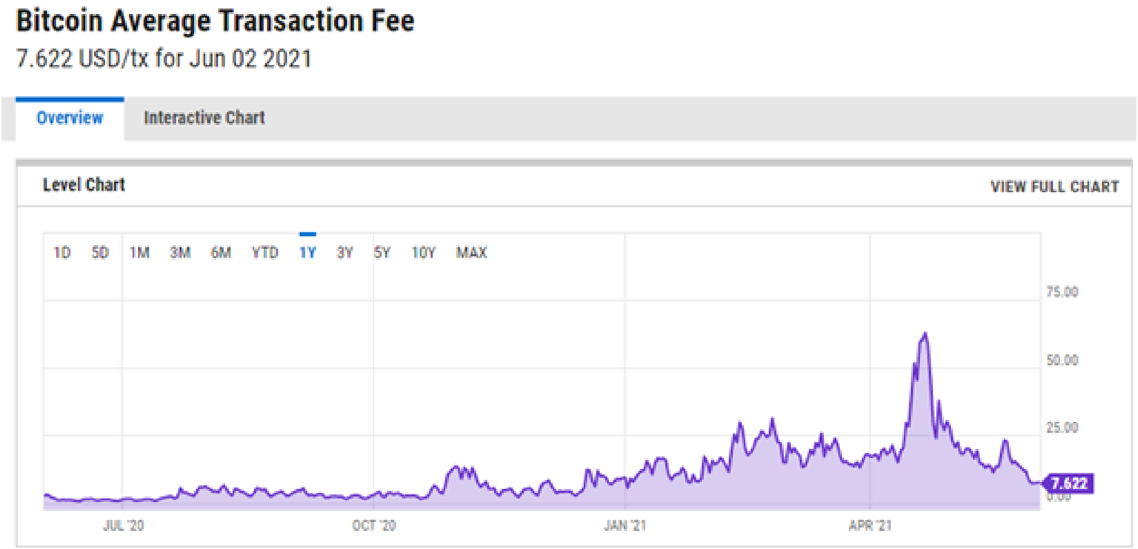
Message in IOTA is now referred to as not just for the transaction because according to the protocol of IOTA, it is not just a value transfer application but a platform for securely storing and transmitting data. When a new message wants to connect with the tangle in order to make a transaction. The IOTA protocol allows for a host of applications to run on the message tangle. Anybody can design an application, and users can decide which applications to run on their nodes. These applications will all use the communication layer to broadcast and store data. Steps for creating a message:

1. First, it needs to go through the Congestion Control Mechanism which is the tips selection algorithm. To do that, the new message needs to choose 2 to 8 existing recent unreferenced messages(tips). These unreferenced messages are already issued by some nodes of IOTA. Congestion Control works as a filter and it provides some sort of white flag for the new message that will be forwarded to the neighbors to gossip.
2. After passing the congestion control filter, it has to be verified whether they are correct or not through checking the valid signature, correct UTXO balances, etc.
3. Sign the message.
4. Perform the Adaptive PoW, which prevents DoS attacks.
5. Gossip to the neighbors. But this gossip protocol is regulated by the access control system.
6. Here, throughput is regulated by a sophisticated access control system that ensures fairness.

Now the problem is, in the current network, the gossip protocol that is used to send messages to the neighbors is actually a flooding problem[10]. We are sending a message to all the neighbors which is not optimized at all. We are basically broadcasting a message to all of our neighbors. But moving away from flooding is not straightforward because of the consistency criteria. The target is to maintain lower delays and a good delivery rate. But to maintain such consistency, it causes message overhead due to message flooding.

1.2.2 Payment

Payment systems are the means through which payments are exchanged between financial institutions, corporations, and people, and they are a vital component of a country's financial system's correct operation. There are a variety of payment systems available in Bangladesh, including Bkash, Rocket, Nagad, and others. For cash outs or transections, all of these payment options entail a slew of fees. For example, the Bkash App Cash Out Fee is 1.75 percent. In other words, if you cash out using the Bkash app, it will cost you 17.50 Taka every thousand Taka. Furthermore, Nagad charges 5 tk for each type of transfer money transaction and adds 15% VAT to cash out transactions[13]. Rocket also takes a 1.8 percent cut of the money you withdraw. As a result, practically every payment method requires consumers to pay an additional charge, which accounts for a significant portion of the money transferred. A system manager, on the other hand, is in charge of the system. As a result, the system is not sufficiently protected, and it may be blocked or hacked at any time. Other alternatives exist, such as blockchain-based micropayments, such as IOTA in our situation. Bitcoin, Ethereum, and other cryptocurrencies are examples. Average Bitcoin transaction costs can skyrocket during moments of network congestion, as they did during the 2017 Crypto-boom, when they hit around 60 dollar. The average transaction fee for Bitcoin has risen to 7.365 dollar one year ago.

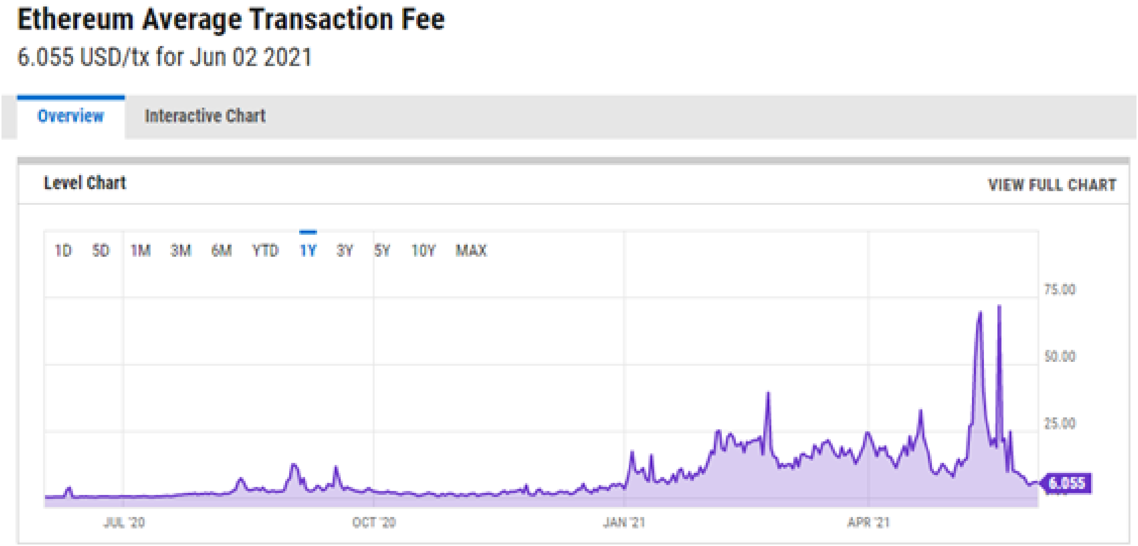


Bitcoin Average Transaction Fee

7.622 USD/tx for Jun 02 2021

Figure 1.1: Bitcoin Average Transaction Fee.

Supply and demand are the primary causes of high bitcoin mining fees. Because each bitcoin block is 1MB in size, miners can only confirm 1MB worth of transactions per block (one every ten minutes). As a result, miner fees have risen dramatically. The fee is paid to the miner who creates the block containing your transaction. The charge is determined by the transaction's size (in bytes) and the age of its inputs [24]. The average transaction cost for Ethereum is currently 5.958, up from 5.75 yesterday and 0.4518 a year ago. This is a 3.63% increase from yesterday and a 1.22% increase from a year ago.



Ethereum Average Transaction Fee

6.055 USD/tx for Jun 02 2021

Figure 1.2: Ethereum Average Transaction Fee.

1.3 Proposed Solution

1.3.1 Payment:

IOTA is a forward-thinking cryptocurrency that reacts to the Internet of Things' expanding technical demands. Rather than depending on a blockchain to conduct these transactions, the IOTA tangle employs an infinite-scale graph with no miners and no blocks[5]. As a result, there are no transaction fees in iota tangle. Furthermore, it is safe since the system is controlled by an algorithm. It stores transactions on its ledger using a directed acyclic graph, which has the potential to be more scalable than blockchain-based distributed ledgers.

1.3.2 Message Overhead:

Due to the use of gossip protocol, a new message is broadcasted to all of its neighbors which causes message overhead due to message flooding[9]. It is not optimized at all. To solve this problem, we are introducing High Mana Based q-learning algorithm to solve this problem. Normally, for a new message, mana is chosen with either criteria 1 or criteria 2. Criteria 1 or Mana 1 is based on the amount of transaction. And Mana 2 is based on request. Basically, we have to request for mana to other nodes in the tangle. So, when a new transaction occurs and if this transaction is referred by some node, then a new transaction gained some mana. But to peer, the new message tends to peer nodes with the same Consensus Mana(cMana). So, our solution is, not to choose peers with the same cMana. Instead of the same cMana, we are using nodes with high consensus mana. High cMana is our reward function. So, we are training the network to choose nodes with high cMana through q-learning algorithm.

1.4 Research Contribution

This research aims to develop a blockchain-based micropayment system for secured cashless small payments. Usually, scalability strongly hampers the micropayments of economic systems based on the blockchain technology[7]. Using bidirectional micropayment or commitment transactions are suitable solutions for secured cashless small payments on economic systems based on blockchain. The contributions of this research are:

1. We created a website for IOTA transactions, balance checks, and token additions.

2. Using Near-Field Communication, we developed a secure cashless payment system (NFC)

3. To gain access to the tangle, we used IOTA's Firefly wallet.

4. IOTA Faucet

5. Client library for IOTA
6. Used Q-learning algorithm to solve message overhead problem of IOTA

Chapter 2

Literature Review

2.1 Background study

Blockchain is a distributed database that a group of individuals controls, store and share information. It is a P2P (Peer to Peer), decentralized and distributed ledger. It involves a third-party intermediary, and they are the miners. According to [1] there are 4 types of blockchains,
i) Public Blockchains
ii) Private Blockchains
iii) Consortium Blockchains
iv) Hybrid Blockchains.
Now considering the concept of micropayment with respect to blockchain, Micropayment is electronic transfer of very small amount of money. The amount which is considered as micropayment varies. Like some organization consider micropayment as less than a dollar, some might say less than or equal to 20 US dollars so on and so forth. According to [5], web micropayment first gained popularity in the 1990s. But most of this fintech companies were far ahead of their time and loses the market interest. Examples could be, Digicash which was founded in 1989, BitPass in 2002. But in 2010, a new generation of micropayment with decentralized Internet associated with it hit the market. Though this new generation is far better than the previous one but still got some problems.

2.1.1 Scalability

One of the most crucial problem of blockchain is "Scalability" which is affecting micropayments also. Blockchain commits to replace usual banking system, credit cards, debit cards and all. Considering VISA as an example, it can handle 1000s of transactions per second. PayPal manages 193 transactions per second, Ethereum does only 20 while bitcoin manages only 7 transactions per second [2]. Bitcoin is currently being traded as the volume of 3 hundred thousand per day. So, the network is currently being operating at its maximum capacity. Same goes for Ethereum [3]. Two popular cryptocurrencies like bitcoin and Ethereum, they are becoming the mainstream day by day as the number of transactions is increasing in an exponential manner. Here is the graph of Ethereum transaction oner last 10 years:

2.1.2 Miner

If we talk about Bitcoin, the basic principle says that, mining the nodes must verify each and every transaction in the network. It is the miners who become the bottleneck within the transaction process. When Bitcoin was started in 2008, verification was achievable because at that time it was not that much popular and network size was not that much bigger. But with the growing popularity, the number of transactions has increased in a drastic manner which causes a maintenance issue [5]. Then comes speed. If we consider public blockchain, it is slow. Both bitcoin and Ethereum uses PoW (Proof of Work). And miners are solving the puzzles which is how proof of work works. But now it is not just a matter of few nodes. Now it is a matter of vast number of nodes. And it requires a huge energy to solve the puzzles. Which causes high energy consumption as miners need to have a proper electricity supply. And this costs a lot. Every time the ledger gets updated with a new transaction even with a small amount; the miners need to solve the problem which means spending a lot of energy. [6]

2.1.3 Slow

Blockchain gets slow when there are too many users in the network. Blockchain network relies on the nodes to function properly. So, quality of nodes is a matter. It determines the quality of the blockchain. And quality also depends on the number of users at a time in the system. Too many users cause a slow system.[6]

2.1.4 High Cost

Another problem of blockchain is “High Cost”. Here cost is associated with managing developers, team, miners, licensing cost and maintenance cost. Considering bitcoin payment, it usually refers to the amount bitcoin owners pay to bitcoin miners whenever they send funds to another bitcoin address. In bitcoin, a block or node can only hold a finite number of transactions. So, when the network is crowded or busy then there are several transactions waiting to be confirmed. In this situation miners will prioritize those transactions with higher fee attached with it. And thus, small payments get less priority. If anyone want to get faster confirmation, then that person must attach larger fee. And smaller amount for those who are not in that much rush which causes a slow transaction process for micropayments. And the fees are always paid by the senders. [7]

2.1.5 IOTA Coordinatorr

The problem of IOTA was, IOTA was secure only based on a temporary component called coordinator. It was considered as a single point of failure. Now the problem is solved with the concept of coordicide. It allows the IOTA to be fully decentralized. This is revolutionary voting module where consensus is reached through proactive communication. In IOTA we call it shimmer. In the blueprint, we unveil all the modules and the freedom they bring for building and integrating the network. Because at the end of the day iota is about freedom. Freedom to transact data and value without fees, freedom for billions of IOT devices to access the network, freedom to infinitely innovate. So that is how the problem of coordinator is also solved. In our research paper for Blockchain based micropayment for secure cashless small payments we are going to use IOTA Tangle as the cryptocurrency architecture rather than Blockchain. Blockchain is the backbone of a new type of internet as it allows digital information to be distributed but not copied [13]. It is originated with the creation of bitcoin in 2008 by Satoshi Nakamoto. But its practical implementation started in 2009. IoT is an umbrella term that covers technologies, design principles, and systems associated with the ever-growing phenomenon of Internet-connected devices – "Things". According to [6], IoT as a phrase is not new. It appeared for the first time in 1999 at Massachusetts Institute of Technology (MIT) Auto-ID Centre and was used to refer to building an Internet based network that cover all things in the world to realize automatic identification of things through information sharing.

2.1.6 IOTA network architecture

We integrate Tangle into IoT blockchains and establish a cross-chain interactive decentralized IOTA data access paradigm as part of our proposed IOTA network architecture based on blockchain. Tangle uses a Directed Acyclic Graph (DAG) instead of a continuous chain architecture and adds blocks on a regular basis. Tangle has a greater transaction throughput (because to parallel verification) and no transaction costs thanks to DAG. As Tangle grows, more people will initiate transactions, the system as a whole will become more safe and quick, confirmation times will be cut, and transactions will be completed faster and faster.

2.1.7 Gossiping protocols

A gossip protocol is a peer-to-peer computer type of technology or procedure based on the spread of infections. Because gossip spreads information in a biological community in a similar fashion to how a virus does, the epidemic protocol and gossip protocol are commonly interchanged. Initially, gossip protocols were used to keep databases that were duplicated over hundreds of sites consistent. It was soon found that gossiping might be used to solve other issues, such as calculating averages across a network of nodes or establishing a network overlay. Maintaining node membership is another issue that has been addressed.

These protocols usually work like this

The basic notion of a gossip protocol is simple. Each node has a group of nodes with which it communicates. Data flows through the system node by node, much like a virus. Every node in the system eventually receives data. Each node processes the data it receives, and the nodes in the network repeat these operations on a regular basis to disperse information. The gossip protocol, in general, is a computer-to-computer interaction method or process based on how social networks disseminate information or diseases spread.

Use of the gossip protocol: The gossip protocol is a participant strategy that provides that information is sent to all network users in current distributed systems. The gossip protocol is called Epidemic Protocol because it disseminates or distributes data in the same way that an epidemic spread a virus in a biological ecosystem. Without incurring undue strain, gossip can be exchanged as frequently as once every tenth of a second. This type of network search could take three seconds to search a huge data center.

Use of gossip protocol

This peer-to-peer gossip protocol is used in modern distributed systems to ensure that information is delivered to all network users. Because it disseminates or transmits data in the same way that an epidemic spread a virus in a biological ecosystem, the gossip protocol is dubbed Epidemic Protocol. To ensure that data is distributed to all members of a group, several distributed systems use peer-to-peer gossip. Example: In a network of 25,000 machines, for example, we can identify the best match after around 30 rounds of gossip: 15 rounds to distribute the search string and 15 more to locate the best match. Due to a gossip exchange can happen as frequently as once per tenth of a second without causing undue burden, this type of network search might explore a large data center in around three seconds. In this scenario, searches may age out of the network after a period of time, such as 10 seconds. By that time, the initiator has figured out the answer, thus there's no point in continuing to talk about the search.

Three basic approaches

The three primary techniques to implementing a gossip protocol are as follows. The method a message is spread to neighbors, as well as who initiates the gossip exchange: the recipient or the sender, distinguish these diverse strategies. • Eager push approach: As soon as nodes receive a message for the first time, they send the entire payload to randomly selected peers. The sender is the one who initiates this approach.

- Pull approach: Nodes poll random peers for information about recently received or available messages on a regular basis. When they become aware of a message they haven't yet received, they ask their neighbor for the payload of that message. This technique is best used in conjunction with a best-effort broadcast mechanism.

- Lazy push approach: Whenever a node first receives a message, it simply gossips the message identifier (for example, a hash of the message), not the entire payload. When peers receive an identifier for a message they haven't yet received, they ask the sender for the payload.

Drawbacks of Gossiping Protocol

One downside of gossip-based broadcast protocols is that, in order to ensure high dependability, they require an excessive amount of message overhead. Structured broadcast protocols, such as those based on tree-construction, do not have this problem; nonetheless, structured protocols are particularly brittle in the face of failures, missing the natural resilience of epidemic protocols. The development of novel broadcast primitives that blend gossip-based and tree-based techniques is a viable approach; this way, one may benefit from the scalability and durability of pure gossip-based solutions while approaching the efficiency of tree-based solutions. Even though Gossip protocols provide system resilience by allowing nodes to continue running without interruption in the event of a breakdown, the information or distributed message may be impacted. If a node turns malevolent, for example, it can gradually modify information so that the message stays legible while also containing fraudulent or incorrect information. And, to add to the congestion, the other nodes, which will continue to function normally, will disseminate this information. According to their theory, the estimation average value in a local node can be viewed as a global average value after $O(logn + log(1/) + log(1/))$ rounds. However, how can we know how many rounds an estimation average value is close enough to the real average value if we don't know the network size.

Gossip Protocol in IOTA

Gossip is most commonly encountered in IOTA's Background data broadcasting protocols, in which any new transaction is broadcast to all neighbors at the same time. The gossip protocol, as we all know, is a point-to-point communication protocol, but instead of contacting each network participant individually, nodes have a restricted number of neighbors with whom they can exchange messages. Any received message is forwarded to the neighbors. When the number of neighbors surpasses one, the messages spread exponentially across the whole network.

The iota protocol is a data storage and delivery infrastructure that is secure. The IOTA protocol allows a wide range of applications to run on the message tangle. Users may pick which applications to run on their nodes, and anyone can design an application. The communication layer will be used by all of these applications to broadcast and store data.

2.1.8 Congestion control mechanism

Congestion control is a technique for monitoring and controlling the total quantity of data entering a network in order to keep traffic levels at a manageable level. This is done in order to prevent the telecommunication network from collapsing due to congestion. As a result, the upstream hub or hubs may get congested, rejecting information from their upstream node or nodes.

Congestion in an organization can occur if the network's load (the number of bundles delivered out of the organization) exceeds the network's capacity.

- Congestion will occur when such a high number of parcels are sucked into the framework, resulting in degraded execution.
- Backups and congestion will generally take care of themselves.

- Congestion indicates a lack of coordination across various systems administration equipment.

categories of congestion control mechanisms

Congestion control mechanisms are split into two groups in general:
Open-loop congestion control (prevention): In open-loop congestion control approaches are applied to forestall clog before it occurs. In these systems, congestion control is dealt with by either the source or the objective.
Closed-loop congestion control (removal): Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols.

Congestion Control in IOTA:

As because IOTA is based on a DAG rather than a blockchain, also, IOTA is feeless and doesn't have miners, so it needs to come up with a different solution. Furthermore, the ICCA uses a scheduler to select communications that have already been scheduled. Messages are regularly written to the local tangle and broadcast to the node's neighbors.
Now we may talk about how the ICCA meets our three main criteria:
i. Fair access: Network access must be offered in proportion to some "limited resource."
ii. Attack resistance: The network will not be disrupted by an attacker node.
iii. Consistency: To preserve consistency, all nodes must write the same messages to their local ledger.
On a local level, we can see that each node plans traffic fairly according to mana. It turns out that this is also true internationally, and that access is based on mana fairly.
Second, nodes will not process the attacker's messages faster than they have been authorized to. As a result, the attacker's inbox will fill up with messages, and their queue will grow. All nodes will notice this, and the attacker will be ejected off the network.
Finally, because the scheduler never deletes honest messages, the approach ensures that they reach all nodes.

2.1.9 IOTA Congestion Control Algorithm (ICCA):

The IOTA Congestion Control Algorithm (ICCA) simplifies the transaction process to reduce the impact of anticipated congestion and to control who has access to write to the ledger. The ICCA does this through three components: The following are the three key components of our congestion control:

Scheduler

The scheduler decides which messages should be written to the local tangle and transmitted to the node's neighbors. Messages are sent at a consistent rate, proportionate to the issuing node's mana. This method prevents any of the nodes from becoming overloaded.

Rate setter

In reaction to congestion events, each node uses unique rules (AIMD, inspired by TCP) to alter its issuance rate. Because traffic in DLTs goes through all nodes, local congestion at a node is all that is needed to indicate network congestion elsewhere. This finding is critical because it opens the door to a congestion-control algorithm that is purely focused on local traffic.

Black lister

If the node does not use the rate setting, their rate will not be reduced as their queue expands, causing their queue to grow even more. As a result, the ICCA employs a black lister to keep queue lengths to a minimum. When this number reaches a specific level, the node is temporarily blacklisted, which means no more transactions from that node will be put to the inbox for a period of time.

Rate control

The rate control system, as useful as it is, may not be required. The ICCA has the potential to be so powerful that this module will become obsolete. However, we aim to ensure maximal protection in the first version of the Coordicide protocol, thus any subsequent reduction or removal of any PoW is left as a future optimization. On the Go Shimmer test net, we'll look at how the Adaptive PoW Rate Control and the ICCA interact.

2.1.10 Auto Peering

Within the Particle convention, a node (or peer) could be a machine putting away the data almost the Tangle, IOTA's fundamental information structure. Nodes moreover commonly act as the passage point for getting to and utilizing the Tangle. In arrange for the organize to work productively, nodes trade data with each other to be kept up-to-date approximately the modern record state. Right now, a manual peering or association handle is used for nodes to enroll commonly as neighbors. Be that as it may, manual peering may be subject to assaults (e.g., social designing) to affect the arrange topology.

The point of the autopeering module overhaul is to ease simulations whereas keeping the same code base that will be utilized in GoShimmer. Normally, unused concepts and inquire about ought to be tried in an exploratory way in arrange to continue to the following level of execution in a convention. A critical step, subsequently, was to present a code base on which we are able experiment and test a few of our numerous speculations. Usually accomplished by actualizing our concepts of the

Coordicide diagram into a model code, which we call GoShimmer. Being able to evaluate the autopeering behavior and execution by means of recreations is exceptionally important for Particle. It permits replying a few questions, such as how numerous peering requests each node needs to send on normal some time recently getting acknowledged, how long an association is planning to final, how quick the convention meets and so on. In addition, it sets the ground for examining assault vectors in a controlled environment.

The analyst has consistently isolated the autopeering module into two fundamental submodules: peer discovery and neighbor selection. The previous is capable for operations, such as finding modern peers and confirming their online status. The last mentioned is dependable for finding and overseeing neighbors for IOTA's nodes. We have too typified the organize layer (P2P communication) and the capacity layer (continuing peer data) through the utilize of Go interfacing.

The neighbor determination and, in specific, the choice around which potential neighbors are best, are made on the premise of their remove. This separate work is based on the private and open salts, as characterized within the Coordicide white paper. As a following step, they will include Mana-depending separations.

As of now, the simulation can be arranged with the taking after parameters:

- N: the full number of peers
- T: the salt lifetime, in seconds
- SimDuration: the length of the simulation, in seconds
- VisualEnabled: the flip to enable/disable the simulation visualizer, open at http://localhost:8844 after beginning the simulation
- dropAll: the flip to enable/disable dropping all the neighbors at each salt update

The taking after liveliness has been recorded whereas running the simulation with the visualizer empowered. It gives us with a pleasant visual representation of the peering process:
Newly built-up joins between peers are highlighted in blue, with the asking and the tolerating peer appeared in blue and green separately. Dropped joins are highlighted in red. Currently, the test system bolsters the taking after measurements for which we offer assessment scripts:

- Convergence: the extent of peers that have the most extreme number of neighbors
- Link survival time: the likelihood that a given connect is still dynamic after a certain sum of time
- Message investigation: measurements almost the number of messages sent and gotten (peering demands, peering reactions and association drops)

Our objective is to execute an algorithm that provides:
1) great topological properties (i.e., properties related to the course of action of the nodes), to permit a great meeting for the voting component built on top of it
2) an irrelevant likelihood of an assault by a malevolent performing artist to be successful

In arrange to do that, the analyst presented an algorithm that combines three diverse factors; one which is unquestionable, one which is erratic, and the last one that's related to something scarce. Here, we are going to clarify how the autopeering process occurs after the node discovery is as of now wrapped up because it is modeled within the now shared test system. Each of the nodes will have set:
1) a public node id (a 32 bytes string)
2) a public salt (a 20 bytes string)
3) a private salt (a 20 bytes string)

The asking distance between nodes A and B is described as follows by analysts:
$d_req(A, B) = hash(node_id(A)) XOR hash(node_id(B) + pub_salt(A))$
To ask a modern association, node A will calculate $d_req(A, B)$ for all nodes it knows and will arrange the nodes by this remove. After that, the node will begin asking from the closest to the most remote, until k associations are acknowledged by the other nodes, and thus established.
They describe the tolerable distance between nodes A and B in the same way:
$d_acc(A, B) = hash(node_id(A)) XOR hash(node_id(B) + priv_salt(A))$
A node A will acknowledge an ask from a node B whenever

1. it acknowledged less than k requests
2. $d_acc(A, B)$ is littler than the acknowledgment separations to one of his acknowledged peers.

or In this case, node A will drop the association to its most distant acknowledged node.

2.1.11 Consensus Mechanism

A consensus mechanism is a fault-tolerant mechanism used in computer and blockchain systems to obtain the necessary agreement among distributed processes or multi-agent systems, such as cryptocurrencies, on a single data value or a single network state.
Consensus decision-making is a creative and dynamic technique for all members of a group to come to an agreement. Rather than merely voting for something and having the majority of the group have their way, a group that uses consensus is dedicated to finding solutions that everyone actively supports, or at the very least can live with.

Blockchain is a consensus mechanism

For any blockchain system to function effectively, consensus procedures are required. They ensure that all nodes are in sync and that the entire network of distributed node operators operates according to the same set of rules and conditions. Consensus mechanisms also protect blockchain users' privacy.
Each transaction on the Blockchain is considered completely safe and validated. This is only possible because to the consensus mechanism, which is an integral part

of any Blockchain network. A consensus algorithm is a mechanism for all peers in a Blockchain network to agree on the state of the distributed ledger at any given time. In this way, consensus algorithms provide blockchain network resiliency and build confidence among unknown peers in a distributed computing environment.

IOTA 2.0 Consensus Mechanism

The consensus approach in IOTA 2.0 is meant to work without the use of authorisation or a leader. It combines two voting protocols: 1. a binary voting protocol (OTV) for pre-consensus and metastability, and 2. a virtual voting protocol (AW) for finality, similar to the Nakamoto consensus's longest chain rule. For two reasons, the combination is essential. First and foremost, we anticipate OTV enforcing a common understanding of what is acceptable and inappropriate behavior. As a result, the approval weight is necessary to allow out-of-synch nodes to catch up.

Improvements

In this prototype, the current IOTA 2.0 DevNet consensus mechanism[12], which is based on FCoB and FPC, has effectively resolved hundreds of conflicts[12]. However, in order to create the greatest DLT, it would like to optimize protocol speed by reducing code, confirmation time, and communication overhead.
The FPC protocol may be altered in two ways. One is to change from FPC to Tangle FPC (OTFPC) Another is to use approval weight instead of direct queries. With OTFPC there is no requirement for a direct query mechanism because the tangle serves as the only medium of communication. This reduces the amount of bandwidth required to run a node by lowering the communication overhead.
Nodes will use FPC to select a winner from a list of competing transactions. Unlike FCoB, OTFPC does not employ quarantine periods, which could result in delays. Incoming messages are immediately added to the tip set[12], allowing every node to express its preference[12]. The local preference of a node is governed by the consensus weight associated with a given conflicting transaction[12].

2.1.12 Approval Weight (AW)

The acceptance weight of a message is the percentage of active consensus mana of nodes who issued a message in its future cone that approves it (i.e., by directly or indirectly referencing it)[12]. Furthermore, approval weight is calculated so that the consensus mana of one node cannot be used to determine the approval weight of two transactions that are incompatible. Assume A and B are two transactions that are incompatible. When I send a message identifying A as a high mana node, my consensus mana contributes to A's approval weight. However, if I issue a message later approving B, my consensus mana is deducted from A's approval weight and added to B[12].

When A's approval weight is larger than 50%, we know that B's approval weight is less than 50% [12].

As a result, a message (or a transaction) is considered complete when its approval weight exceeds 50% plus the strength of a potential attacker.

Hans' On Tangle Voting proposal included AW as a major tenet. Nodes would always choose tips referring the transactions with the highest approval weight in his proposal. However, it's unclear if this idea would work well if the approval weights of two competing transactions were tied, necessitating the use of a tie-breaker like FPC[12]. Consent weight is based on the concept of cumulative weight from the original white paper, but it has been tweaked to work in a world where mana is used as a Sybil protection mechanism.

2.2 Related Work

Purpose of this part is to provide review of previous work related to micropayment in ambience of blockchain. It also talks about secure cashless small payments. Here we analyze different aspects of blockchain to resolve the issue of micropayment. Here we analyze two frameworks, Ethereum and IOTA Tangle and their effects on blockchain based micropayment. And how IOTA tangle is solving cost issue of blockchain based micropayment as well as how it is also ensuring security of the whole transaction system. IOTA tangle is also resolving issue regarding speed that Ethereum or bitcoin blockchain technology usually face. Micropayments are related to the payment scheme which enable the small payments. It is a kind of transaction which deals with small amount of money ranging from less than a dollar to around five to 20 US dollars. So, the amount of micropayment varies. This paper provides a solution that a normal micropayment face during transaction. It solves the transaction speed due to the computation and verification of transaction. This feature includes the verification of micro coins do not require any digital signature. This paper provides a solution where the payer commits a total amount of payment to the payee say a bitcoin X. Then the payer can generate micro coins from X and pay the payee with micro coins for each micropayment transaction. So, here the payer commits a particular amount of bitcoin to the payee. This scheme provides security as all the micro coins are linked to the commitment; the verification is done by hashing. Each micro coin only needs to store a hash value and no sign in is required for each transaction. It also saves the communication cost during the micropayment [7]. For economic system, blockchain is a secure means for asset transfer and get an attention of the global economic community. Nevertheless, there are some challenges like scalability, which is strongly hampered the growth of economic systems based on the blockchain technology [7]. For overcoming this problem, a suitable design for a scalable and cost-efficient sophisticated routing protocol based on bidirectional micropayment transactions has been promoted. Every node of this transactions charges minimum fee for reducing original cost in sending transactions between parties. Moreover, another challenge is inefficient routing mechanisms and timelocks generated per channel. As a result, there is a developed suitable routing algorithm for economic systems also designed. It is claiming on analyzing the simulation on different sections of the design which is suitable for the micropayment network[7]. Two frameworks are considered for the blockchain based micropayment. Ethereum and IOTA Tangle.

2.2.1 IOTA Tangle

The Internet of things technology that connects sensor, control and machinery which will help humanity to move forward. But public blockchain has some major problems particularly IoT, with high transaction cost and lack of scalability. To overcome this problem integrating tangle into IoT blockchain for building a cross chain decentralized access model for better privacy, scalability with zero transaction cost. The notary mechanism to maintain cross chain network and apply IPFS and BigchainDB to solve the data storage and device tagging problem. For the privacy issue they proposed data access control model and special transaction will be designed **POPOV2019160**. With their experiment data it is claimed the framework is more efficient for IoT devices with less resource, among multiple consortia than blockchain structure.

2.2.2 Mana

We utilized IOTA 2.0 for our research, and Mana is introduced as a tool that may be used in a variety of functions for the IOTA network and the IOTA token. When a transaction containing a value amount called Mana is executed, it is ”pledged” to a given node ID.[11]. In a transaction, a given amount of iota tokens are transferred from one address to another address, a node selected by the issuer is pledged with Mana or trust. This number refers to the amount of IOTA that was sent in the transaction. The Mana pledged to each node ID will be saved as a ledger extension. Mana can only be obtained by persuading a token bearer to donate it. For this reason, Mana is Delegated Proof of Token. For this reason, Mana is a parallel reputation token to the IOTA token that is held by addresses at a rate proportional to the stake they hold. Pending Mana will be pledged to nodes becoming Mana on value transactions. The amount of Mana this node will get is proportional to the number of iota tokens sent on the transaction[4].
The main goal of Mana is to grant nodes participating in the IOTA network with a ranking or reputation System that will allow us to distinguish honest working nodes that have a validated history from new nodes. Mana can be pledged to the node to issue an IOTA tokens transaction, but it can also be sent to other nodes. Pending Manas are generated at a rate proportional to the stake they hold Mana. When IOTA tokens are spent from an address, the pending Mana generated by the address is converted to Mana. Mana will be pledged to a node in the future. The monies on the receiver’s address are now generating pending Mana. Mana and pending Mana both decay at a pace proportional to their value, preventing Mana from becoming out of control over time. The mana lifecycle starts with pending Mana on every funded address, proportional to its balance. Say that address A sends 10 Miotas to address B. The full node chosen by the issuer in this operation will then get a proportional amount of Mana pledged. This way, nodes will stake Mana over time as they work for the network, but they will continuously lose some mana given the decay mechanism.
The deployment of the IOTA 2.0 upgrade includes two methods for calculating

Mana[11]. The first method of calculating Mana is known as ”mana 1,” in which the pledged Mana is equal to the amount of tokens transferred in the transaction. Mana 2 is an upgrade of Mana 1 that adds delegated evidence of ownership and proof of node activity. Because it is unaffected by subsequent token transfers, Mana 2 evolves in a predictable way throughout time.. This predictability may be required by users participating in a ”access mana market” who wish to keep control of their purchased or leased access. In the IOTA network, participants will be automatically allocated Mana in IOTA 2.0, depending on the number of their credits in IOTA.Then there will be two types of Mana:

1. Consensus Mana is the abbreviation for Consensus Mana. With these tokens, network points vote on which transactions should be declared genuine and distributed on a regular basis. If there is a disagreement, the goal is to find a speedy agreement by consulting neighboring network points. IOTA compares this to how flocks of birds naturally organize themselves in flight, where the collective selects where to go without the need for a leader.

2. aMana is an abbreviation for Access Mana. This form of Mana is utilized to get the ability to initiate and validate transactions. Priority can be obtained through aMana if a situation happens in the IOTA 2.0 network where a queue forms. Coordinate is done by forcing every node to create a node identity in the Peer Discovery.Because generating an arbitrary large number of identities is not an expensive activity, each node identity is tied to two Difficult-to-obtain resources: access Mana(aMana) and consensus Mana (cMana). Both types of Mana can be thought of as vital resources for certain areas of the network.
When a transaction is completed, it generates a particular amount of aMana and cMana, which is proportional to the number of IOTAs exchanged. Each node ID's access and consensus Mana must be recorded as an extension of the ledger state. A node can only obtain aMana or cMana by persuading some ken holders to pledge to it.
A Sybil protection mechanism is required for any permissionless system. Mana can only be obtained by persuading a token holder to pledge it in a transaction. Mana is Delegated Proof of Token Ownership in this scenario[11]. Varied modules with different natures and requirements utilize access and consensus Mana as Sybil defense. As a result, it's only reasonable to employ multiple formulas to get the correct Mana for each module. Consensus Mana is the Mana in charge of the system's security; on the other hand, access Mana is in charge of distributing network access during times of congestion. Mana is difficult to acquire in arbitrary amounts. Through the formation of several identities, The Sybil defense prevents an attacker from exerting excessive control over the network. Although numerous factors such as network utilization have an impact on a node's message quota, a node's Mana dictates in terms of the total network throughput, how many messages it can transmit [11].
To provide maximum freedom and security in the network, The pledge method is completed twice, first for the consensus modules and again for the congestion control modules. Because the incentives for security and access may be at variance, this separation guarantees that users may always behave in the best interests of the network while maintaining their ability to assign access according to their economic interests[11]. The token holder can thus delegate their access without giving the

delegate any additional ”weight” on the consensus process.
The weight of a node’s consensus and access mana is proportional to the total ”active mana” (Mana held by active nodes) in the network. In a hypothetical circumstance, a node owns 5% of the total access mana, but only 50% of it is active. Tanglecite-Mana will receive 10% of the total data allowed by the protocol from each node. Voting power is proportional to active consensus mana in the same way. For example, the more consensus mana a node has, the more FPC inquiries it receives. The more voting power it possesses, the better. Similarly, the top active mana holders issue the random numbers in the dRNG. Although access mana ensures minimal network access, total active Mana determines the actual access allowed.

Chapter 3

Methodology

3.1 Website

The purpose of the proposed IOTA tangle for micropayment system is for secured cashless payment. In previous section, we have already discussed something about the IOTA tangle and micropayment system. As well as, we have mentioned the security system of micropayment using IOTA tangle. The model requires designing the process that will be working in two different methods for transactions. One is web-based, another one is NFC system for our secured payment. The NFC system has some difference. To do so, we design two diagram to describe the method how our system will work.

The Figure 3.1 provides a basic idea about web view. Firstly, the system will start with a server. According to the input, the system will choose user type.Then the user get the option for choosing Request Token of Make Transaction. If he choose Request Token he/she will get token form IOTA Faucet and if he choose make transaction he will make transec through Firefly Wallet.
From the API, the transaction data will forward to the IOTA framework. In IOTA framework, we all know when a new transaction come it will be introduced as a node. According to IOTA algorithm, if a new node come it must be verified by other two nodes that are already in the IOTA tangle. The verification process is completed by solving puzzle. After verification, the new node will be a part of the tangle. Then the node will be ready for transaction. After that, IOTA will validate the payment securely. Then it will go to the system and system will check if the transaction is successful or not. If transaction is not successful, it will go the reject transaction stage. From the reject transaction stage, it will go to the API again.
If the transaction is successful, the transaction record will be recorded in the "public ledger" which is a cloud base storage of our system and ended the system. In this way web API will be worked for our system.
On the other hand, if the data is valid it will go to the API. Then it will work same as our web API. Briefly, API will send data to the IOTA Framework. Then the framework will complete its process and complete the transaction. If the transaction is valid, it will send it to transaction successful state and after that save to cloud and end. Otherwise, it will do the same from the API state for invalid transaction. All process is showed in the Flowchart below:

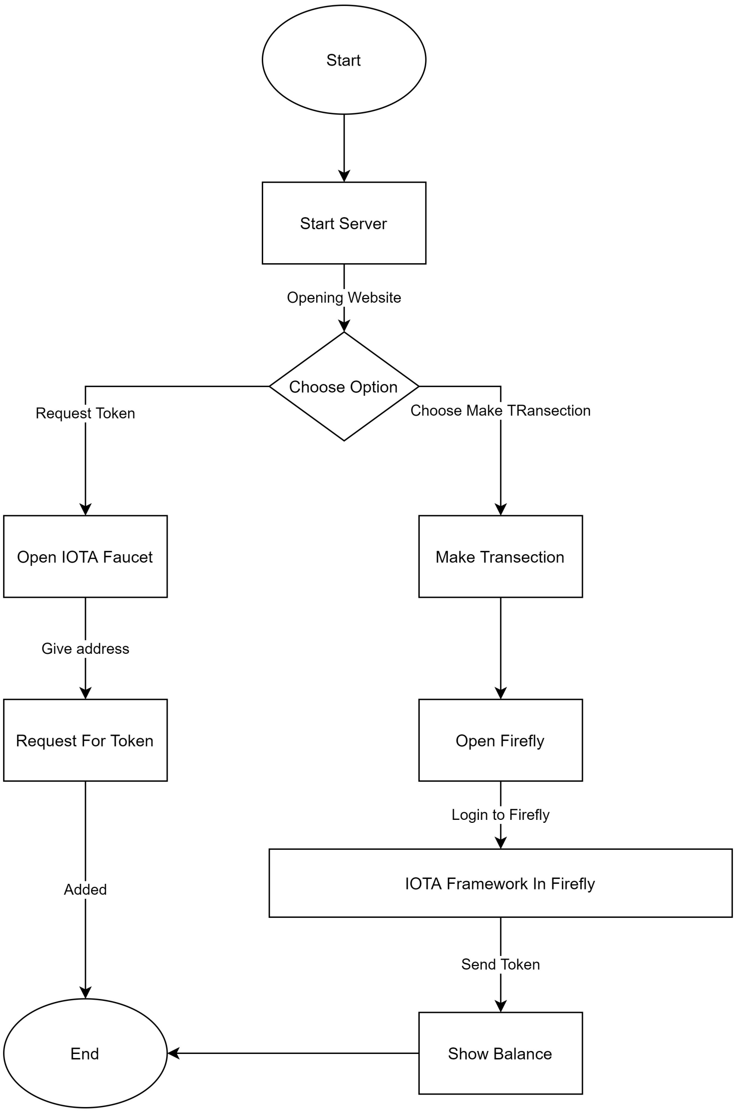


Figure 3.1: Website Working Flowchart

3.2 NFC

In the NFC part the user put his card on the NFC device (RFID). Then if the card is valid it will go to the Firefly Wallet. After making Transaction It will show the balance.
All process is showed in the Flowchart below:

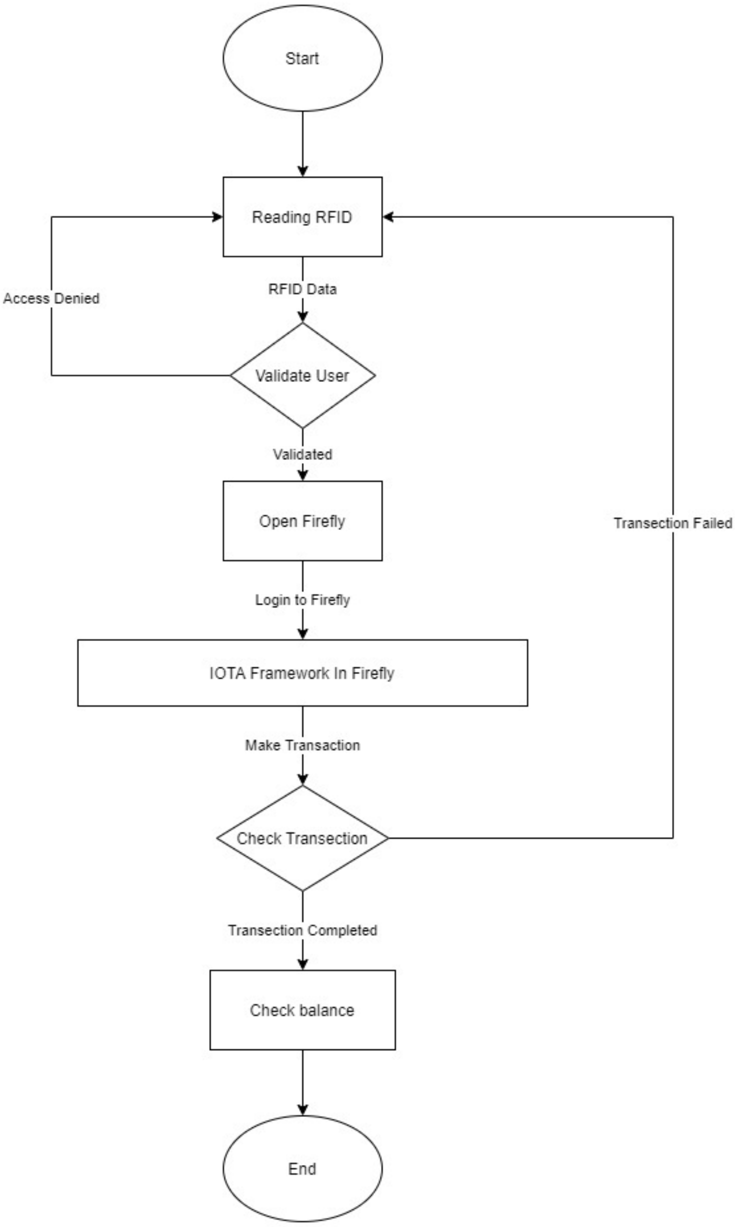


Figure 3.2: NFC Working Flowchart

3.3 Q- learning algorithm for Message Overhead

As we have seen that, due to gossip protocol, IOTA is broadcasting new message information to all of its nodes. Which is not optimized at all. And this section is open for future search. So, we have come up with a solution. To reduce message overhead, we are using the Q-learning algorithm. So, the basic idea is to refer to nodes with high mana. We are choosing nodes that has mana greater than average of mana of all of it's neighbors. And then with the help of gossip protocol, IOTA will send information of a new message to all of it's high cMana based nodes which are chosen as peers. So, here, they are not broadcasting information to all of it's nodes. Instead of that, it is multicasting. Now, the network needs to learn to choose these high cMana nodes. In order to let the network learn by itself to choose these nodes, we are using Q-learning algorithm. And as a part of Reinforcement Learning, where the reward function is, cMana.

3.4 Reinforcement Learning

Reinforcement Learning is a learning process with goal[2] to learn good policies for sequential decision problems, by optimizing a cumulative future reward signal. The learning process of reinforcement a computational learns to behave optimally in a given environment to get as many as reward as possible by interacting with it continually. The key entities of interest are the environment the action reward and the state. This whole paradigm of exploring the environment and learning through actions rewards and States establishes the foundation of reinforcement learning. In the background, the agent experiences a range of scenarios during its learning process. States are what we call them[3]. While in that state, the agent can choose from a set of approved actions, each of which can result in a different reward or penalty, which the learning agent learns to optimize over time to behave optimally in any given environment.

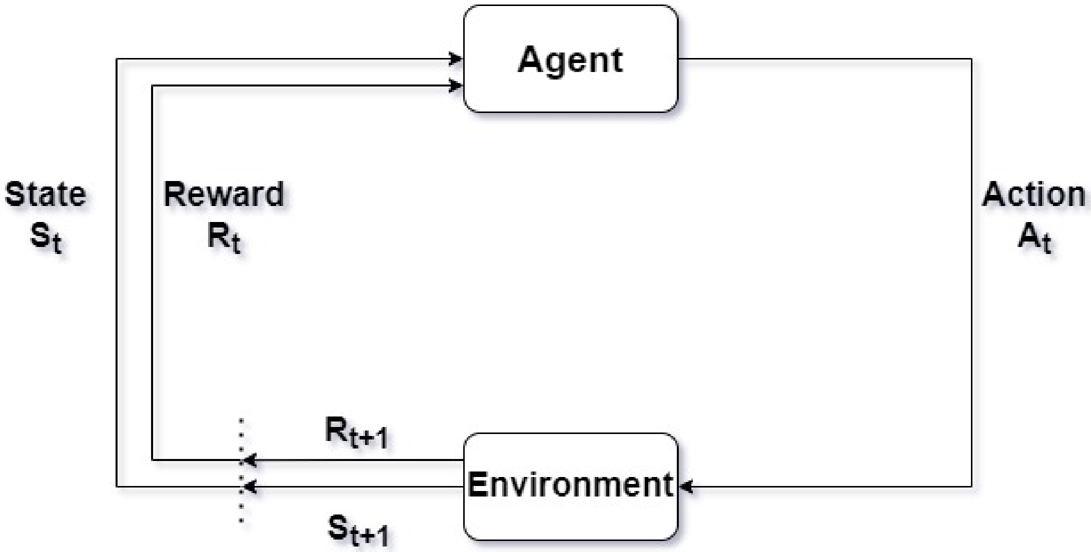


Figure 3.3: Workflow of Reinforcement Learning

3.5 Q-Learning

Q-learning is a type of model-free reinforcement learning in which Q-values, also known as action values, are used to iteratively improve the learning agent's behavior[1]. By continuously performing all actions in all states, it learns which are the best overall, as measured by long-term discounted reward.

Action-Values or Q-Values: Consider a computational agent navigating through a discrete, finite world, selecting one action from a finite set at each time step. With the agent as the controller, the controlled process is done using Markov process. For states and actions, Q-values are defined. Q(S, A) determines how effective it is to take action A given the current situation S. In the TD-Update technique, an agent performs an action within a state and analyzes the consequences in terms of the immediate reward or penalty it receives, as well as its estimate of the state's worth. It will be utilized to iteratively compute this estimation of Q, which we will look at in the next sections (S, A).

Episodes and Rewards: An agent begins in a start state depending on the actions it does and the environment in which it interacts and transitions from one state to the next during the course of its lifetime. Each state's agent performs an action, receives a reward from the environment, and then moves on to the next. No more transitions are allowed if the agent reaches one of the end points. This is the point at which an episode is declared to be completed.

TD-Update or Temporal Difference: The following is a representation of the Temporal Difference or TD-Update rule:

$$Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma Q(S', A') - Q(S, A)) \tag{3.1}$$

This update rule is applied at every step of the agent's interaction with the environment to estimate the value of Q. The following terms are defined:

1. S: The agent's current state.
2. A: The current action has been chosen by some policy.
3. S': The agent's next destination state.
4. A': Pick the following best action based on current Q-value estimation, i.e. choose the action with the highest Q-value in the next state.
5. R: Current Reward as seen in the environment due to the current action.
6. γ : (>0 and <=1) : Future Reward Discounting Factor. Future incentives must be discounted since they are less desirable than current incentives. Because the Q-value estimates expected rewards from a state, the discounting rule also applies.
7. α : The time it takes to update the estimation Q (S, A).

Selecting an Action to Take with the ϵ-greedy Policy:

ϵ-greedy Policy is a specific policy for selecting actions based on current Q-value estimates. It goes like this:

1. Choose the action with the highest Q-value with probability (1-ϵ).
2. Choose any action at random with probability (ϵ).

3.5.1 Recommended Requirements:

Nodes need to maintain enough computational power to run reliability, to handle potentially high rate of messages per second. The followings are the minimum specsfor running a node discountedireward:

1. 4 cores or 4 vCPU
2. 8 GB RAM
3. SSD storage

3.5.2 Q-learning

Q-learning based Configuration

In MDP, we have a set of states S, and a set of actions A and a set of rewards R. At each time step t = 0, 1, 2, 3, ... agent receives some representation of environment's state $S_t \epsilon\ S$. Based on this state the agent selects an action A$_t \epsilon\ A$. And this give us the state action pair(S_t, A_t).

Time then increased to the next stem as t+1 and the environment is transitioned to a new state S_{t+1} ϵ S. At this time the agent receives a numerical reward R_{t+1} ϵ R for the action A_t taken from the state S_t.

We can think of the process of receiving reward as an arbitrary function which maps the state, action pairs to rewards. At each time t, we have:

$$f(S_t, A_t) = R_t \tag{3.2}$$

The agent wants to not only increase immediate reward but also cumulative reward.

Considering a 7*7 matrix where there are 7 nodes. 7 nodes indicates 7 agents. Here among the 7 computational agents, one agent move around the some discrete and finite world. It is choosing one actions among collections of actions at every step.

Some constants to consider:
Episode number = 1000, epsilon = 0.9, $Episode_decay$ = 0.998, learning rate = 0.1 and discount as 0.99. Here the set of states where s ϵ **S**, are nodes. For example,

customer X wants to do the transaction, according to his territory he got node 6 as a tip. So node 6 is selected and for the further transaction node 6 will be used as root node or starting node. And node 6 got 5 neighbors. These state space can be defined as an array with cMana as environment state:

$$[0, 100, 45, 100, 100, 10, 0]$$

Now agent will randomly choose any action of the given state because at first it got no idea of the given environment. So the agent needs to explore the given environment. After a time, when the agent has enough exploration then it will choose the best action according to the q-table and we get a state, action pair(S$_t$, A_t).

Q-table: :
We are filling up the q table with negative numbers. If the choosen action caused an index or neighbor with mana >= to average mana then the agent is receiving a reward at that time t.
Agents goal is to maximize the expected discount return of rewards. The agent will be choosing an action at each time step in order to maximize the expected discounted reward.
We can define return G with time t as:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T \tag{3.3}$$

To define the discounted return, we need to define the discount rate, γbetween 0 and 1. Here we are choosing discount = 0.99. Discount rate will be the rate which discounts future rewards and will determine the current value of future rewards. We can define discounted reward as:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \\ &= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \end{aligned} \tag{3.4}$$

Returns at successive time steps are related to each other. We can define this as:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + ... \\ &= R_{t+1} + \gamma(R_{t+2} +_t {}_{+3} + \gamma^2 R_{t+4} + ...) \\ &= R_{t+1} + \gamma(G_{t+1}) \end{aligned} \tag{3.5}$$

The return at time t is a sum of an infinite number of terms, the return is actually finite as long as the reward is between 0 and 1 and constant. For example, is the reward at each time step is a constant 12 and (gamma) <1 then the return is:

$$G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma} \tag{3.6}$$

Policies:
Policy is a function that maps a given state to probabilities of selecting each possible action from that state.
Policy is denoted as (pi). An agent follows a policy.
If an agent follows policy (pi) at time t, then pi(a|s) is a probability that A$_t$ $= aif S_t = s$.
This means that, at time t, under policy (π), the probability of taking an action a in state s if $(\pi)(a|s)$.

Value functions:
Value functions are function of state, action pairs which estimates how good it is for an agent to perform an action at a given state[14]. It depends on expected returns. The reward an agent expects to receive depends on which action the agent takes in which state. And value functions are defines with respect to policies. There are two types of value function:

1. State-value function:
   The statevalue function for policy π, donated as v$_\pi$, tells us how good any given statre is for an agent following policyπ. In other words, it gives us the value of a state under π.

$$\begin{aligned} v_\pi &= E[G_t|S_t = s] \\ &= E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \end{aligned} \tag{3.7}$$

2. Action-value function:
   Similarly, the action-value function for policy π , donated as q_π, tells us how good it is for the agent to
   tion from a given state while following policy[14] π. In other words it gives us the value of an action under[14] π.

   Formally, the value of action α in state s under policy π is the expected return from starting from state s at time t, taking action α, and following policy πthereafter. Mathematically, we define q$_\pi(s,a)$ as

$$\begin{aligned} q_\pi(s,a) &= E[G_t|S_t = s, A_t = a] \\ &= E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a] \end{aligned} \tag{3.8}$$

here q(π) (s,a) is the Q-function and the output from the function at any given state, action pair is called Q-value [14]. The letter Q represents the quality of taking a given action in a given state.

Optimal policy:
The goal of Reinforcement Learning algorithm is to find a policy [14] that will return a lot of rewards for the agent if the agent follows the policy[14]. For example, a policy $\pi$is considered to be better than or equal the same as policy π':
π>= π' if and only if v$_\pi(s) >= v_\pi(s)$ for all sϵS.

A policy that is better than or at least the same as all other policies is called the optimal policy. Again there are two optimal value functions:
1. Optimal State-Value Function and
2. Optimal Action-Value Function
Here we will discuss Optimal Action-Value Function[14]. The optimal policy has an optimal action-value function, or optimal Q-function, which we denote as q$*$ and define as
$q(s, a) = max_\pi q_\pi(s, a)$ for all s ϵS and a $\epsilon A(s)$. In other words, q gives the largest expected return achievable by any policy π for each possible state-action pair. So, after selecting an action, according to the according to the optimal action value function which will take state as an input and will return the largest reward for that particular state and action from q-table.

So according to Bellman equation, the equation is:

$$\text{Q'}(s_t, a_t) = r(s_t, a_t) + \gamma_{at+1} max\text{Q'}(s_{t+1}, a_{t+1}) \tag{3.9}$$

And according to the bellman equation, we are updating new q-value. We are following this procedure for all neighbors but ultimately the agent will learn which node to select by selecting the high cMana neighbors through q-learning algorithm. If still the agent won't get the sufficient mana then it will check for it's neighbor's (efficiently selected) neighbor for requesting mana. And if the mana is sufficient enough then it will stop visiting neighbors and customer's transaction will be completed.

Overall process is showed in the Flowchart below:

Algorithm 1 Algorithm:

```
value = Select any node
$iota = qLearning(mana, manaArr[value])$
**if** $IOTA < sufficientMana$ **then**
    $manaArr = (arr[i] >= average)$
    $qLearning(iota, manaArray)$
    $got\ sufficient\ mana$
**else**
    $break$
**end if**
```

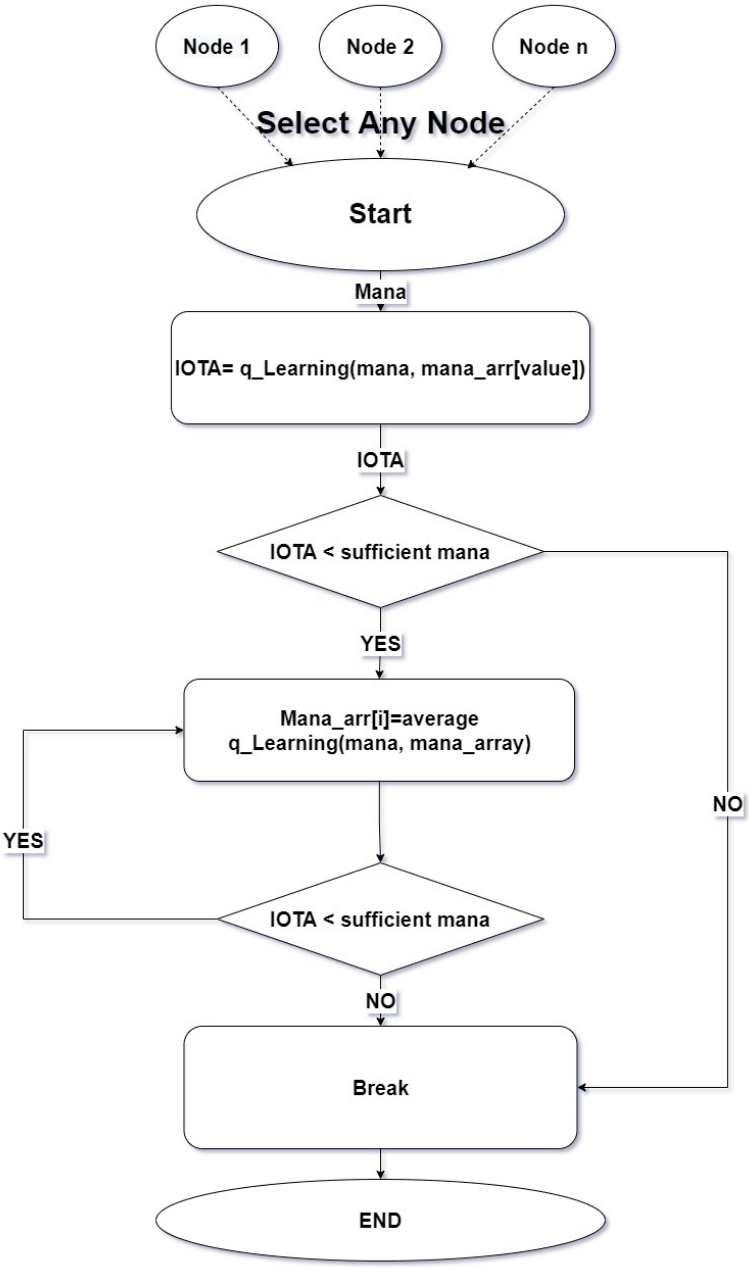


Figure 3.4: Algorithm working Flowchart

Chapter 4

Implementation and Result Analysis

4.1 Implementation

IOTA technology provides an immutable, transparent audit trail that builds trust across the entire network. And IOTA's Tangle is an open, feeless, and scalable distributed ledger which is designed to support frictionless data and value transfer [16]. It is an innovative type of distributed ledger technology (DLT) which is specifically designed for IoT (Internet of Things) environment [17]. It is a protocol for IoT and a scalable DLT. IOTA is a cryptocurrency, and the architecture is called IOTA Tangle. It solves the design limitation of the traditional Blockchain technology through promising of high scalability, no fees, and near-instant transfers [17]. Tangle starts with an alpha node which propagates transactions and through connecting with other nodes it creates a network. And the rule is that any new transaction needs to approve previous 2 transactions or sites. Each transaction can get a personal weight and they hinder successful attacks in the network because as the transactions or sites become older, they are getting stronger or gain cumulative weight [18]. This section describes implementation of the proposed model of Blockchain-based micropayment system for secured cashless small payments. The implementation part is divided into 2 separate sections.
1. Website
2. NFC

4.1.1 Website

We are making our website using HTML and python. And the home of our website is showing the interface like this. To connect the website and server we have to use flask in our python code. Here we are using iota client library to connect our website with the IOTA tangle. To confirm our connection with iota tangle we have to add the global client link https://api.lb-0.h.chrysalis-devnet.iota.caf. Confirming the connection with iota tangle uses IOTA SEED SECRET as environment variable and the value is our address.

For making the transaction we have to click the Make Transaction button then the

Firefly wallet open and the user give his details to make the transaction. The Address of the receiver will automatically copy to clipboard. So that the payeer can easily paste and pay the virtual currency Mi. The website then shows the previous balance and then current balance after the transaction. If the balance of the merchant is increased the payment will be successful.
The processes image are given below:

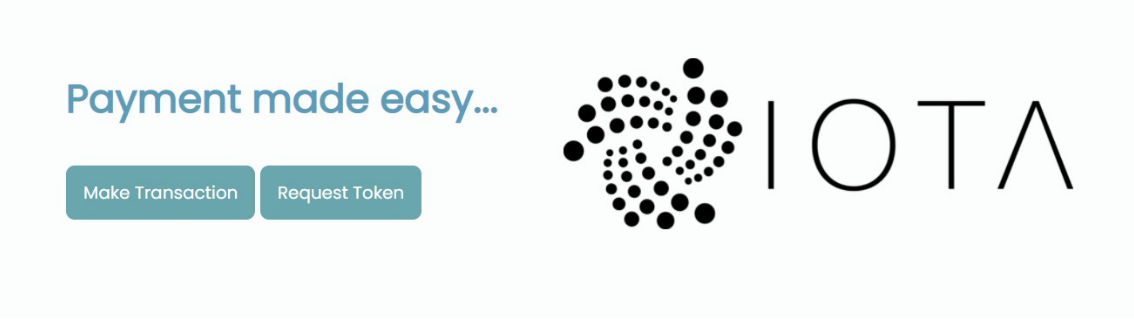
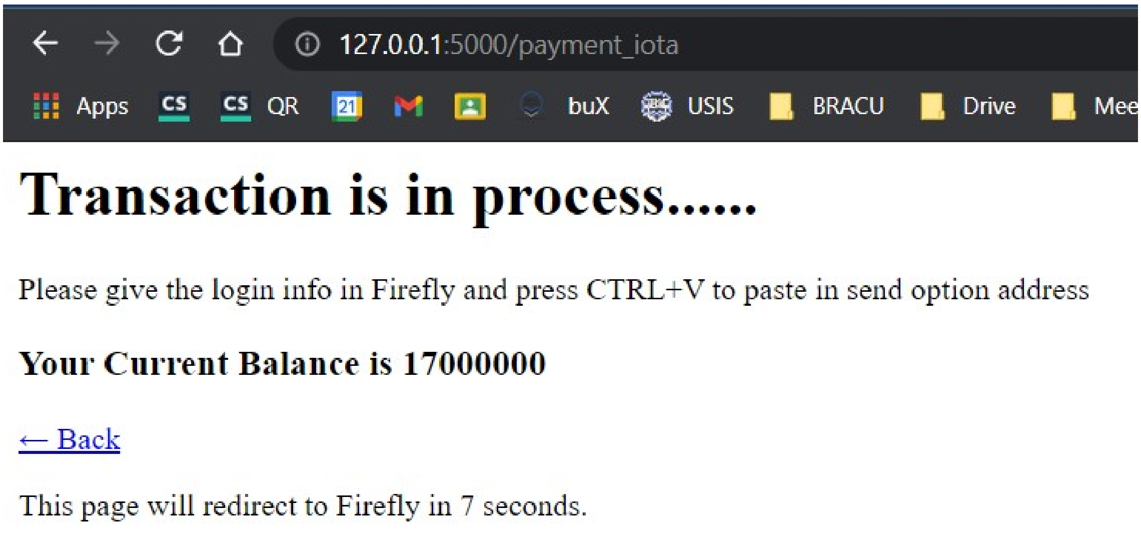


Figure 4.1: Website Home

From Figure 4.1 User can choose make transaction to payment.



Transaction is in process......

Please give the login info in Firefly and press CTRL+V to paste in send option address

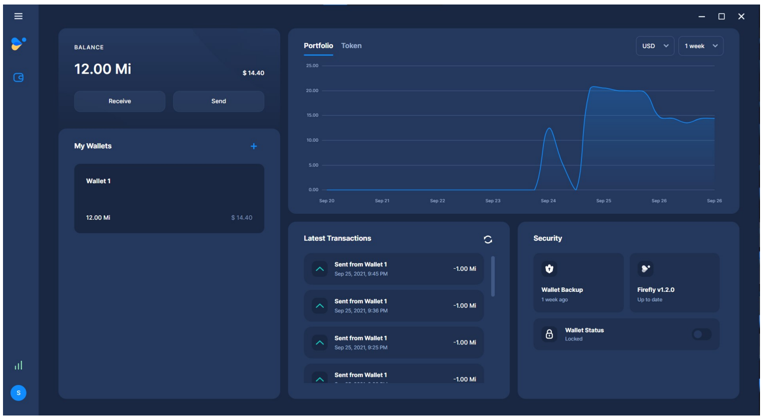
Your Current Balance is 17000000

← Back

This page will redirect to Firefly in 7 seconds.

Figure 4.2: Opening Firefly

Here user have to wait for 10 seconds to open Firefly wallet. Here the merchant can see the balance.
In figure 4.3 we can see the firefly interface. That is the user view.
In the figure 4.4 the user paste the address and amount then make the payment by clicking send.Then the user should log out from firefly.
In figure 4.5 the Merchant can the balance. From the website.


BALANCE

12.00 Mi $ 14.40

My Wallets

Wallet 1

12.00 Mi $ 14.40

Latest Transactions

Sent from Wallet 1 Sep 25, 2021, 9:45 PM -1.00 Mi

Sent from Wallet 1 Sep 25, 2021, 9:36 PM -1.00 Mi

Sent from Wallet 1 Sep 25, 2021, 9:25 PM -1.00 Mi

Sent from Wallet 1 -1.00 Mi

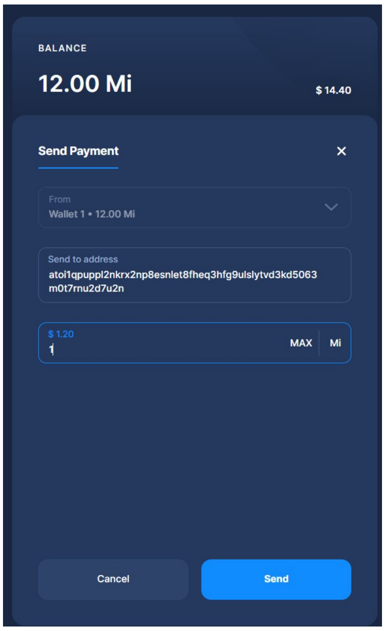
Security

Wallet Backup 1 week ago

Firefly v1.2.0 Up to date

Wallet Status Locked

Figure 4.3: Firefly Interface


BALANCE

12.00 Mi $ 14.40

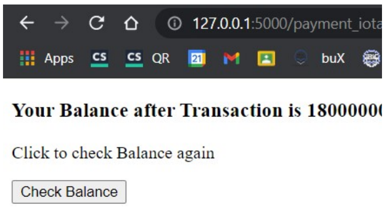
Send Payment

From
Wallet 1 • 12.00 Mi

Send to address
atoi1qpuppl2nkrx2np8esnlet8fheq3hfg9ulslytvd3kd5063m0t7rnu2d7u2n

$ 1.20
1

Figure 4.4: Make Paymet


127.0.0.1:5000/payment_iota

Your Balance after Transaction is 1800000(

Click to check Balance again

Figure 4.5: Make Paymet

The algorithm of joining with IOTA is:

Algorithm 2 Join with IOTA

```
initialize client variable
if No IOTA SEED SECRET then
    Initialize IOTA SEED SECRET
    Open Firefly
else
    Open Firefly
end if
```

4.1.2 NFC

In our research we are making a NFC device for transection. Here we are using RFID-RC522 reader and writer via Arduino UNO module. For the testing purpose we are using one NFC(RDIF) card and one NFC (RFID) tag (Figure-3) . Firstly we wrote the Arduino machine code in Arduino IDE then compiled and installed it in our UNO module . Then we connect our NFC device with the IOTA tangle through Python.

The Algorithm what we are using to read the RFID Data.

Algorithm 3 Checking UNO device

```
Make Object
if No IOTA SEED SECRET then
    Initialize IOTA SEED SECRET
    RFID Data = readLine()
    Decode arduino Serial
    Strip Arduino Data to remove string
    Convert the Data to Int
else
    Read RFID data Again Open Firefly
end if
```

In both website and NFC we are representing a scenario where the computer device is our Merchant point and the Firefly Wallet that is opened is for the payee. The website and the NFC console will show the balance of merchant and payee will use only the Firefly Wallet like Shopping card pose machine.

To connect with the IOTA tangle we use the iota client library and for joining the Arduino with python we use the serial library. For joining the client for our testing purpose we use 'https://api.lb-0.h.chrysalis-devnet.iota.cafe'. It is the DevNET global link for researching purposes. Confirming the connection with iota tangle uses IOTA SEED SECRET as environment variable and the value is our address. After connecting for a transaction we use the official Wallet of IOTA that is Firefly. Though the Trinity Wallet is restricted for the testing purpose we are choosing Firefly Wallet. From there we add our Virtual currency from https://faucet.devnet.chrysalis2.com/.

Then we are able to make our transaction. If the NFC Card is valid the transaction will be processed by giving the necessary information. If the NFC card is not added with the proper validity it just denies the transaction.
The whole devices and codes of our NFC device is given below:

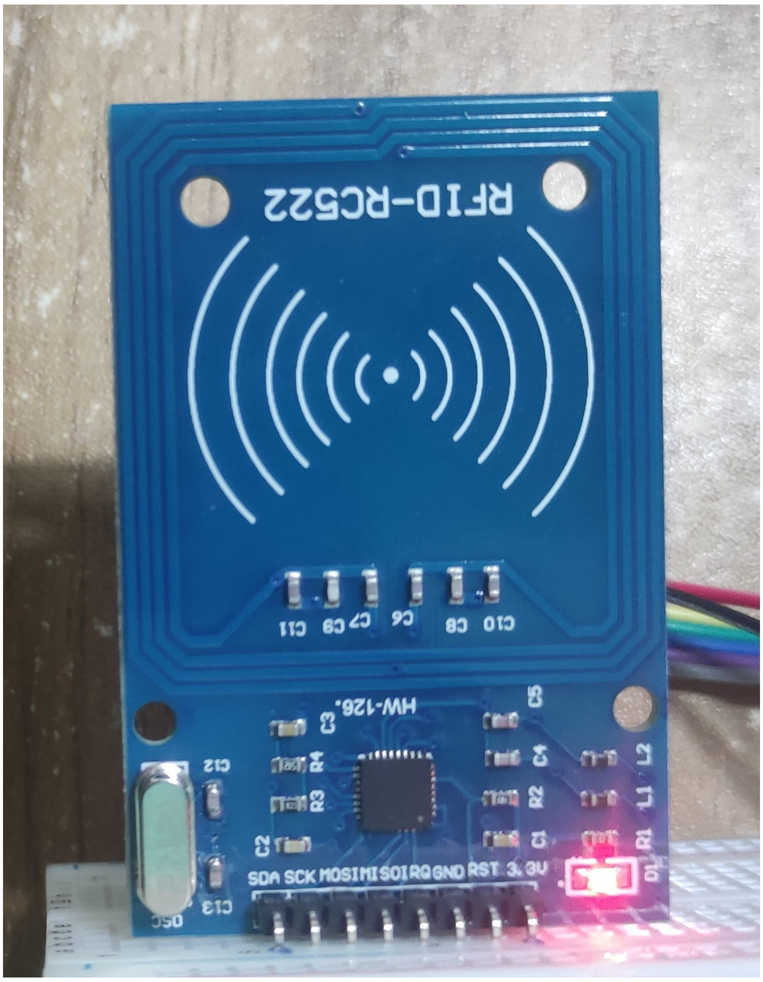


Figure 4.6: RFID-RC522

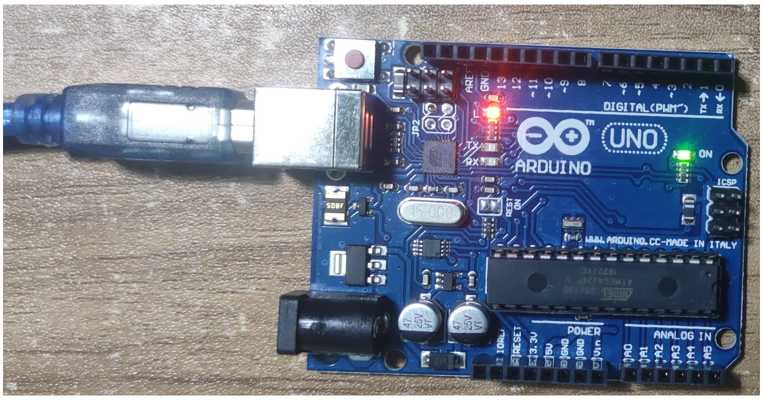


Figure 4.7: UNO Module

Figure 4.8: RFID Card and TAG

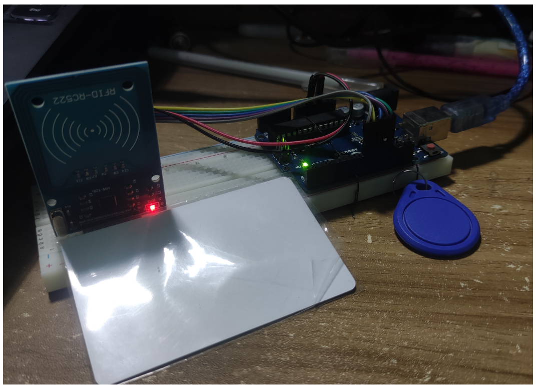
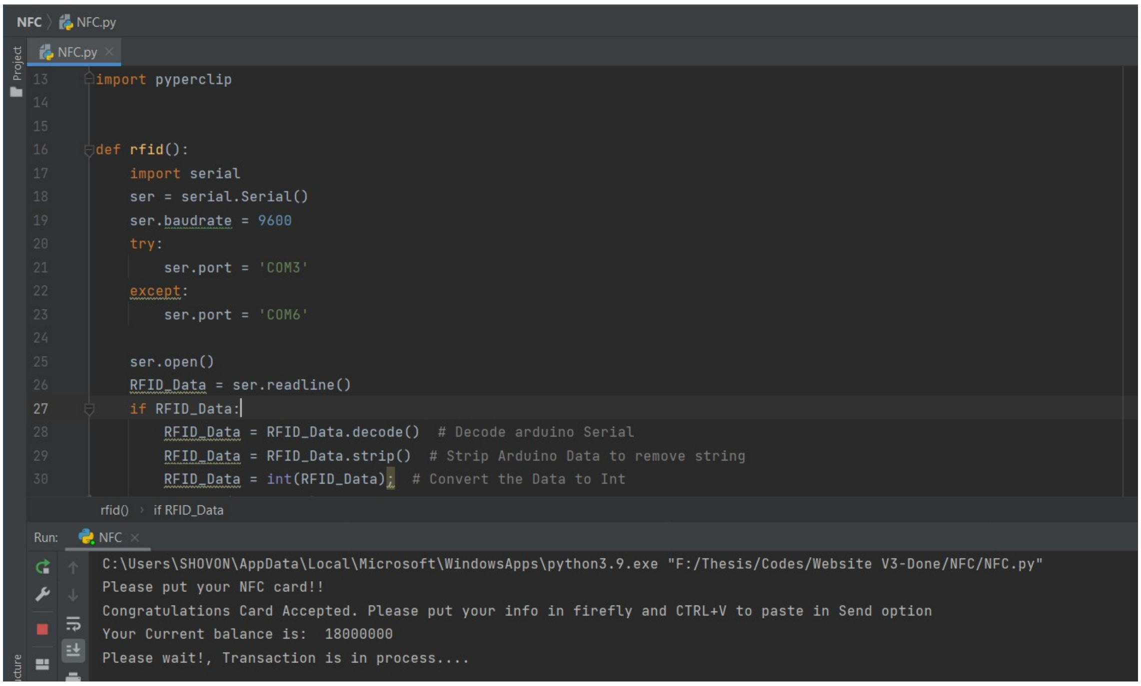


Figure 4.9: Complete NFC Device



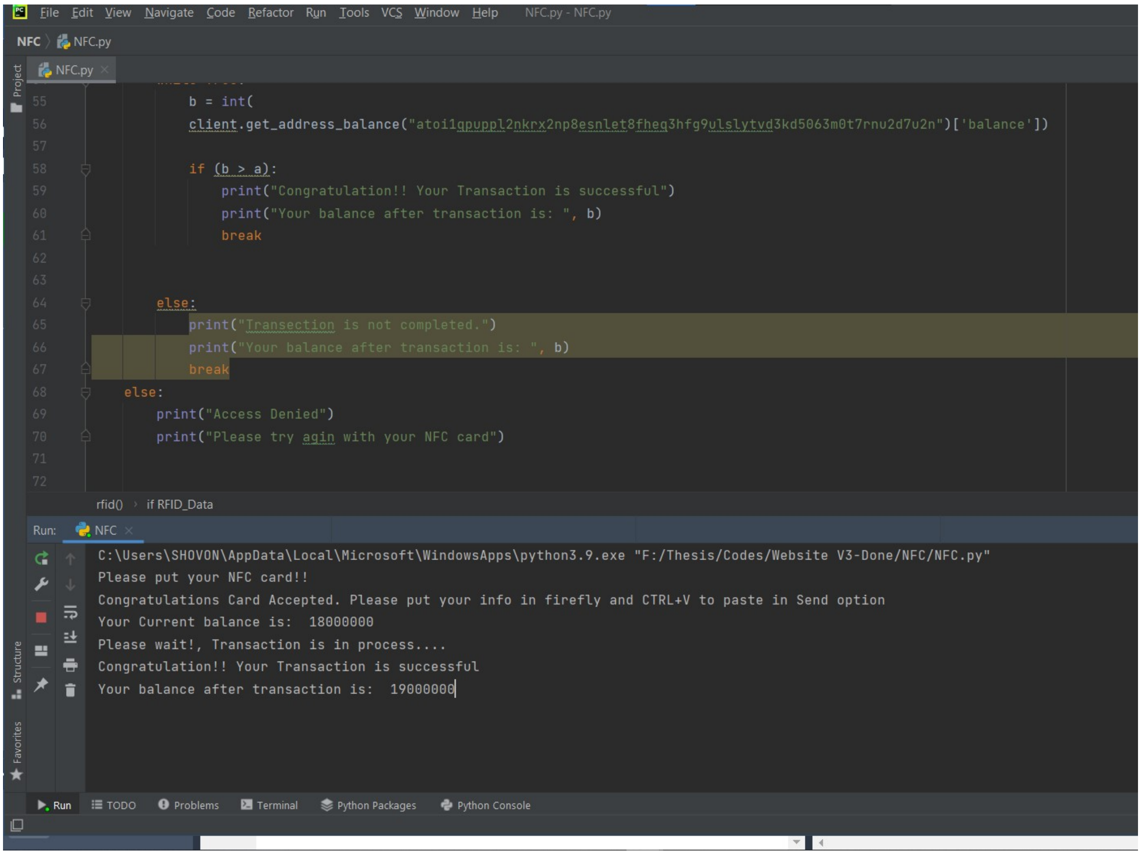
```python
import pyperclip


def rfid():
    import serial
    ser = serial.Serial()
    ser.baudrate = 9600
    try:
        ser.port = 'COM3'
    except:
        ser.port = 'COM6'

    ser.open()
    RFID_Data = ser.readline()
    if RFID_Data:
        RFID_Data = RFID_Data.decode()  # Decode arduino Serial
        RFID_Data = RFID_Data.strip()  # Strip Arduino Data to remove string
        RFID_Data = int(RFID_Data);  # Convert the Data to Int
```

```
C:\Users\SHOVON\AppData\Local\Microsoft\WindowsApps\python3.9.exe "F:/Thesis/Codes/Website V3-Done/NFC/NFC.py"
Please put your NFC card!!
Congratulations Card Accepted. Please put your info in firefly and CTRL+V to paste in Send option
Your Current balance is:  18000000
Please wait!, Transaction is in process....
```

Figure 4.10: Running and waiting for NFC (RFID) Card



```python
            b = int(
            client.get_address_balance("atoi1qpuppl2nkrx2np8esnlet8fheq3hfg9ulslytvd3kd5063m0t7rnu2d7u2n")['balance'])

            if (b > a):
                print("Congratulation!! Your Transaction is successful")
                print("Your balance after transaction is: ", b)
                break


        else:
            print("Transection is not completed.")
            print("Your balance after transaction is: ", b)
            break
    else:
        print("Access Denied")
        print("Please try agin with your NFC card")
```

```
C:\Users\SHOVON\AppData\Local\Microsoft\WindowsApps\python3.9.exe "F:/Thesis/Codes/Website V3-Done/NFC/NFC.py"
Please put your NFC card!!
Congratulations Card Accepted. Please put your info in firefly and CTRL+V to paste in Send option
Your Current balance is:  18000000
Please wait!, Transaction is in process....
Congratulation!! Your Transaction is successful
Your balance after transaction is:  19000000
```

Figure 4.11: Payment Completed

4.2 Comparison and Result Analysis

4.2.1 Comparison analysis for Mana:

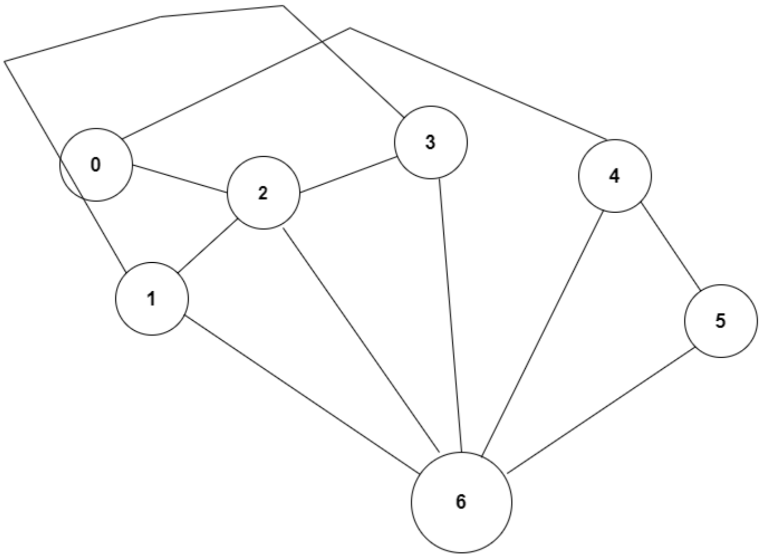


Figure 4.12: Payment Completed

Suppose we are considering a situation where node 6, which is the current tip node got neighbor's. And they are node 1, node 2, node 3, node 4 and node 5. Now if we utilize q-learning with mana as reward then node 6 will choose its neighbor's efficiently with mana >= average mana of neighbor's. Node 6 needs to request it's neighbor's till it got the sufficient mana. Sufficient mana depends on the network's throughput.

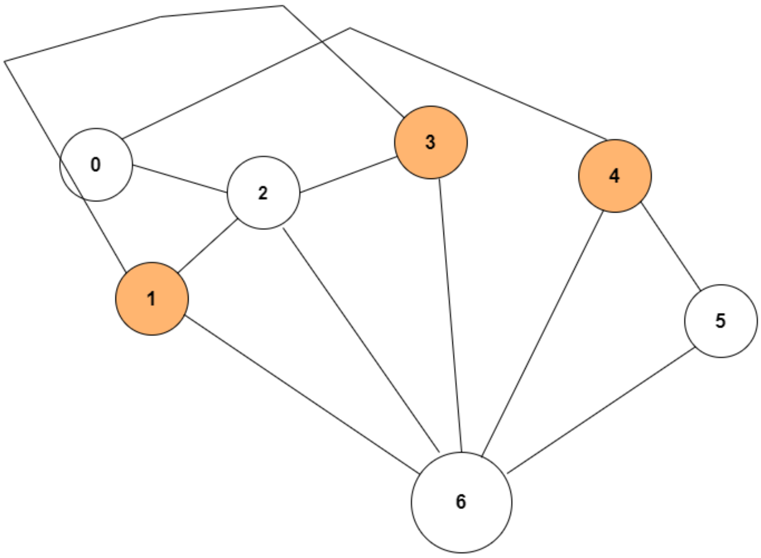


Figure 4.13: Payment Completed

Here, node 6 is efficiently choosing neighbor's using mana as reward. And because of the q-learning algorithm, node 6 or agent will eventually learn how to choose it's neighbor's efficiently over time.

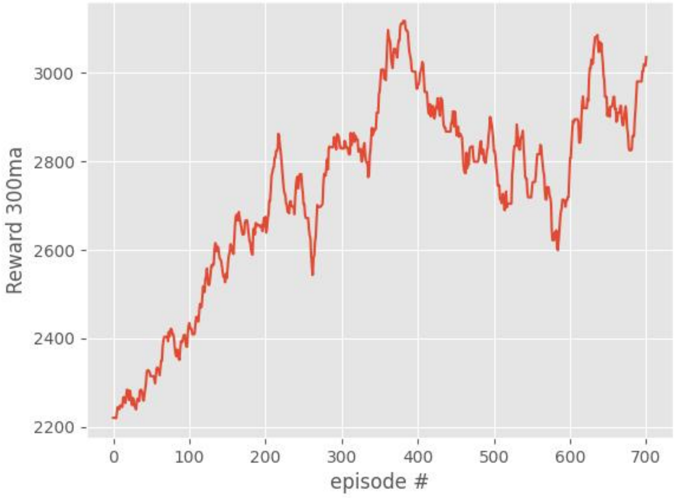


Figure 4.14: Payment Completed

At this position, agents reward will increase. Figure 4.14 shows that the graph eventually increases till episode 400 (approximately). Till then agent is searching within it's neighbor's. But as neighbor's are also providing mana to complete it's transaction, at a time agent's reward will also decrease because the neighbor's are not able to provide enough mana. At that time, agent will start searching for mana within it's neighbor's neighbor. For example, next efficiently chosen neighbor's neighbor is node 1. Then same q-learning process will run for node 1 and it will continue it's searching and acquiring for mana till it gets sufficient amount. And again, from the figure 4.14 is showing us that again reward per episode or mana of the agent or node 6 is increasing. This sufficient mana is around 70% of the whole network.

Now we run the algorithm without reward or mana.

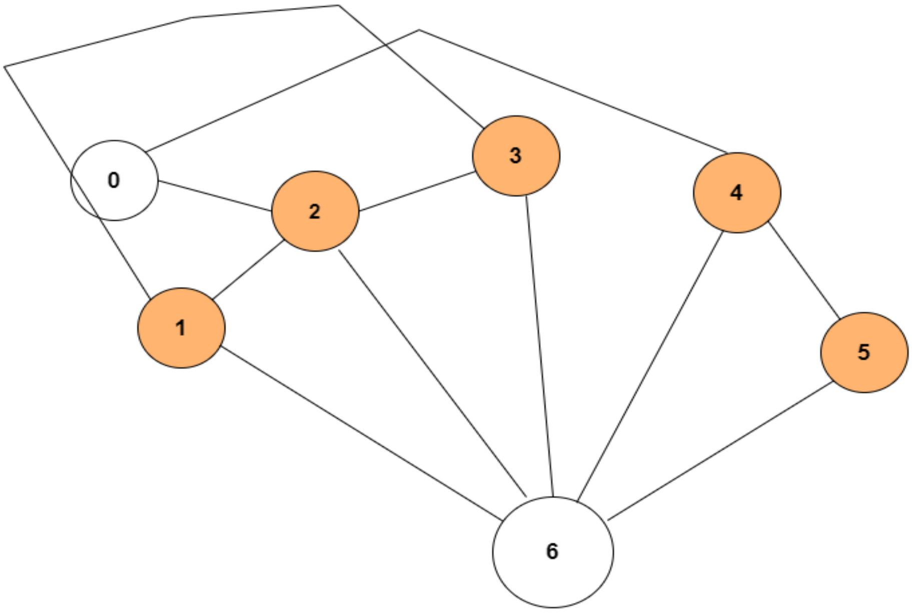


Figure 4.15: Episode vs Reward without q-learning

In this case we found that, agent is linearly searching for all of it's neighbor or went to all of it's neighbor's for mana. So, at first agent will receive high mana but it is asking to all of it's neighbor's. Which is the problem called message flooding. It is flooding the message to all of it's neighbor's. It will take huge amount of time to collect mana and from figure 4.16 we can see that eventually, mana dropped because now agent is not able to even ask from neighbor's neighbor for mana. And we get a downward graph.

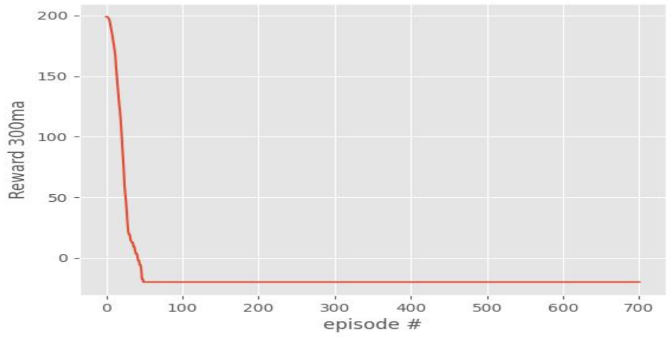


Figure 4.16: Episode vs Reward without q-learning

After running the code several times, there is a plain graph with no reward or mana.

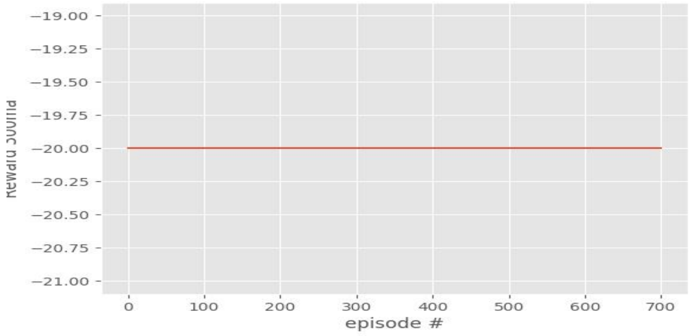


Figure 4.17: No reward for not using q-learning

So from the above comparison we came to the conclusion that, q-learning resolves the message flooding problem and efficiently choose neighbor's based on mana.

4.2.2 Comparison analysis for Number of nodes visited:

In case of number of nodes visited, we had to check in which algorithm there is more number of nodes visited. Whether it's with mana or withour mana. First of all, considering the algorithm using mana. Algorithm with mana, works better because it had to visit less number of nodes to acquire needed mana or approval wright. The following figure shows the output for running the entire code for multiple times.
Figure 4.18 shows that we got at most 1050 nodes or we our algorithm had to visit at most 1050 nodes to gain approval weight.
But before mana, new transaction had to visit around 5000 nodes to gain sufficient approval weight.

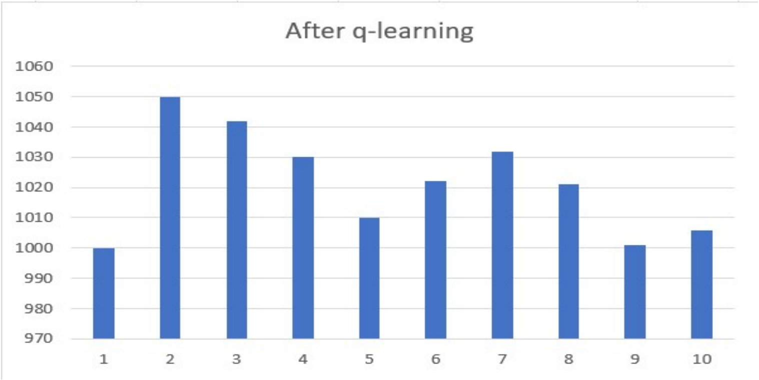


Figure 4.18: Number of nodes visited with sufficiently choosing neighbor's

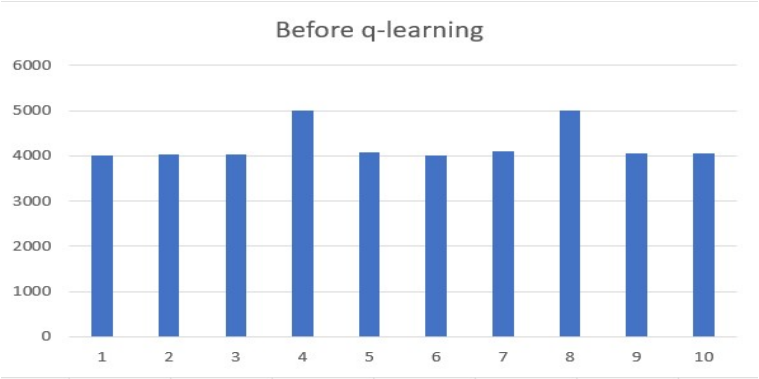


Figure 4.19: Number of nodes visited without sufficiently choosing neighbor's

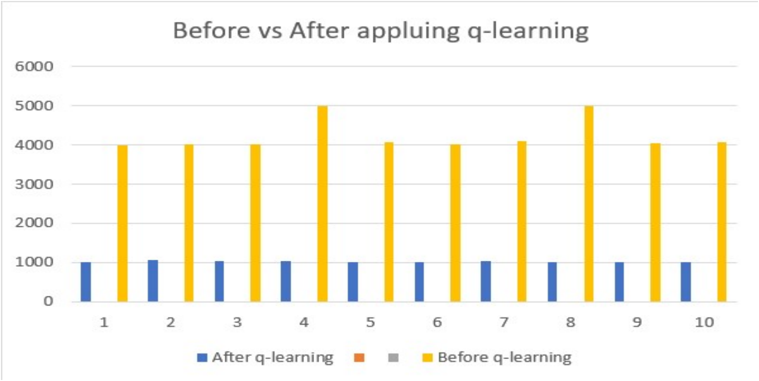


Figure 4.20: Number of nodes visited with mana vs without mana

And figure 4.20 shows that algorithm with mana works better because using this algorithm new transaction had to visit less amount of nodes to gain sufficient mana otherthan algorithm without mana.

Chapter 5

Conclusion

The intrinsic features of blockchain technology, especially the requirement for ever-increasing storage and limited scalability, continue to limit its usage for IoT-based collaborative platforms. In this research, we look at the criteria for interoperability between blockchain and tangle distributed systems. For safe cashless micro payments, we provide a scalable and cost-effective advanced routing system based on bidirectional micropayment transactions[7]. The combination of blockchain and tangle improves IoT functionality and storage while maintaining a high degree of reliability, data accessibility, integrity, and security. In the backend, a Blockchain-based platform is implemented, which is largely utilized for data storage and smart contracts. The applications operate on a Tangle-based framework in the frontend to make them compatible with IoT devices. We show how the existing system would benefit from the protocol's adoption, as well as how the protocol's adoption might benefit systems. A good result on transaction addressing method will significantly improve the secure routing protocol's performance and efficiency for micropayment networks[7].

Bibliography

[1] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[2] R. Sutton and A. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998. DOI: 10.1109/TNN.1998.712192.

[3] X. Bu, J. Rao, and C.-Z. Xu, "Coordinated self-configuration of virtual machines and appliances using a model-free learning approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 681–690, 2013. DOI: 10.1109/TPDS.2012.174.

[4] S. Popov, "The tangle. white paper," *Available at: Iota. org*, 2016.

[5] L. Ankney, "Iota: No transaction fees and infinite scalability — how does it work? | by leslie ankney | medium," Sep. 2017.

[6] Y. Jiang, C. Wang, Y. Huang, S. Long, and Y. Huo, "A cross-chain solution to integration of iot tangle for data access management," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1035–1041. DOI: 10.1109/Cybermatics_2018.2018.00192.

[7] Q. Xia, E. B. Sifah, K. Huang, R. Chen, X. Du, and J. Gao, "Secure payment routing protocol for economic systems based on blockchain," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 177–181. DOI: 10.1109/ICCNC.2018.8390309.

[8] S. Popov, *Iota: Feeless and free*, 2019. [Online]. Available: https://blockchain.ieee.org/technicalbriefs/january-2019/iota-feeless-and-free.

[9] YouTube, Aug. 2021. [Online]. Available: https://www.youtube.com/watch?v=OCOag0CKXvY.

[10] I. Foundation. "Congestion control - iota research symposium 2021." (2021), [Online]. Available: https://www.youtube.com/watch?v=OCOag0CKXvY&t=2557s.

[11] ——, *Explaining mana in iota*, Mar. 2021. [Online]. Available: https://blog.iota.org/explaining-mana-in-iota-6f636690b916/.

[12] W. Sanders, *Improvements to the iota 2.0 consensus mechanism*, Oct. 2021. [Online]. Available: https://blog.iota.org/improvements-to-the-iota-2-0-consensus-mechanism/.

[13] Nagad. “Profit.” (), [Online]. Available: https://nagad.com.bd/en/service/profit-2/.

[14] *Policies and value functions - good actions for a reinforcement learning agent.* [Online]. Available: https://deeplizard.com/learn/video/eMxOGwbdqKY.