IOTA for Internet of Things

by

Payal Kothari

A Project Report Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. R. K. Raj

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

May 2018
Abstract

IOTA for Internet of Things

Payal Kothari

Supervising Professor: Dr. R. K. Raj

IOTA is an open source project started in 2015. It is a cryptocurrency which uses Tangle instead of the traditional Blockchain technique. Tangle and Blockchain differ in their structures. Tangle overcomes some of the limitations of Blockchain which makes IOTA more suitable for trading amongst Internet of Things. The total supply of IOTA is fixed, we cannot mine IOTAs. As mining is not involved, there is no transaction fee. This makes micro-payments possible in Machine to Machine (M2M) economy. IOTA does not need a massive amount of processing power, so it can be used by small IoT products.

This paper provides an introduction to IOTA, detailed explanation of how IOTA uses Tangle, related work done by third parties to support IOTA project, a hypothesis on domination of IOTA in IoT market, creation of a full node in IOTA main network, demonstration of IOTA’s Python library.
# Contents

Abstract .................................................................................................................. ii

1 Introduction ................................................................................................. 1
   1.1 Introduction .................................................................................. 1
   1.2 Background ............................................................................... 2
   1.3 Related Work ......................................................................... 4
   1.4 Hypothesis ............................................................................. 5
   1.5 Roadmap .................................................................................. 5

2 Design and Implementation ................................................................. 6
   2.1 Create a Full Node .................................................................. 6
   2.2 Connect IOTA wallet to Full Node ............................................ 7
   2.3 Demonstrate usage of IOTA Python library to communicate between two
       nodes .................................................................................. 9

3 Analysis ...................................................................................................... 14

4 Conclusions ............................................................................................. 15
   4.1 Current Status ....................................................................... 15
   4.2 Future Work .......................................................................... 15
   4.3 Lessons Learned ................................................................... 16

Bibliography .................................................................................................. 17

A Code Listing ............................................................................................... 20
List of Tables

1.1 IOTA Unit System [16] . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
List of Figures

1.1 Tangle ................................................................. 3

2.1 iric Configuration Tool ........................................... 8
2.2 IOTA Peer Manager .................................................. 9
2.3 Grafana: Full Node Data ........................................... 10
2.4 Grafana: Market Data .............................................. 11
2.5 Grafana: Neighbor Nodes Data ................................. 11
2.6 Light Wallet/ Full Wallet ......................................... 12
2.7 IOTA Wallet ........................................................ 12
2.8 Wallet Transaction History ....................................... 13
2.9 Node Information using API call ............................. 13
Chapter 1

Introduction

1.1 Introduction

Internet of Things is a network of physical devices which talk to each other. These devices have different types of computing systems in them. The computing systems include sensors, actuators, softwares etc. IoT devices collect data using sensors and provide useful results or transfer the collected data to other IoT devices. These devices can be used in various fields such as home automation, energy management, environment monitoring [3]. A simple example of using IoT in home automation is, refrigerators can inform the users on their phones that they are running out of milk.

IoT is growing rapidly. It makes things a lot easier in our day to day life, but there are several problems that we still have to solve. Security is one of the problems with IoT [13]. Hackers can easily access the IoT devices installed in our house such as a webcam. Privacy is another important issue [13]. IoT devices collect data, the vendors of these IoT devices can know when the user leaves his house or what time he sleeps etc. Connectivity amongst devices is another challenge for IoT ecosystem. IoT devices are increasing exponentially. Experts have predicted that by 2020, there will be about 30 billion IoT devices [3]. This shows that we cannot have centralized IoT system. It has to be a decentralized network to avoid problems in connectivity [1]. IoT network needs a way for easy Machine to Machine (M2M) interaction.

We can solve some of the above mentioned issues using Blockchain technology. But,
Blockchain has some limitations such as scalability, transaction fee, hardware requirements. It is not highly scalable [15]. It involves mining, so there is a fee for every transaction. For mining it requires hardwares which use massive computing power. IOTA solves these three limitations of blockchain and satisfies requirements of IoT network.

IOTA uses a distributed ledger same as Bitcoin. The network is highly scalable as the approval process is parallelized. The total supply of IOTA is fixed which is 2779530283277761. There is no concept of mining, so there is no transaction fee. IOTA doesn’t use a massive amount of computation power to approve a transaction. IOTA also provides decentralized peer to peer (P2P) solution. These properties of IOTA are good for Machine to Machine (M2M) transactions.

1.2 Background

The core technology of IOTA: Tangle is a Directed Acyclic Graph (DAG) as shown in Figure 1.1. Each block in the Tangle graph is a transaction. New blocks are added in the graph from the right side. Block 0 in Figure 1.1 is called as a genesis block. This is the first created block in the Tangle network. All IOTAs were created by a genesis block [12]. The total supply of IOTAs will never change and no more IOTA will be created in the future. The tokens created by genesis block were distributed to IOTA founders. Founders sold some tokens to other people and this is how trading in the Tangle network started.

There are no specific nodes that are responsible to approve transactions. All nodes are involved in the approval process. Each new transaction selects past two transactions and it validates these selected transactions. In Figure 1.1, block 6 is a new transaction which wants to join the network. It selects block 4 and block 5, it will validate and approve the selected transactions. After the approval process, block 6 can broadcast its own transaction in the network. Now, some future transaction will approve block 6. This process of selection of past transactions is called as tip selection.

When a node wants to make a transaction, it follows following three steps [15]:
1. Signing: The node which is making a transaction, signs the transaction with its private key.

2. Tip selection: New transaction selects past two transactions. These selected transactions are called as branchTransaction and trunkTransaction. MCMC (Markov Chain Monte Carlo) algorithm is used to randomly select two transactions from available tips.

3. Proof of work (PoW): In addition to approving the selected past transactions, every node has to complete a proof of work to broadcast its own transaction in the network.

Once the above three steps are completed, the new transaction will be a part of the Tangle network. Now, some future transaction will select and validate this block. This is how peers approve each others’ transactions, so no one has to pay any transaction fee. So, micropayment and nanopayments are possible. As multiple transactions can happen at the same time and multiple tips can be approved at the same time. This approval process is parallel. This makes Tangle network grow very fast [15].

IOTA is in testing phase at this time. The team has created a GUI wallet, where we can create our account and make transactions. Table 1.1 shows IOTA unit system. To login to
Table 1.1: IOTA Unit System [16]

|                | 10^15 | 10^12 | 10^9  | 10^6  | 10^3  | 1
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peta IOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tera IOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Giga IOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mega IOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kilo IOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the wallet user needs 81 characters long seed/password. This seed is unique for every user. Client libraries are also available for developers in different programming languages such Java, Python, JavaScript, C#, Go [15]. All these are available on Github, so anyone can contribute to improving the libraries. Mainnet and testnet are created to simulate a network of blocks where nodes make transactions. The IOTA team is stress testing the network at this time. A coordinator is being used to confirm the approved transactions in the mainnet [24]. This is for testing purpose only. The coordinator will be removed once the network becomes sufficiently large.

1.3 Related Work

IOTA is a large community project. Many third parties have created tools to make things easier to test. If anyone wants to add a new transaction in the mainnet then that user has to ask for IP address and port number of other two nodes which are already present in the network. Then the user has to manually add these IP addresses and port number to make them peers.

There are two tools available developed by the third party to automatically find peers and add them to user’s node. One tool is called Nelson [14] is developed to work with Ubuntu and CentOS operating systems. Another tool called Bolero [17] is developed to work with Windows and Mac. Peer manager GUI [9] is available which shows data peers of specific full node. A tool named IOTA ctps [4] is available which takes data from IOTA
full node and calculates useful metrics. People who are interested in cryptocurrency and
IoT contribute to the available libraries.

1.4 Hypothesis

As explained in Section 1.1, blockchain technology cannot satisfy all the requirements of
IoT network. The hypothesis for this project is IOTA can satisfy IoT requirements such as
scalability, decentralization, fee-less transactions. These properties of IOTA are explained
using different implementations in this project.

1.5 Roadmap

- Section 2 explains design and implementation information.
- Section 3 provides analysis of the results found in Section 2.
- Section 4 presents current status of the project, lessons learned while working on this
  project and possible future work.
- Appendix A provides code snippets which are explained in Section 2.
Chapter 2

Design and Implementation

2.1 Create a Full Node

IOTA foundation has developed IOTA reference implementation (IRI) [5]. Using IRI, we can create our own IOTA full-node and this full-node will be a part of the IOTA mainnet. This full-node handles approval of tips, proof of work etc. When we create a full-node, we have to find peers and add them manually to sync our node with the IOTA main network.

Following third party tools were used while creating a full-node for this project:

1. IRI playbook [19]
2. Nelson [14]
3. IOTA Peer Manager [9]
4. Data Exporter [10]
6. Prometheus [25]

IRI playbook helps us create a full-node (using IRI) [19]. This playbook installs other third party tools mentioned above. While using IRI playbook, a manual addition of peers is not required.

Full node was created using a VPS (Virtual Private Server). This VPS was created using Google Cloud Platform. The specifications of VPS were set as follows [21]:
• Operating System: Ubuntu 16.04 (amd64) Server Cloud Image (Xenial)

• 2 CPUs, 4 GB RAM

• 30 GB SSD persistent disk

Script given in the IRI playbook was used to install and run the IRI-playbook [20]. This script handles installation of system packages and git according to the servers system specifications. After running the script, the IOTA full-node was up and running in the IOTA main network. Nelson tool installed by the playbook finds peers automatically. ”iric” is a configuration tool provided by the playbook which provides various options such as enable/disable Nelson, get info of your own node, get your neighbors’ info etc. Figure 2.1 shows all the options.

IOTA peer manager provides a GUI to view details of peers, the user can even manage the peer from this GUI. Peer manager can be access via: http://node’s_external_ip:8811 [22]. Figure 2.2 shows the peer manager of the IOTA node created for this project. It shows a graph for each peer, we can add new peers manually using IP address and port or we can remove the existing peers.

IOTA data exporter exports data related to the full node and shows graphs on Grafana as shown in Figure 2.3. It also uses Bitfinex API [2] and shows graphs for market data [10] as shown in Figure 2.4. Last data source for Grafana is the metrics calculated by iota ctps [4] as shown in Figure 2.5.

2.2 Connect IOTA wallet to Full Node

IOTA wallet is a GUI designed to easily send and receive IOTAs. IOTA wallet can be installed via: https://github.com/iotaledger/wallet/releases. This GUI provides two options as a light wallet or a full wallet as shown in Figure 2.6. Full wallet option stores the whole database of Tangle on the user’s local machine. To avoid downloading a huge database of Tangle, light wallet option can be selected. Light wallet can be
connected to one of the full nodes provided by the IOTA Foundation [23] or our own full node using IP address and port number of the node as shown in 2.7a. For this project, the IOTA light wallet was connected to the full node created in Section 2.1.

After connection of wallet to an appropriate full node, there are options to send and receive IOTAs. The receive section as shown in Figure 2.7b provides an address, this address can be used to receive IOTAs. Similarly, IOTAs can be sent to other node’s address. Figure 2.8a shows the history of previous transactions.
2.3 Demonstrate usage of IOTA Python library to communicate between two nodes

PyOTA is a Python library which implements official API provided by IOTA Foundation. The IOTA API can be accessed via: https://iota.readme.io/v1.2.0/reference. PyOTA’s git repository was cloned and installed locally for this project. The git repository can be accessed via: https://github.com/iotaledger/iota.lib.py. Two wallets were created for this project. To log into a wallet, we need 81 character long string called seed. This is the only string which is used to access the wallet so it must be saved securely. For this project, assume wallet_1 is associated with seed_1 and wallet_2 is associated with seed_2.

An IOTA object was created using the URL of VPS where full node created in Section 2.1 is running. The wallets were connected to this full node as explained in Section 2.2. Listing A.2 shows creation of an IOTA object [8].
Now, this created IOTA object can be used to call other functions on the IOTA node. For example, “get_node_info()” [8] call shows node’s information as shown in Figure 2.9. Any node’s account balance can be checked using ”get_inputs()” API call [6] as shown in Listing A.3. This call will print the account balance of wallet_1 as 3006, that we can see in Figure 2.7b. Listing A.1 shows how to transfer one IOTA to other node [7]. The address in the ProposedTransaction() block is the address of wallet_2. The tag for this transaction is given as ”BYPAYAL”. Figure 2.8b shows that wallet_2 received 1 IOTA with the same tag.
Figure 2.4: Grafana: Market Data

Figure 2.5: Grafana: Neighbor Nodes Data
Figure 2.6: Light Wallet/ Full Wallet

Choose Wallet Type:

Would you like to run a light node or a full node?

Light Node

A light node connects to a remote node, either owned by you or someone else.

Full Node

A full node will download and sync the tangle to your disk and requires java to be installed.

(a) Connection of light wallet to Full Node

(b) IOTA wallet transaction options

Figure 2.7: IOTA Wallet
Figure 2.8: Wallet Transaction History

(a) Wallet 1 transaction history
(b) Wallet 2 transaction history

Figure 2.9: Node Information using API call
Chapter 3

Analysis

Scalability, decentralization and fee-less transactions in IOTA network were tested in this project. Figure 2.3 shows that total transactions which includes approved as well as un-approved transactions is very high. Even if the total number of transactions is high, the average confirmation rate is 95.4%. This shows that 95.4% of the unconfirmed transactions are getting confirmed per second. The percentage of avg. confirmation rate proves that the scalability of the IOTA network is very high. Scalability is achieved in the IOTA network by making the approval process parallel i.e multiple transactions can join the network at a time and multiple transactions get approved at a time.

In the IOTA network, there are no specific nodes which are responsible to confirm the pending transactions. New transactions validate the pending transactions, this provides the decentralization.

As shown in Section 2.3, we can use the IOTA libraries to make transactions between any two IoT devices. The usage of libraries is very simple and easy to understand. Figure 2.8 shows a transaction between Wallet_1 and Wallet_2. One IOTA was sent from Wallet_1 to Wallet_2, the complete amount was received by Wallet_2, no transaction fee was deducted. This shows fee-less feature of IOTA technology.
Chapter 4

Conclusions

4.1 Current Status

I have created a virtual private server on Google Cloud Platform. I have installed the provided IOTA Reference Implementation (IRI) on my VPS to create an IOTA full node. My full node is running in the IOTA main network. I used IRI Playbook [19] for this installation. This playbook provides access to Grafana [11] which monitors live data. My Grafana account shows graphs for my node, the market data, and the metrics provided by IOTA ctps [4] as shown in Figure 2.3, Figure 2.4 and Figure 2.5. My peer manager account shows live data of my full node’s peers as shown in Figure 2.2.

I have installed an IOTA wallet on my laptop and connected my wallet to the full node created in Section 2.1. I have created two accounts on this wallet. My code uses IOTA Python library and demonstrates communication between these two accounts.

4.2 Future Work

My current work demonstrates use of few IOTA Python library’s calls, in the future remaining calls can be tested. IOTA libraries are also available in other programming languages such as Go, Javascript etc. These libraries can also be used to communicate between IOTA nodes. There are many third party projects available to support IOTA such as Nelson [14], [17], data exporter [10], CarrIOTA Field [18]. As this is a community project any contribution to existing third party or IOTA foundation projects is valuable.
4.3 Lessons Learned

I learned the difference between Tangle and Blockchain technology. I read whitepaper for Tangle which provides a detailed explanation of Tangle. I studied the requirements of IoT (Internet of Things) devices. I learned how IOTA provides advantages over other cryptocurrencies and why these advantages are good for IoT devices. This project helped me learn available third party projects which are supporting the IOTA project. I worked with services such as Google Cloud Platform and Grafana.
Bibliography


Appendix A

Code Listing

Listing A.1: Send one IOTA from wallet 1 to wallet 2

```python
transfer = IOTA_object.send_transfer(
    depth = 3,
    transfers=[
        ProposedTransaction(
            address = Address(wallet_2_address),
            value = 1,
            tag = Tag(b'BYPAYAL'),
        ),
    ],
    min_weight_magnitude = 14,
)
```

Listing A.2: Creation of an IOTA object

```python
url = "http://external_ip_of_VPS:14267"
seed = "seed_1"

IOTA_object = Iota(url, seed)
```

Listing A.3: Sender’s account balance

```python
sender_data = IOTA_object.get_inputs() # returns a python dictionary
print("* Sender’s balance : " + str(sender_data["totalBalance"]))
```