nomadic labs

Nodes in the Tezos blockchain

Table of contents

Introduction

Introduction

The Tezos blockchain distinguishes itself from other blockchains through a set of specific characteristics:

1. The on-chain governance mechanism, which allows for protocol upgrade through a series of amendments[1]
2. The LPOS consensus algorithm (Liquid Proof Of Stake)
3. Michelson, Tezos' smart contract language, which allows formal verification implementation

Blockchain is a peer-to-peer network in which every peer is a node ensuring a set of diverse functions within the distributed software. In this document, we will explore the notion of nodes in Tezos, focusing on their role as well as their limitations.

[1] Document for the amendments

Definition

Definition

Within a blockchain, nodes are vital elements in order to keep the network decentralized.
A node is a physical machine which stocks a copy of the current state of the blockchain and updates it with all the information it receives from other nodes. Therefore a node is connected in a peer-to-peer way.

A node systematically communicates the information it receives to all the other nodes to which it is connected in order to, later on, allow them to transmit this information to their own network of nodes. This ensures that every node from the Tezos network possesses the same information.

The peculiarity of public blockchains lies in the fact that anyone can launch a node. The chain is therefore expected to have a large number of nodes, thus reducing the risks associated with the potential failure of a critical actor of the network.
Once launched, a node connects to other peers (an average of forty) in the network, establishing a communication channel and thus allowing the interested parties to exchange information.

By mid-2020, the Tezos blockchain had more than 1,000 active nodes on the network. [1]

[1] Inspecting Tezos decentralization, analysis by Baking Bad

Peer-to-peer network

Peer-to-peer network

The Tezos network implements a gossip protocol whose main operations are linked to:

- **A read / write scheduling mechanism** allowing to control the bandwidth
- **An encrypted communication service**
- **Black, gray and white listing services** allowing to establish or refuse connections requested by certain peers

Differences between a node and a baker

Differences between a node and a baker

On the Tezos blockchain, the bakers are entities taking up the role of block creators (they are the equivalent of miners on Bitcoin). A node is therefore not necessarily attached to a baker. However, a baker must necessarily be linked to a node in order to monitor the arrival of new blocks, to retrieve pending transactions and to distribute its own blocks if and when selected to add a block to Tezos blockchain.

Many bakers own their own node, however the concepts of 'bakers' and 'nodes' are distinct. Thus, while a baker will always need to interact with at least one node, a node, on the other side, can be baker-independent.

The different types of nodes

The different types of nodes

When launching a Tezos node, several configurations allow it to synchronize with the other nodes of the blockchain network. Each node type leads to a synchronization, which speed will vary and which will allow different functions to be performed. Below are the 3 main modes of node configurations:

- **Archive mode**: a node storing the entire blockchain state. A node launched on the Tezos blockchain in the archive mode will download the entire blockchain[1] from the launch of the network. Early 2021, the entire blockchain will take about $145GB$.
- **Full mode** (default mode): a node storing the entire blockchain, without storing the 'context' (the blockchain state containing information such as : `` Alice's account[2] owns $5\frac{1}{3}$'') below a certain level called **savepoint**. It is possible to rebuild an Archive Node from a full snapshot.
- **Rolling Mode**: the lightest existing node mode, which stores a fragment of the chain removing anything before that fragment (blocks, operations, archived contexts).

It is possible to use a *snapshot* (a representation of a copy of the blockchain's state at a given point in time) to allow a node to synchronize quicker. Snapshots can be used to initialize full and rolling nodes. It is also possible to switch between the different types of nodes.[3].

[1] Archive nodes keep the contexts and "checkout tickets" of all blocks
[2] In reality, Alice has a pseudonym represented by her public key hash.
[3] See the different modes' types

Node configuration

Node configuration

The different types of node previously introduced allow users to choose the most suitable one depending on the amount of information needed. An explorer [1], for example, will tend to need an archive node, while a user wanting to experiment with basic node deployment may simply use a rolling version.

Launching a node in one of the different modes:

- `$ tezos-node run --history-mode (archive, full, experimental-rolling)`

Importing a full / rolling snapshot[2]:

- `$ tezos-node import snapshot path/to/snapshot`

Obtaining snapshots:

- https://snapshots.tulip.tools/#/
- https://snapshots-tezos.giganode.io/
- https://xtz-shots.io/

[1] Tool allowing to monitor blocks creations, transactions,...

[2] A snapshot with the corresponding extension (*.roll* or *.full*) is needed.

Possible actions with a node

Possible actions with a node

In a blockchain, a node acts as an interface between the user and the peer-to-peer network. Two main uses exist:

- **Information storage**: since a node keeps a real-time copy of the current state of the blockchain, it allows learning about what is happening in real time. This use, for example, allows to feed indexers, which retrieve information from the node and process it to make it more humanly-readable.
- **Information transmissionn**: user-initiated transactions go through the nodes before becoming part of the blockchain. Nodes are thus an entry point for transactions intended to be validated by the bakers.

Generally, these uses are complementary. For example, a user might often need to check the contents of the chain to verify that what he is sending is not immediately rejected.
A baker will use both of the aforementioned aspects to operate. Indeed, it will have to read the pending transactions' validation, then read the context[1] in order to ensure that the transactions are applicable before validating them by including them in a block that will be transmitted to the rest of the network.

[1] Blockchain's current state

Communicating with a node

Communicating with a node

As seen in the previous slide, communicating with a node allows access to the peer-to-peer network. Bakers automatically communicate with the node made available to them, but it can be useful for a user to be able to interact directly with his node when wanting to inquire about the chain's status or when wanting to push transactions himself.

Direct communication with a node happens through the use of Remote Processing Call (RPC)[1].

It is also possible, if needed, to use a dedicated user interface such as a client[2], indexer[3] or wallet[4].

However, in practice, indexers are the most practical tool for a human to obtain on-chain information.

[1] RPC calls guide
[2] Tezos client guide
[3] Tezos indexers
[4] Tezos wallets

Advantages offered by a node

Advantages offered by a node

For operational motives, an entity might wish to have control of its own Tezos node. Such control helps avoiding dependence on a third party node that could be failing. Firstly, this means independence from a blockchain node that could be prone to failure and in addition helps increase chain security by contributing to the peer-to-peer network.

Owning a node means that one can manage his own transactions and ensure that these are broadcasted to the network.

Running a node makes it possible to contribute to the Tezos network and thus increases its security and decentralization. The higher the nodes quantity, the more decentralized and robust a blockchain will tend to be.

Becoming a baker on the Tezos blockchain usually involves controlling a node and also provides benefits. Indeed, any baker in the network receives rewards in return for his work in validating the blocks and securing the chain. In particular, a baker can include its own transactions without paying any fees. Added to this is the obtention of voting rights relative to the amendments[1] proposed by the chain participants.

[1] Amendment document

Possible node attacks

Possible node attacks (1/2)

A blockchain node can suffer from different types of attacks, inherent to all software based on a peer-to-peer network. It is usually recommended to any node to secure its communication channels and to store its private keys (if it has any) in a cold wallet [1].

Denial of service:

The denial of service attack (or DoS) is a computer attack that makes a service unavailable. In the case of the Tezos node, this would mean attacking a node. Certain requests, that are sometimes very resource intensive (the calculation of baking rights up to arbitrarily high levels for example), can be sent to a node with the goal of making the given node unavailable from a lack of resources necessary to respond to said request.

It must therefore make sure to limit access to these requests. As this is a very classic attack on the Internet, the usual preventions apply.[2].

[1] Key storage technic which uses a hardware material not connected to the internet (examples : Ledger and Trezor products).

[2] https://en.wikipedia.org/wiki/Denial-of-service_attack

Possible node attacks (2/2)

Sybil attack:
This is an attack that consists in creating a large number of identities on the peer-to-peer network with the aim of isolating one or more nodes from the rest of the network and giving them false information. In Tezos, this attack consists in generating a large quantity of nodes, in order to isolate one or more targeted nodes to which the attacker would communicate compromised information (erroneous blocks for example).

To prevent this kind of attack, one of the features of the Tezos blockchain allows any node to change peers regularly and automatically, in order to make sure that the attacker cannot maintain a long enough grip for the chain to be affected.

In addition, creating a node identity costs a certain amount of computing time[1]. Therefore, generating a node takes a few seconds, but generating enough nodes to split the network would take a long time, even with a powerful machine.

Finally, if attacks are observed on the network, it would be possible to increase the level of work required by peers and regenerate all identities to prevent the attack, and to contain it completely if all peers do so.

[1] To create an address, it must be given a certain level of "work" ranging from 0 to 255. This level corresponds to the number of zeros in front of the hash of its identity. When creating an identity, the program generates different versions until it finds one with a sufficient level. One can then choose to refuse to connect to peers whose identity is lower than a certain value. By default this value is set to 26.

Node access services

Node access services

Some nodes, such as SmartPy, are publicly accessible:

- `https://mainnet.smartpy.io`

There also exist some other node services, including:

- `https://tezoslink.io/`
- `https://tezos.giganode.io/`

nomadic labs

Let's keep talking

https://tezos.com
https://developers.tezos.com