Decentralized Domain Registry Using Blockchain

Kavita Poojary[1*], Shefali Sawardekar[2], Bhavna Arora[3]

Dept. of Computer Engineering, Atharva College of Engineering, Mumbai – 400095, India

Publication Info

Article history:

Received : 16 February 2020

Accepted : 21 May 2020

Keywords:

Blockchain, Distributed, Distributed File System, DNS, IPFS, InterPlanetary File System

Corresponding author:

Kavita Poojaryz

e-mail: *kavita16543@gmail.com*

Abstract

We present Decentralized Domain Registry (DDR), a framework that replaces current top-level DNS framework and authentication specialists, which will offer adaptable, secure and strong DNS framework DDR utilize an area name possession scheme base on blockchain. DDR evacuates existing DNS quality, for example, DDOS assaults, DNS ridiculing and restriction via government. DDR gives decentralized validated record area name possession which will wipe out the requirement for endorsement specialists. DDR is turn around good with DNS. The system will reduce latency using Interplanetary File System (IPFS) through end to end content delivery.

1. INTRODUCTION

We are all now connected by the internet, like neurons in a giant brain. As of June 2018, the internet users worldwide were 4.2 Billion among 7.6 Billion, which is almost 55% of the total population, which itself is a big number. Roughly in one moment, there will be tens of thousands of new Facebook posts, thousands of new tweets and applications downloaded. Since the internet is generally accessible, only one moment of worldwide online action is jam-stuffed, brimming with occasions, from correspondence with others to information stockpiling to diversion alternatives in abundance. The proportion of data moved to the internet in a singular second is staggering.

All the internet has relied upon IP addresses. When worldwide start to finish availability was imagined for interchanges with all internet has, IP addresses be comprehensively novel. Computers can only recognize numbers. Moreover, the human brain can recall words efficiently rather than numbers. This is where the Domain Name System comes into picture. The Domain Name System, normally referred to as DNS, is a critical part of the internet. The Domain Name System (DNS) is the phonebook of the internet. People access the data online through domain names, as spotify.com or edx.com. Internet browsers between act through Internet Protocol (IP) addresses. DNS interprets space names to IP addresses so programs can stack Internet assets.[1]

Recently there have been leaks concerning the classified information explaining NSA's spying capabilities which has raised questions regarding the security of SSL and TLS and the certificate authorities. These kinds of dangers to DNS, alongside security concerns, were not viewed as structuring the convention, yet DNS is excessively generally utilized and excessively coordinated with the internet to be supplanted.[2] There have been numerous investigations and endeavors to propose a DNS framework dependent on a dispersed hash table (DHT). We stretch out on those papers by executing a Blockchain supported Domain Name framework (DNS), limiting inactivity separation instead of jump separation and actualizing a common record of possession.[3]

A blockchain is a distributed, decentralized and circulated record used to exchange records across numerous PCs so the record can't be altered without modifying all the consequent square and consensus of the system. Every block includes the cryptographic hash of the previous record in the blockchain, connecting the two. The linked blocks structure a chain.

2. BACKGROUND AND RELATED WORK

2.1. A. Using blockchain to validate DNS Records

DDR makes use of operation proof mechanism of Bitcoin to evidence possession of domain name. Miners are rewarded

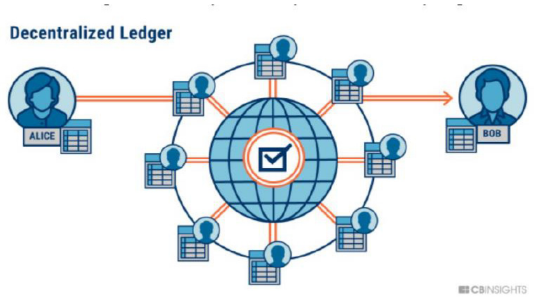


Fig. 1: The above figure shows a Decentralized Ledger/Blockchain

with right to claim a domain name instead of rewarding them with cryptocurrency.[4] Each block contains transactions (Fig. 1) that indicate miner who claim for a fresh domain name or transfer of domain ownership. A case for new space is approved by a reference to an unclaimed mining reward possessed by the asserting client. Each transfer is validated by reference to a previous claim record or transfer record and needs to indicate his proprietorship by the moving party. Each space name in the framework can be related right now to the new proprietor's open key. New spaces can be proclaimed, and long-standing areas can be moved among proprietors.[5]

2.2. B. Distributed Decentralized Domain Name Service

It proposes a structure that takes into account the least inactivity streamlining. This framework points just to supplant legitimate Top-Level Domain servers at present oversaw by enlistment centers, where most records are essentially a forward to a definitive DNS server oversaw by the area proprietor, as opposed to supplanting all degrees of DNS. This constraining of degree permits us to keep on exploiting DNS expansions and as spots duty of dealing with the system with the individuals who have an impetus for its kept working. It has consistently discrete parts which give DNS effective record stockpiling, area name proprietorship the executives and confirmation, and DNS in reverse similarity, which may all be separately supplanted or have singular advancements. It utilizes a Distributed Hash Table to store DNS records in an appropriate manner and a blockchain to oversee area name proprietorship and uses open and private key encryption for marking and checking records.[6]

2.3. C. InterPlanetary File System (IPFS)

The Interplanetary File System (IPFS) is a shared circulated document framework that tries to associate all registering gadgets with a similar records arrangement. At the end of the day, IPFS gives a high throughput content-tended to square stockpiling model, with content tended to hyperlinks. This structures a summed up Merkle DAG, an information structure whereupon one can assemble formed document frameworks, blockchains, and even a Permanent Web. IPFS consolidates an appropriated hash table, a boosted square trade, and a self-ensuring namespace. IPFS has no single point of failure and hubs are not required to confide in one another.[7]

2.4. D. Content Addressing

Its multi-hash checksum, including joins particularly recognize all substance. When being diverged from content-tended to capacity, a regular nearby or arranged stockpiling gadget is alluded to as area tended to. In an area tended to capacity gadget, every component of information is put away onto the physical medium, and its area recorded for later use. CAS stockpiling works most effectively on DNS information which doesn't change frequently. In these partnerships, enormous zone records will be put away for as much as 10 years, without any progressions and rare access. CAS is intended to make the looking for a given report content brisk and confirm that the recovered DNS record is indistinguishable from the one initially put away.

2.5. E. Peer Discovery

While discovering all DDR network peers, flask server adds source code (Fig. 2) to the IPFS network and generates hash value of the source code. if user tampers the source code then that user will not able to connect the DDR network. Then using IPFS dht findprovs, IPFS finds all the peers who are having same source code. After getting all DDR user's IPFS id, DDR finds all the active users at that movement.

3. PROBLEM DEFINITION

To access any internet resource, we require the IP address of that resource provided by Domain Name System (DNS). To make sure that the data over the internet is secured, it is necessary that the internet be secure. To make sure that the internet is secured, the DNS must be secure. Current DNS (Fig. 3) are maintained by private organizations, governments and Internet Service Provider (ISP), which cannot be trusted directly. The current DNS system is also

```
def register_nodes():
    res = api.add('app.py')
    h = res['Hash']
    os.system("ipfs dht findprovs "+h+"> peers")
    time.sleep(10)
    lis = []
    with open("peers", "r") as f:
        _ = f.readline()
        for line in f:
            line.rstrip("\n")
            li = api.dht_findpeer(line[:-1])
            li = li['Responses'][0]['Addrs']
            for ele in li:
                if ele.split("/")[1] == "ip4":
                    a = ele.split("/")[2]
                    try:
                        response = request.get(f'http://{a}:5000/chain', timeout=1)
                        if response.status_code == 200:
                            if a != "127.0.0.1": lis.append(a)
                    except: print("", end="")
```

Fig. 2: Peer Discovery

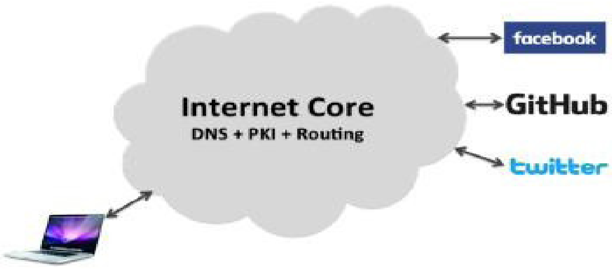


Fig. 3: Existing System of the current Domain Name Service

vulnerable to attacks such as DNS spoofing, DDoS attacks, cache poisoning, and DNS amplification, which must be overcome to have a reliable and trustworthy DNS. +People make the internet for people.[8]

In some cases, DNS has to undergo censorship by the government authorities, which violates the privilege to web get to, otherwise called the privilege to broadband or opportunity to associate. A central DNS resolver accesses the current DNS. The centralized nature of this system has few shortcomings. If the DNS resolver fails, the entire network comes to a standstill and response time will also be more significant.[9]

4. PROPOSED METHODOLOGY

A Domain Name System looks the same for end user, but internally, it works in a totally different way. DDR will be a (Fig. 4) completely distributed Domain Name Service functioning over a Blockchain. DDR, instead of replacing the DNS protocol, adds robustness to the system as a whole. Internally, DDR signs all DNS records using public/private keys, providing additional security internal to the DNS system. DDR uses newer authentication practices and a means of decentralized proof of ownership. Because this system is intended to be reverse compatible with the existing DNS protocol, we serve the data provided by the IPFS after the blockchain has authenticated it to other DNS clients. DNS nodes incorporated into the DDR system will not request data from other DNS servers and will only exchange data via IPFS.

Every time a request is made for DNS, the nearest peer should process and serve the request made. IPFS does this job. IPFS reduces latency by serving the DNS request in a minimal amount of time by processing it from the nearest block possible from where it was queried. All the peers of the blockchain system have a complete list of DNSs. When a DNS request is made, it tries to find the required DNS in the nearest of the blocks or zones possible. This zone name is sent at the DDR system, which redirects the respective zone and responds. When the user types the name of a particular website in browser, the DDR will send a request to the blockchain that will send the zone file name. This name will be forwarded to IPFS by DDR. The IPFS will search the zone file from the nearest peers and return the website to the browser.

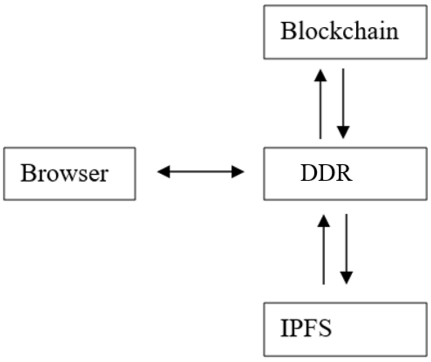


Fig. 4: Block Diagram which shows the various components which are needed in order to create DDR

DDR software is the major component of the system through which all other modules coordinate. It is the end point for the system. DDR is backward compatible with DNS protocol. DDR communicates with browser via port 53. For the end-user DNS encapsulates all the working of the system and it looks like a traditional DNS (Fig. 5) for the end-users using it.[10]

5. IMPLEMENTATION

5.1. A. DDR Software

This module is completely written using Python. DDR will be installed in every system. It will be the endpoint for the system. DDR makes the system backward compatible with DNS protocol. DDR will communicate with the browser via port 53. For the end user DNS will encapsulate all the working of our system, and it looks like a traditional DNS for the end-users using it. DDR software is the main component of the system through which all other modules coordinate. Since it is developed in Python which is compatible (Fig. 6) with all major platforms/OS like Windows, macOS and Linux making it platform-independent and easy to setup (Fig. 7).

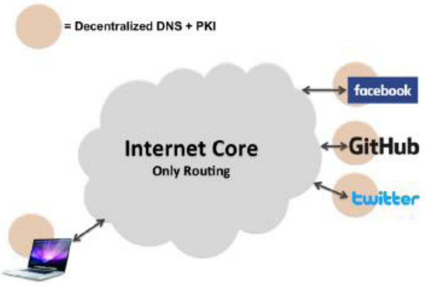


Fig. 5: Proposed System for Decentralized Domain Registry using blockchain

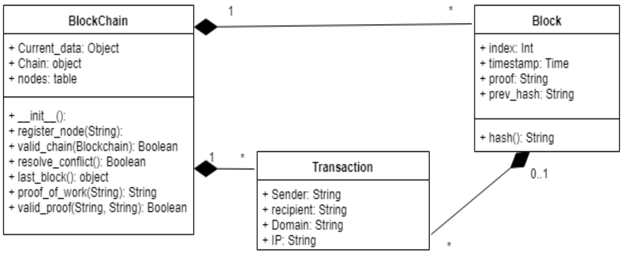


Fig. 6: Context level Diagram

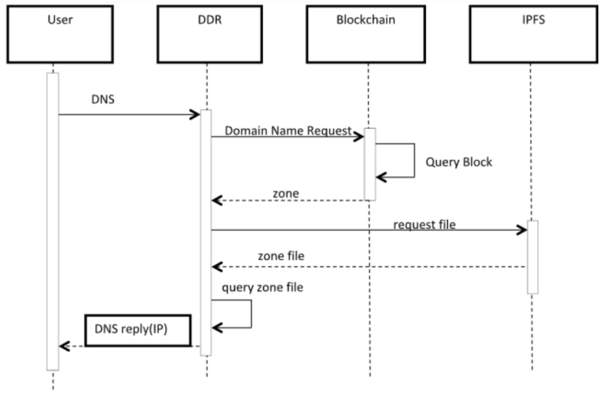
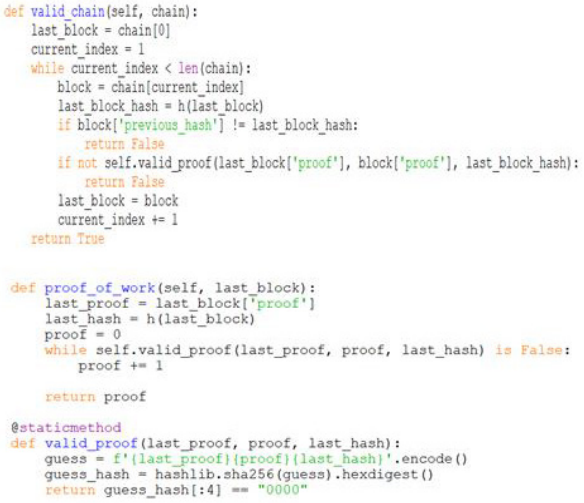


Fig. 7: Technical conversion of the distinct components into virtual modules



```
def valid_chain(self, chain):
    last_block = chain[0]
    current_index = 1
    while current_index < len(chain):
        block = chain[current_index]
        last_block_hash = h(last_block)
        if block['previous_hash'] != last_block_hash:
            return False
        if not self.valid_proof(last_block['proof'], block['proof'], last_block_hash):
            return False
        last_block = block
        current_index += 1
    return True


 def proof_of_work(self, last_block):
     last_proof = last_block['proof']
     last_hash = h(last_block)
     proof = 0
     while self.valid_proof(last_proof, proof, last_hash) is False:
         proof += 1

     return proof

 @staticmethod
 def valid_proof(last_proof, proof, last_hash):
     guess = f'{last_proof}{proof}{last_hash}'.encode()
     guess_hash = hashlib.sha256(guess).hexdigest()
     return guess_hash[:4] == "0000"
```

Fig. 8: Blockchain Module

5.2. B. Blockchain Module

The blockchain module is responsible for providing security, robustness and decentralization to the system. Blockchain will maintain all the work related with ownership and electronic trust. All the blockchain blocks will be stored in JSON (Fig. 8) that makes it easy to query upon large data size. Each peer in the network will have one copy of the blockchain, consisting of all the blocks present in the chain. The blockchain module will distribute any new block for verification and validation over the entire peer network. Using the distributed consensus approach, the block will be added to the chain. Blockchain module will serve to DNS module by providing a distributed ledger.[11]

5.3. C. IPFS Interface

This module will be using IPFS APIs to distribute and retrieve the zone files over the network. It makes use of content addressing to locate the zone files in the nearest peer in the network. It will help reduce the latency of DNS requests giving the response much faster which is one of the major concerns of DNS (Fig. 9). IPFS module will serve to DDR module by providing it the required zone file. The zone file fetched by IPFS will be processed by DDR and the required web page location (IP address) and other details.[12]

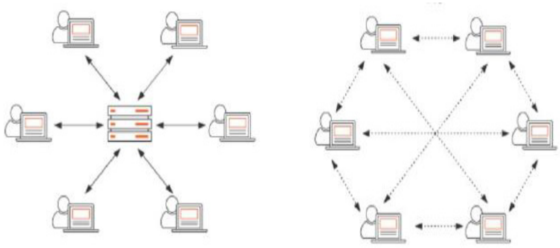
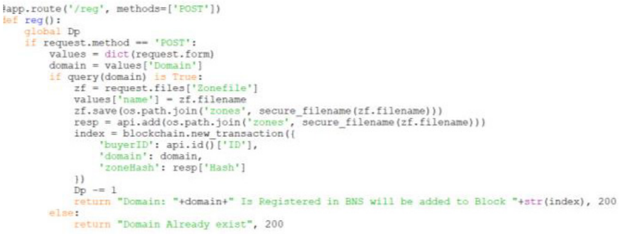
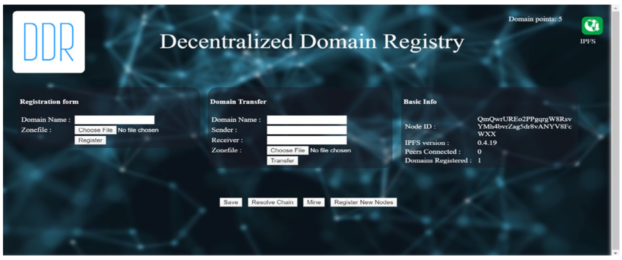


Fig. 9: (a) Traditional HTTP Network (b) IPFS P2P Network



```
@app.route('/reg', methods=['POST'])
def reg():
    global Dp
    if request.method == 'POST':
        values = dict(request.form)
        domain = values['Domain']
        if query(domain) is True:
            zf = request.files['Zonefile']
            values['name'] = zf.filename
            zf.save(os.path.join('zones', secure_filename(zf.filename)))
            resp = api.add(os.path.join('zones', secure_filename(zf.filename)))
            index = blockchain.new_transaction({
                'buyerID': api.id()['ID'],
                'domain': domain,
                'zoneHash': resp['Hash']
            })
            Dp -= 1
            return "Domain: "+domain+" Is Registered in BNS will be added to Block "+str(index), 200
        else:
            return "Domain Already exist", 200
```

Fig. 10: Web Interface Module



Domain points: 5

IPFS

Decentralized Domain Registry

Registration form

Domain Name :

Zonefile : Choose File No file chosen

Register

Domain Transfer

Domain Name :

Sender :

Receiver :

Zonefile : Choose File No file chosen

Transfer

Basic Info

Node ID : QmQwrUREo2PPgqrgW8RsvYMh4bvrZag5dr8vANYV8FcWXX

IPFS version : 0.4.19

Peers Connected : 0

Domains Registered : 1

Save Resolve Chain Mine Register New Nodes

Fig. 11: Dashboard

Table 1: Experimental Response Time

Server	*IP Address*	*Response Time (s)*
Google Public DNS	8.8.8.8	0.2515
Cloudflare DNS	1.1.1.1	0.2606
DDR	127.0.0.1 Localhost	0.0653

5.4. D. Web Interface

Web interface will be hosted on all local machines who have DDR software installed. Web interface helps us to manage domain name credits, ownership rights (Fig. 10). Using web interface users will buy or sell the domains. It also keeps account of domain credits for every individual user in the network of peers. Web interface will be developed in HTML5, CSS3 and web sockets. Web sockets play major role in communication between web interface and DDR.

6. RESULTS

The following experimental results have been identified as per our testing on sample data. These tests have been carried out using 3 different DNS servers, Google public DNS, Cloudflare DNS, and DDR. (Table 1)

Dashboard will help users manage (buy and transfer domain names) and check the DDR node's information. (Fig. 11).

7. CONCLUSION

Client can question DNS door, which was an individual from the IPFS and mining system. In the event that the questioned area had a record put away in the IPFS and a proprietor built up in the system, the server would respond with the put away DNS records. In any case the server would answer with a DNS disappointment. With all of these components working together, a system with the following features is created:

Scalability: More number of users using DDR will increase the number of DDR nodes, which will eventually increase the strength of the blockchain network.

Vigor: The IPFS and Blockchain are equally vigorous to failure & attack.

Extensibility: The DNS switch similarity permits any DNS augmentation to be used, if dynamic goals is required a name server record can be put away in the IPFS to highlight a client's specific DNS servers.

Decentralization: IPFS and Blockchain can work without any controlling third party organization/server, which makes the system more secure.[13]

8. ACKNOWLEDGMENTS

We express our earnest gratefulness to our Honorable Principal Mr. S. P. Kallurkar for the encouragement and facilities. We want to express our deep sense of gratitude to Prof. Suvarna Pansambal, Head of Department of Computer, Atharva College of Engineering, Malad, Mumbai, for her generous support and valuable suggestions.

9. REFERENCES

[1] Incognito Software **(2007)**. Understanding DNS (the Domain Name System) [White Paper].

[2] Aishwarya Sreekanth and Prashant Sri. Dyn DDOS Cyberattack – a case study.

[3] Thousand Eyes Report. **(2018)** "Global DNS Performance Benchmark Report"

[4] Park, C.-J., Ahn, S.-J., & Chung, J.-W. **(1997)**. The improvement for integrity between DHCP and DNS, High Performance Computing on the Information Superhighway. The Improvement for Integrity between DHCP and DNS, 511–516.

[5] Mockapetris, P. **(1987, November)**. Domain names - implementation and specification.

[6] B. Benshoof, A. Rosen, A. G. Bourgeois and R. W. Harrison **(2016)**. Distributed Decentralized Domain Name Service IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW),1279-1287.

[7] Juan Benet, "IPFS - Content Addressed, Versioned, P2P, File System".

[8] M. Lemley, D. Levine, and D. Post **(2011**). Don't break the internet. Stanford Law Review Online, vol. 64.

[9] S. Crocker, D. Dagon, D. Kaminsky, D. D. McPherson, and P. Vixie **(2011)**. Security and other technical concerns raised by the DNS filtering requirements in the protect IP bill. [White Paper].

[10] V. Ramasubramanian and E. G. Sirer. **(2004).** The design and implementation of a next generation name service for the internet.ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, 331– 342.

[11] Swan, M **(2015)**. Blockchain: Blueprint for a new economy.

[12] S. Nakamoto **(2008)**, "Bitcoin: A peer-to-peer electronic cash system," Consulted, vol. 1.

[13] Bedford Taylor, M **(2013)**. Bitcoin and the age of bespoke silicon. Compilers, Architecture and Synthesis for Embedded Systems, 1–10.