Computer Science & Information Technology 144

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

2nd International Conference on Blockchain and Internet of Things (BIoT 2021), June 19~20, 2021, Copenhagen, Denmark.

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai (Eds),
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-42-8
DOI: 10.5121/csit.2021.110801 - 10.5121/csit.2021.110811

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 2nd International Conference on Blockchain and Internet of Things (BIoT 2021), June 19~20, 2021, Copenhagen, Denmark, 9th International Conference on Data Mining & Knowledge Management Process (DKMP 2021), 11th International Conference on Computer Science, Engineering and Applications (CCSEA 2021), 10th International Conference on Embedded Systems and Applications (EMSA 2021) and 2nd International Conference on Natural Language Computing and AI (NLCAI 2021 was collocated with 2nd International Conference on Blockchain and Internet of Things (BIoT 2021). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The BIoT 2021, DKMP 2021, CCSEA 2021, EMSA 2021 and NLCAI 2021 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, BIoT 2021, DKMP 2021, CCSEA 2021, EMSA 2021 and NLCAI 2021 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the BIoT 2021, DKMP 2021, CCSEA 2021, EMSA 2021 and NLCAI 2021.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair	**Organization**
David C. Wyld,	Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds)	Wireilla Net Solutions, Australia

Program Committee Members

Abdel-Badeeh M. Salem,	Ain Shams University, Egypt
Abdelbadeeh salem,	Ain Shams University, Egypt
Abdelgawad,	Central Michigan University, USA
Abdellatif I. Moustafa,	Umm AL-Qura University, Saudi Arabia
Abderrahim Siam,	University of Khenchela, Algeria
Abhishek Shukla,	R D Engineering College, India
Addisson Salazar,	Universitat Politècnica de València, Spain
Ahmad A. Saifan,	Yarmouk University, Jordan
Ahmed Farouk AbdelGawad,	Zagazig Univ, Egypt
Ahmed Farouk AbdelGawad,	Zagazig University, Egypt
Ajay Anil Gurjar,	Sipna College of Engineering & Technology, India
Akhil Gupta,	Lovely Professional University, Punjab
Ali A. Amer,	Taiz University, Yemen
Ali Abdrhman Mohammed Ukasha,	Sebha University, Libya
Amal Azeroual,	Mohammed V University, Morocco
Amari Houda,	Networking & Telecom Engineering, Tunisia
Amizah Malip,	University of Malaya, Malaysia
Anand Nayyar,	Duy Tan University, Vietnam
Anastasios Doulamis,	National technical University of Athens, Greece
Ankur Singh Bist,	Chief AI Scientist at Signy Advanced Technology, India
António Abreu,	ISEL, Portugal
Attila Kertesz,	University of Szeged, Hungary
Badir Hassan,	Abdelmalek Essaadi University, Morocco
Bobby Barua,	A hsanullah University, Bangladesh
Bouchra EL IDRISSI,	Information Science School (ESI), Morocco
Bouchra Marzak,	Hassan II University, Morocco
Chang-Yong Lee,	Kongju National University, South Korea
Christian Mancas,	Ovidius University, Romania
Chuan-Ming Liu,	National Taipei University of Technology, Taiwan
Claude Tadonki,	MINES ParisTech-PSL, France
Claudio Schifanella,	University of Turin, Italy
Dadmehr Rahbari,	University Of Qom, Iran
Dariusz Barbucha,	Gdynia Maritime University, Poland
Dimitris Kanellopoulos,	University of Patras, Greece
Domenico Rotondi,	Fincons SpA, Italy
Elżbieta Macioszek,	Silesian University of Technology, Poland
Fadiya Samson Oluwaseun,	Girne American University, Turkey
Felix J. Garcia Clemente,	University of Murcia, Spain
Grigorios N. Beligiannis,	University of Patras, Greece
Hala Abukhalaf,	Palestine Polytechnic University, Palestine
Hamid Ali Abed AL-Asadi,	Basra University, Iraq
Hamid Khemissa,	USTHB University Algiers, Algeria
Hamza Baniata,	University of Szeged, Hungary
Hedayat Omidvar,	National Iranian Gas Company, Iran

Hyunsung Kim,	Kyungil University, Korea
ilango velchamy,	CMR Institute of Technology, India
Ilham Huseyinov,	Istanbul Aydin University, Turkey
Islam Atef,	Alexandria University, Egypt
Israa Shaker Tawfic,	Ministry of Science and Technology, Iraq
Iyad Alazzam,	Yarmouk University, Jordan
Jawad K. Ali,	University of Technology, Iraq
Jayavignesh T,	School of Electronics Engineering, India
Jesuk Ko,	Universidad Mayor de San Andres (UMSA), Bolivia
Jia Ying Ou,	York University, Canada
Joan Lu,	University of Huddersfield, UK
Jonah Lissner,	technion - israel institute of technology, Israel
Jong-Ha Lee,	Keimyung University, South Korea
Juntao Fei,	Hohai University, P. R. China
Kamel Hussein Rahouma,	Minia University, Egypt
Ke-Lin Du,	Concordia University, Canada
Khalid M.O Nahar,	Yarmouk University, Jordan
Khandaker Foysal Haque,	Central Michigan University, Michigan
Kire Jakimoski,	FON University, Republic of Macedonia
Koh You Beng,	University of Malaya, Malaysia
LIM Wei Hong,	UCSI University, Malaysia
Luca Virgili,	University of Marche, Italy
Luisa Maria Arvide Cambra,	University of Almeria, Spain
M.A. Jabbar,	Vardhaman College of Engineering,India
Mabroukah Amarif,	Sebha University, Libya
Malka N. Halgamuge,	The University of Melbourne, Australia
Malka N. Halgamuge,	University of Melbourne,Australia
Mamoun Alazab,	Charles Darwin University, Australia
Manish Kumar Mishra,	University of Gondar, Ethiopia
Maumita Bhattacharya,	harles Sturt University, Australia
Meenakshi Sharma,	Galgotias University, India
Mihai Carabas,	University POLITEHNICA of Bucharest, Romania
Mihai Horia Zaharia,	Gheorghe Asachi Technical University, Romania
Mirsaeid Hosseini Shirvani,	Islamic Azad University, Iran
Mohamed Arezki Mellal,	M'Hamed Bougara University, Algeria
Mohamed Ismail Roushdy,	Ain Shams University, Egypt
Mohammad A. Alodat,	Sur University College, Oman
Mohammad Ashraf Ottom,	Yarmouk University, Jordon
Mohammed BENYETTOU,	University Center of Relizane, Algeria
Mohammed Mahmood Ali,	Osmania Universtiy, India
Morteza Alinia Ahandani,	University of Tabriz, Iran
Mostafa Ghobaei-Arani,	Islamic Azad University, Iran
Mourad Chabane Oussalah,	University of Nantes, France
Mu-Song Chen,	Da-Yeh University, Taiwan
MV Ramana Murthy,	Osmania University, India
Nadia Abd-Alsabour,	Cairo University, Egypt
Nahlah Shatnawi,	Yarmouk University, Jordan
Nikolai Prokopyev,	Kazan Federal University, Russia
Nisheeth Joshi,	Banasthali University, India
Oleksii K. Tyshchenko,	University of Ostrava, Czech Republic
Omid Mahdi Ebadati,	Kharazmi University, Tehran
P.V.Siva Kumar,	VNR VJIET, India

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

2nd International Conference on Blockchain and Internet of Things (BIoT 2021)

9th International Conference on Data Mining & Knowledge Management Process (DKMP 2021)

11th International Conference on Computer Science, Engineering and Applications (CCSEA 2021)

10th International Conference on Embedded Systems and Applications (EMSA 2021)

2nd International Conference on Natural Language Computing and AI (NLCAI 2021)

LEA-DNS: DNS RESOLUTION VALIDITY AND TIMELINESS GUARANTEE LOCAL AUTHENTICATION EXTENSION WITH PUBLIC BLOCKCHAIN

Ting Xiong[1], Shaojin Fu[1], Xiaochun Luo[2] and Tao Xie[1]

[1]National University of Defense Technology, Hunan, China
[2]PLA News Media Center, Beijing, China

ABSTRACT

While the Domain Name System (DNS) is an infrastructure of the current network, it still faces the problem of centralization and data authentication according to its concept and practice. Decentralized storage of domain names and user local verification using blockchain may be effective solutions. However, since the blockchain is an add-only type database, domain name changes will cause out of date records to still be correct when using the Simplified Payment Verification (SPV) mechanism locally. This paper mainly introduces Local Enhanced Authentication DNS (LEA-DNS), which allows domain names to be stored in public blockchain database to provide decentralization feature and is compatible with the existing DNS. It achieves the validity and timeliness of local domain name resolution results to ensure correct and up to date with the Merkle Mountain Range and RSA accumulator technologies. Experiments show that less than 3.052Kb is needed for each DNS request to be validated, while the validation time is negligible, and only 9.44Kb of data need to be stored locally by the web client. Its compatibility with the existing DNS system and the lightness of the validation protocols indicate that this is a system suitable for deployment widely.

KEYWORDS

Domain name system, Blockchain, RSA accumulator, Merkle Mountain Range.

1. INTRODUCTION

DNS is a distributed database with a centralized data governance model that maps the names to values online, primarily controlled by the Internet Corporation for Assigned Names and Numbers (ICANN[1]). In this regard, ICANN manages the top-level domain (TLD) and therefore controls the root name server. In practice, if a client wants to contact a host with a specific name, it must first send a query to the DNS server to obtain the host's IP address. In order to improve efficiency, the DNS server may maintain a replica of this information in its cache, based on how often the domain name is requested. In the case that the DNS server does not hold the requested knowledge, the query will be propagated to the root name server. Next, the basic name server will find the server of the corresponding TLD, and then forward the query to the corresponding authoritative name server, which may return the requested IP [13].

[1] https://www.icann.org/resources/pages/governance/bylaws-en

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 01-17, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110801

Due to its centralized management architecture, DNS root is vulnerable to many attacks. The article [14] divides the current issues facing DNS into two categories: **centralization problem and data authenticity problem**. The centralization problem is that because users default to all DNS root servers being trusted, there will be malicious servers to attack [10]. The failure of this trust anchor is far more than a theoretical threat. The controversy surrounding the closure of *wikileaks.org* shows that the trust anchor failure occurs in the real world[2]. Distributed Denial of Service (DDoS) attacks are another threat[3]. Cache poisoning [16] and renumbering issues[9] are also related to centralization problem.

Using Blockchain technology in DNS is an effective solution to both of the above problems. Because of the decentralization nature of the blockchain and the untamperability nature of the data, the DNS resolver only needs to provide proof of the existence of the information in the blockchain, and the local browser can effectively solve the problem of DNS centralization and data authority by running the verification program. However, the blockchain is an add-only database, and an attacker may provide an outdated proof of existence to spoof the client to achieve the attack. As shown in Figure 1, an "old Tx" transaction can have all the block headers stored by the light node, then the SPV can be used to prove its existence. However, if a new transaction "new Tx" is based on "old Tx" with modifications, "old Tx" is still correct. But we cannot prove that the "old Tx" is the latest unless we download the whole chain to verify that there are no further transactions. Blockchain has the natural advantage of storing unmodified data, but this correspondence may be adjusted if the pair of <name, value> the DNS is stored, such as in the case of marketplace transactions for domain names. Attackers may take stale transactions(may store the old pair of <name, value>) to trick users, but users don't perceive them. However, compared with the traditional network, the performance of the public chain is very low, and it is difficult to directly resolve the domain name on the blockchain in the production environment. It is feasible to expand the security of the original domain name system, rather than pushing it back. In this paper, we try to solve these problems and design a DNS extension called LEA-DNS.

Contributions:

(1) Drawing on the Namecoin's architecture, we designed a UTXO-based structure for storing and transacting DNS <name, value> pairs to fit our system design.
(2) We draw on the block structure design of miniChain[3] and boneh[1] to simplify the blockchain design applied to stateless blocks as a public blockchain database for storing key-value pairs.
(3) We design a system called LEA-DNS, that allows users to perform enhanced verification of domain name resolution result locally. The system not only verifies the validity of the data, but also the timeliness of the result.
(4) Simulated experimental results show that each DNS response with verification does not exceed 3.052 Kb in size, and only a small amount of data needs to be stored locally (less than 9.44 Kb). Local validation time takes less than 10ms.

Organization of the Paper:

The rest of this paper is organized as follows. The related works are presented in Section 2. Then we give a brief introduction to our LEA-DNS system in Section 3. In Section 4, we present the details of the design of our system. In Section 5, we theoretically and experimentally evaluate our prototype implementation of LEA-DNS. Finally, we conclude our paper in Section 6.

[2] https://news.netcraft.com/archives/2010/12/03/wikileaks-org-taken-down-by-us-dns-provider.html
[3] https://en.wikipedia.org/wiki/Distributed_denial-of-service_attacks_on_root_nameservers

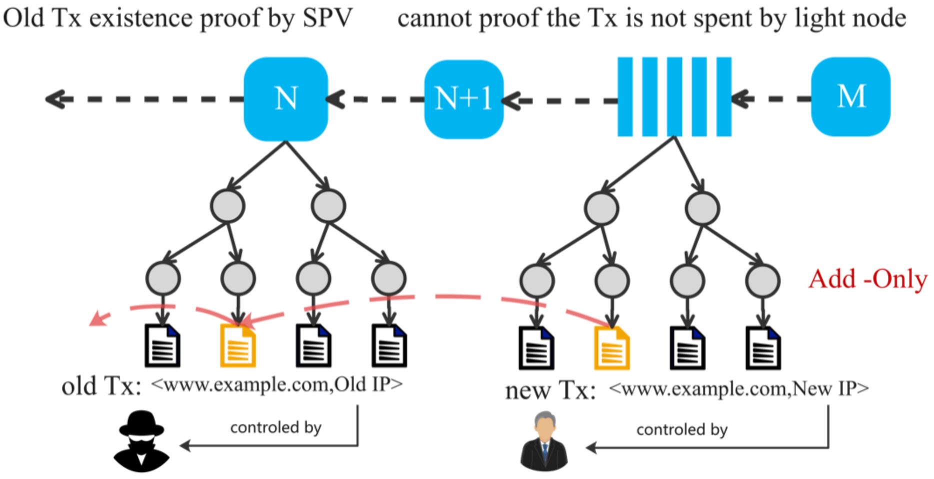

Old Tx existence proof by SPV
cannot proof the Tx is not spent by light node
N
N+1
M
Add -Only
old Tx: <www.example.com,Old IP>
new Tx: <www.example.com,New IP>
controled by
controled by

Figure 1. The Unspent Proof Problem of Transactions in Blockchain by Light Node. Old Tx existence proof can be proved by SPV and old Tx spent proof cannot be proved only by light node, so only give a SPV, the light node may be cheat by the invalid old Tx.

2. Related Work

Decentralized systems were in principle used to improve the robustness and availability of domain name resolution tasks as well as enabling the feature of by passing censorship campaigns and tampering [6, 7,17].

In the Byzantine fault-tolerant DNS, a client sends a request to all the replicas that runs the Byzantine fault-tolerant consensus algorithm and waits for enough authenticated replies. It can tolerate one-third malicious servers behaving arbitrarily. However, Byzantine fault-tolerant systems increase the communication overhead squared back as the number of nodes increases. So, the performance of Byzantine fault-tolerant algorithm decreases quadratically as more servers are added. In DHT based DNS schemes, DDNS and Overlook are peer-to-peer name services designed to enhance load balancing and fault tolerance properties. DDNS is a DNS alternative using a peer-to-peer distributed hash table built on top of Chord. The Overlook is based on Pastry. Both DDNS and Overlook have much higher latencies than conventional DNS.

With the birth of bitcoin, blockchain technology is increasingly being used in distributed DNS technology. The Ethereum name service (ENS) uses smart contracts to manage the *.eth* registrar by means of bids and recently added the support for .onion addresses. Namecoin is a cryptocurrency based on Bitcoin, with additional features such as decentralized name system management, mainly for the *.bit* domain. It was the first project to provide an approach to address Zooko’s triangle[4] since the system is secure, decentralized and user-chosen names (human meaningful). Nevertheless, contrary to well-established blockchains like Bitcoin, Namecoin’s main drawback is its insufficient computing power, which makes it more vulnerable to the 51% attack. Blockstack is a well-known blockchain-based domain name storage system that overcomes the main drawbacks of Namecoin. The architecture of Blockstack separates control and data planes, enabling seamless integration with the underlying blockchain. EmerDNS[5] is a decentralized domain name system that supports all the range of DNS records. Nebulis[6] is a

[4] http://en.wikipedia.org/wiki/ Zooko%27s triangle

[5] https://emercoin.com/en/documentation/blockchain-services/emerdns/emerdns-introduction

[6] https://www.nebulis.io/

globally distributed directory that relies on the Ethereum ecosystem and smart contracts to store, update, and resolve domain records. Moreover, Nebulis proposes the proposal of using off-chain storage (i.e. IPFS) as a replacement for HTTP. OpenNIC[7] is a hybrid method during which a group of peers manage the name space registration, but the name resolution task is completely decentralized. OpenNIC provides DNS resolution and namespace over a collection of domains, including those maintained by blockchain solutions like New Nations [8] and EmerDNS. In addition, the OpenNIC resolver has recently added access to domains managed by ICANN. Additionally, to namespace registrar, users can even create their own TLD upon request [8, 13, 15].

However, none of them consider the data authenticity problem properly. Though the SPV capability is available, clients need to pay too much overheads to verify the resolution results. BlockDNS [14] give a solution, but it still cannot solve the problem showed in Figure 1(explained in Section 1).

3. OVERVIEW OF LEA-DNS

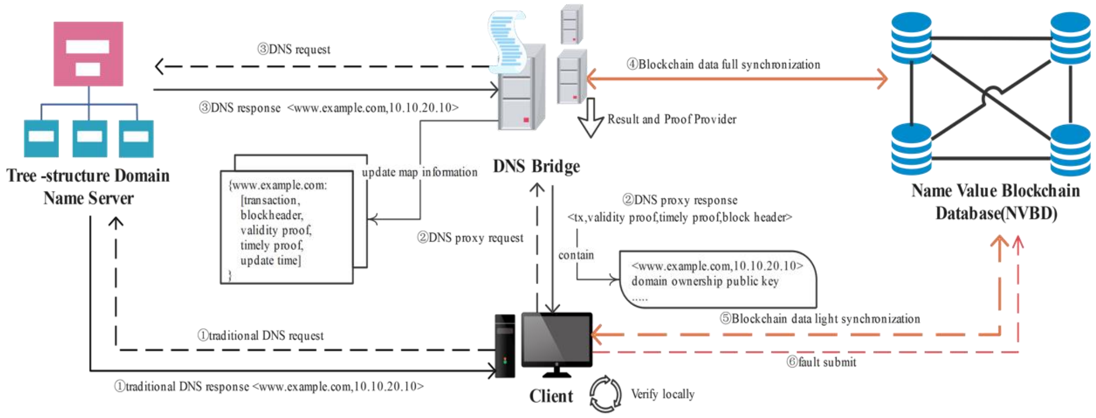

③DNS request
③DNS response <www.example.com,10.10.20.10>
④Blockchain data full synchronization
Result and Proof Provider
Tree -structure Domain Name Server
update map information
DNS Bridge
Name Value Blockchain Database(NVBD)
{www.example.com:
[transaction,
blockheader,
validity proof,
timely proof,
update time]
}
②DNS proxy response
<tx,validity proof,timely proof,block header>
②DNS proxy request
contain
<www.example.com,10.10.20.10>
domain ownership public key
.....
⑤Blockchain data light synchronization
①traditional DNS request
⑥fault submit
①traditional DNS response <www.example.com,10.10.20.10>
Client
Verify locally

Figure 2. System Architecture of LEA-DNS.

As shown in Figure 2, LEA-DNS is composed of four main components:

3.1. Tree-structure DNS

Tree-structure DNS is the traditional tree architecture domain name system. To maintain good compatibility with the LEA-DNS, it can be deployed directly without any modification to the current DNS architecture. In other words, LEA-DNS is transparent to the legacy DNS.

3.2. Name-Value Blockchain Database(NVBD)

Name-Value Blockchain Database (NVBD) is a decentralized way of storing <name, value> pairs that require enhanced validation. The design of NVBD is borrowed from Namecoin, Blockstack and miniChain. The <name, value> is stored directly in the transaction, allowing domain name registration, assignment, and transfer operations. Due to the immutability nature of the blockchain, we consider the data stored by the blockchain itself to be authenticated (the data has been verified by enough honest nodes when confirmed). Since the operation of each

[7] https://www.opennic.org/
[8] http://www.new-nations.net/

transaction in the blockchain is controlled by the public and private keys, we bind <name, value> to the transaction, and the ownership of the domain name is signed with the private key, we can easily verify the data through the public key in the transaction, which confirm the source of authenticated.

3.3. DNS Bridge

DNS Bridge is a set of proxy servers that handle requests and provide verifiable DNS response. The DNS Bridge is required to synchronize block and transaction information with the NVBD as step 4) in Figure 2. The DNS request is specially processed to obtain a transaction containing a <name, value> pair and proof of validity and timeliness of the result, which is sent to the Client. Note that DNS Bridge is a trusted institution, because it can’t forge real proof. If the proof provided by DNS Bridge can be verified locally, it can be trusted by users. In order to prevent DNS Bridge from being attacked by DOS, multiple nodes can provide services.

3.4. Client

Client can firstly request services directly from the traditional DNS. Secondly, it can send a verifiable request to DNS Bridge as step 2). In step 5), the Client needs to maintain communication with the NVBD at the same time but only needs to synchronize the latest block headers and save a small amount of other data to verify the transaction locally about guaranteeing validity and timeliness. If the validation fails, the error-proof message is submitted to the NVBD as step 6) in Figure 2.

4. DESIGN OF THE SYSTEM

In this section, we focus on the main techniques used to design each part of the system. While many of the market mechanisms in Namecoin were examined in the article[11], this section focuses on the technical details, including transaction design and the block structure of NVBD, DNS Bridge and Client.

4.1. Transactions Design

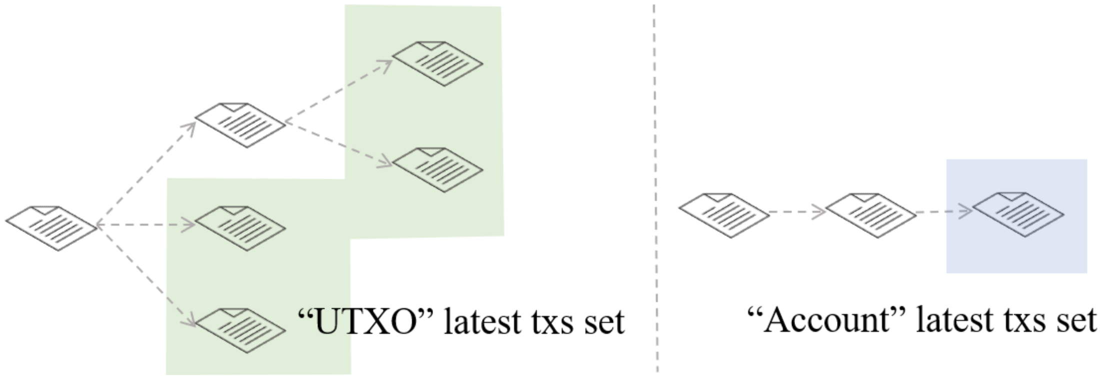
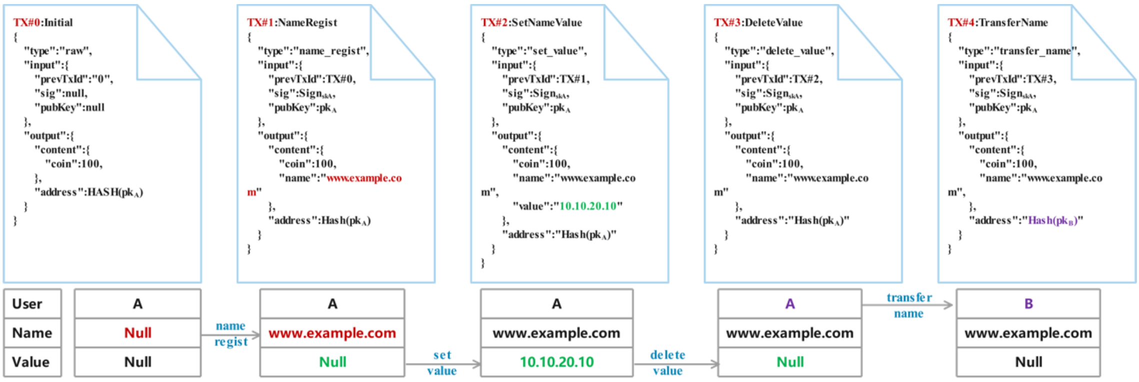


Figure 3. The difference between UTXO model and Account model.

As shown in Figure 3, because of the parallel characteristics of UTXO model, it can process multi transactions at the same time without mutual influence. The set of unspent transactions “UTXO” can be used as the set of the latest transactions. The Account model is based on the account, all transactions are added serially, and the latest transaction set size is only 1. From the perspective of transaction, UTXO model is more suitable for timeless proof, because it contains more transactions than Account model. If the account is taken as the basic unit of proof, the Account model can also be applied. For convenience, this paper uses UTXO model.

Let's assume that domain name provider A has a public-private key pair (pk_A, sk_A) and its user address is $HASH(pk_A)$. The public-private key pair for domain name provider B is (pk_B, sk_B), and its user address is $HASH(pk_B)$. The key to a domain name store with blockchain as the storage interpretation is that the data is transparent, traceable, and verifiable, but anonymity can be ignored, so our user address is a direct public key hash, and the user can also perform transactions on the same address. To simplify the situation, we only consider the transaction of one-to-one addresses, and do not consider the transaction fee, so we omit the index field of the transaction output, a simple process design is shown in Figure 4.



```
TX#0:Initial
{
  "type":"raw",
  "input":{
    "prevTxId":"0",
    "sig":null,
    "pubKey":null
  },
  "output":{
    "content":{
      "coin":100,
    },
    "address":HASH(pkA)
  }
}
```

```
TX#1:NameRegist
{
  "type":"name_regist",
  "input":{
    "prevTxId":TX#0,
    "sig":SignskA,
    "pubKey":pkA
  },
  "output":{
    "content":{
      "coin":100,
      "name":"www.example.co
m"
    },
    "address":Hash(pkA)
  }
}
```

```
TX#2:SetNameValue
{
  "type":"set_value",
  "input":{
    "prevTxId":TX#1,
    "sig":SignskA,
    "pubKey":pkA
  },
  "output":{
    "content":{
      "coin":100,
      "name":"www.example.co
m",
      "value":"10.10.20.10"
    },
    "address":"Hash(pkA)"
  }
}
```

```
TX#3:DeleteValue
{
  "type":"delete_value",
  "input":{
    "prevTxId":TX#2,
    "sig":SignskA,
    "pubKey":pkA
  },
  "output":{
    "content":{
      "coin":100,
      "name":"www.example.co
m"
    },
    "address":"Hash(pkA)"
  }
}
```

```
TX#4:TransferName
{
  "type":"transfer_name",
  "input":{
    "prevTxId":TX#3,
    "sig":SignskA,
    "pubKey":pkA
  },
  "output":{
    "content":{
      "coin":100,
      "name":"www.example.co
m",
    },
    "address":"Hash(pkB)"
  }
}
```

User	A	A	A	A	B
Name	Null	www.example.com	www.example.com	www.example.com	www.example.com
Value	Null	Null	10.10.20.10	Null	Null

Figure 1. NVBD Transactions Design and the State Change. The conversion from transaction TX#0 to transaction TX#4 represents four operations on a domain name: registration, assignment, deletion and transfer.

Regist Name:

Domain name provider A first needs a "raw" transaction as its initial transaction, as shown in Figure 4, TX#0. When A needs to register a domain name,it takes the "raw" transaction as input and outputs a transaction of type "name regist". The output contains the domain name field. If the domain name is duplicated, the transaction will not be packaged into the block.

Set Name Value:

Assigning a value to a registered domain name or a transferred domain name is known as an IPv4 address in LEA-DNS. When A needs to specify an IP address for the domain name, it takes the transaction "name regist" as input and generates a transaction of type "set value".

Delete Value:

When A needs to reassign a value to a domain name or to transfer a domain name, A needs to enter a transaction of type "set value" and generate a "delete value" type of transaction, the original assignment is deleted.

Transfer Name:

Transfer of a registered domain name. When A needs to transfer its domain name to B, A needs to enter a transaction of type “name regist” or “delete value” and generate a transfer of type “transfer name” transaction, the output address of which is the hash of the public key of B. At this point, the public key of B is not public and is anonymous to a certain extent.

Update Value:

The value update operation is similar to the domain assignment operation. Its input and output are both a transaction of type “set value”.

All transactions are verified similarly to Bitcoin, but with the addition of a determination of domain duplicity.

4.2. Block Design

First, we’d like to understand about accumulators. Basically, a cryptographic accumulator[5] is an algorithm to mix an outsized set of values into one short commitment, and enables to compute a brief membership witness (or nonmembership witness) of any element that has (or not) been accumulated. RSA accumulator is predicated on modular exponentiation under the strong RSA assumption[12]. In decentralized public blockchains where no single trusted accumulator manager exists, the essential RSA accumulator doesn’t satisfy the need, anyone who knows the secret keys p and q can use the Euclidean theorem $\varphi(N) = (p-1)(q-1)$ to calculate the order of RSA group, which may further forge any membership and nonmembership witness. Boneh[1] built a stateless blockchain[4] supported UTXO commitment by using the RSA accumulator, which needs plenty of deletion operations. Since the complexity of deletion operation is $O(n^2)$, the efficiency of the accumulator updates would drop rapidly when the amount of deletion operations increases. We use the Chen’s work[3],which divides the UTXO to STXO(i.e., spent transactions outputs) and TXO(i.e., all transactions outputs).A transaction in UTXO indicates its validity like a transaction in TXO but not in STXO.

Since LEA-DNS allows user validation of <name, value> to be done locally, the simplest idea is to store the full blockchain in the NVBD to validate transactions of the “set value” type. However, storing the full blockchain data would significantly increase the storage cost for the user. A more lightweight approach is similar to Bitcoin’s light wallet, where only the block header data is stored locally, and the full node sends the transaction’s SPV proof to verify the transaction’s validity in the blockchain. But the SPV scheme doesn’t solve the problem of whether the transaction is the most recent, or to determine if the transaction is a UTXO except all UTXOs data needs to be synchronized locally. But UTXO grows at a rate that the average user can’t afford. These solutions are not feasible. We modified the structure of Chen’s work[3] to allow users to verify the validity and timeliness of transactions by storing only a small amount of data.

The design architecture of the block is shown in Figure 5.

STXO Commitment:

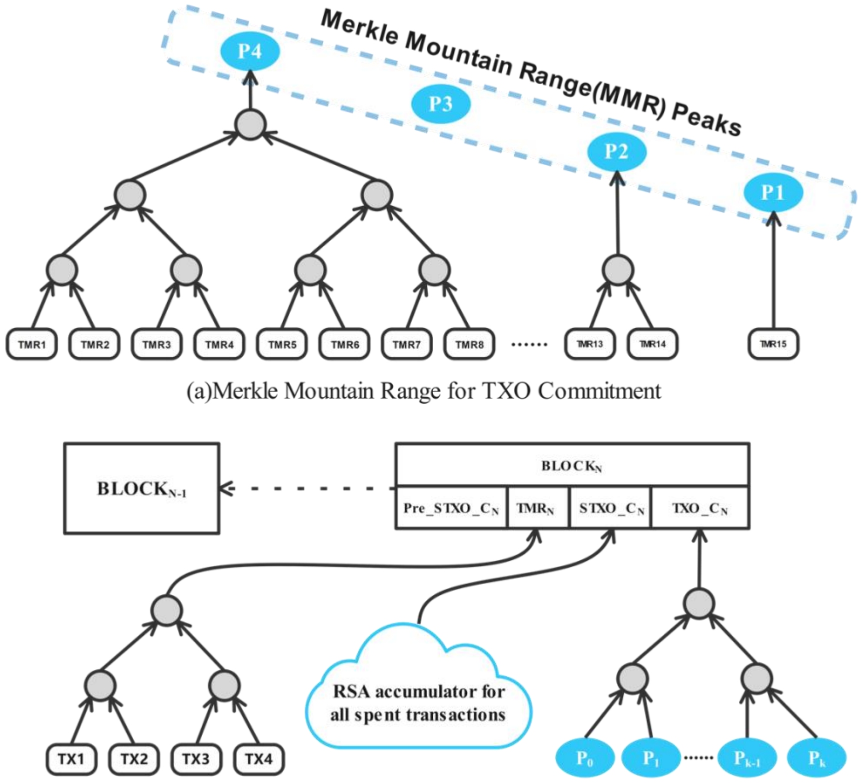
STXO_C is an append-only data structure which contains all spent transaction outputs, removing the time-consuming deletion operations needed by UTXO commitment. Specifically, each block header contains an accumulator which represents the current STXO set. A transaction can be provided a nonmembership witness which specifies that the transaction was not spent before.

STXO_C is essentially an RSA accumulator, and when a transaction is added to a block, we simply add the UTXO spent on that transaction to the accumulator. The initialization accumulator is generated by a security parameter λ and returns an accumulator A_0 .

The whole process of handling transactions is similar to Bitcoin, except that it needs to update the accumulator by marking the transaction input as spent when a blockchain mining node receives a transaction. The update algorithm accepts an old accumulator A_t and a transaction TX, and updates A_t to A_{t+1} by performing a $HashToPrime$(i.e., a function that transfer hash to prime) of the input transaction in TX. The number of transactions in a block means the times an accumulator needs to be updated from Pre_STXO_C to STXO_C.

TXO Commitment:

TXO_C is a commitment for all transactions. Traditional verification can check the Merkle path from transaction to TMR directly, but this requires the verifier to store all block headers. To reduce the verifier's storage overhead and speed up this verification approach, we use the MMR approach. The user only needs to store all the MMR Peaks in Figure 5(a), provided with the Merkle path from the transaction to the TMR and the Merkle path from the TMR to the MMR root which is constructed by MMR Peaks. The number of MMR Peaks increases logarithmically with the length of the blockchain, so this overhead is small.



(a)Merkle Mountain Range for TXO Commitment

(b)Key fields in block

Figure 2. NVBD block architecture. (a) represents the Merkle Mountain Range (MMR)[2] which connects all the block headers. (b) represents the main fields in block header containing 4 parts:(1)TMR organizes all transactions within a block through a merkle structure, (2) STXO_C represents a commitment for all spent transactions, (3)TXO_C represents a commitment for all MMR Peaks through a normal merkle structure. (4) Pre_STXO_C represents the previous block STXO_C.

4.3. DNS Bridge Design

DNS Bridge is required to synchronize all of NVBD's block information and update the DNS and transaction map information based on the transactions in the block. It forwards the user's DNS request, and finally form a verifiable DNS response back to the user. In order to let user verify the validity and timeliness of the returned <name, value> pair, it is necessary to generate the validity and timeliness proof of the "set value" type of transactions corresponding to <name, value>.

Validity Proof:

A validity proof of a transaction is a proof of the existence of a transaction in the blockchain. The proof of existence of a transaction is divided into two parts. 1) Merkle path of the transaction to TMR . 2) The Merkle path from TMR to MMR root(i.e., TXO_C).

Timely proof:

A timely proof of a transaction is a proof of the non-existence (i.e., unspent) of the transaction from the beginning of the generated block n to the specified block $m(m > n)$. In Li's paper[12], this is called a nonmembership witness. The situation described above is full timely proof. If n is not the block where the transaction is located inside, we take Δ to represent the difference in height from the located block height to the height of the $block_m$, then call it a Δ-timely proof. We use $tf_m(x_n)$ to represent the timely proof of the transaction x_n to $block_m$.$\Delta tf_m(x_n)$ means in the latest Δ blocks, the x_n is not been spent. The algorithm for generating $tf_m(x_n)$ is shown in Algorithm 1. Assuming $x_n \in UTXO$ and$x_n \notin STXO$, the algorithm first obtains the set of all unspent transactions $STXO_{n:m}$ from the $block_n$ to $block_m$, and simultaneously performs a $HashToPrime$. Then calculate the product of all primes as p. Since x_n and p are different prime numbers, it is easy to find $Bezout$ coefficients a and b such that $ax + bp = 1$ by using extended Euler's theorem. The final calculation $d = A_n^a$, returns (d, b) as $tf_m(x_n)$.

Algorithm 1 Timely Proof

```
Input:
$block_n$ previous accumulator $A_{n-1}$(Pre_STXO_C$_n$);
$block_m$ accumulator $A_m$(STXO_C$_m$);
    proof transaction $x_n$;
    all spent transactions from $block_n$ to $block_m$ presented by STXO$_{n:m}$ .
Output:
    timely proof $tf_m(x_n) \leftarrow \{d, b\}$.
$p \leftarrow 1$
$x_n \leftarrow HashToPrime(x_n)$
for $stx$ in STXO$_{n:m}$ do
    $p \leftarrow p \cdot HashToPrime(stx)$
end for
$a, b \leftarrow Bezout(x_n, p)$
$d \leftarrow A_n^a$
return $tf_m(x_n) \leftarrow \{d, b\}$
```

Algorithm 2 Timely Proof Update

```
Input:
$block_m$ accumulator $A_m$;
    old proof $tf_m(x_n)$;
    transaction $x_n$;
    all spent transactions from $block_m$ to $block_{m'}$ presented by $\mathrm{STXO}_{\mathrm{m:m'}}$ .
Output:
    new timely proof $tf_{m'}(x_n)$.
$p \leftarrow 1$
$d, b \leftarrow tf_m\ (x_n\ )$
for $stx$ in $\mathrm{STXO}_{\mathrm{m:m'}}$ do
    $p \leftarrow p \cdot HashToPrime(stx)$
end for
$a', b' \leftarrow Bezout(HashToPrime(x_n), p)$
$r\ \leftarrow a'b$
return $tf_{m'}(x_n) \leftarrow \{dA_m^r, b'b\}$
```

Timely Proof Update:

$tf_m\ (x_n\ )$ with the growth of the blockchain will become out of date. Assume the current block height is m', in order to update the proof, we can recalculate the $tf_{m'}(x_n)$, and the new $tf_{m'}(x_n)$ can be computed based on $tf_m\ (x_n\ )$.The specifific update algorithm is shown in Algorithm 2. Here we give the proof:

Suppose there is a Timely Proof $tf_m(x_n) \leftarrow \{d, b\}$. it satisfies the following conditions (1) through Algorithm3:

$$d^x A_m^b = A_n; x = HashToPrime(x_n) \tag{1}$$

when add some TXs to A_m,the prime product of TXs is p, from Algorithm2, we can get the new timely proof $tf_{m'}(x_n) \leftarrow \{dA_m^r, b'b\}$ and new accumulator $A_{m'} = A_m^p$,the conditions (2) provided:

$$\begin{gathered} a'x + b'p = 1 \\ r = a'b \\ \hat{d} = dA_m^r \\ \hat{b} = b'b \end{gathered} \tag{2}$$

If the proof update Algorithm2 is true, then equation (3) should be satisfied.

$$\hat{d}^x A_{m'}^{\hat{b}} = A_n \tag{3}$$

By procedure (3), we can verify that equation (4) always holds.

$$\begin{aligned} \hat{d}^x A_{m'}^{\hat{b}} &= (dA_m^r)^x A_{m'}^{\hat{b}} \\ &= d^x A_m^{a'bx} A_m^{pb'b} \\ &= d^x A_m^{b(a'x+b'p)} \\ &= d^x A_m^b = A_n \end{aligned} \tag{4}$$

Therefore, DNS Bridge returns a message in a format similar to <tx, block header, validity proof, timely proof>, where tx is a "set value" type of transaction that contains the <name, value> pair and the domain name provider's public key information.

4.4. Client Design

The client receives the messages returned by DNS Bridge and can verify the validity and timeliness of the message content. The prerequisite for the client to be able to perform validation is that 1) only some latest block header information needs to be synchronized 2) all MMR Peaks are saved.

Validity Verify:

Initially, users are required to download all MMR Peaks collections. After that, the MMR Peaks collection can be updated by itself each time a new block header is synchronized. The validity verify is divided into two steps: 1) Check if the Merkle Root of MMR Peaks is equal to the TXO C of the latest synchronized block header, if it is equal, proceed to the second step, otherwise resynchronize the block and check again. 2) Check Merkle path form transaction to TMR and Merkle path from TMR to MMR Peaks, if so, the verification is passes or succeeds, otherwise the verification fails.

Algorithm 3 Timely Proof Verify

```
Input:
$block_n$ accumulator $A_n$;
$block_m$ accumulator $A_m$;
    timely proof $tf_m(x_n)$;
    transaction $x_n$.
Output:
Verify result $true\ or\ false$.
1: $x_n \leftarrow HashToPrime(x_n)$
2: $a, b \leftarrow tf_m(x_n)$
3: return $d^x A_m^b == A_n$
```

Timely Verify:

The user receives the timely proof $tf_m(x_n)$ and the block header where x_n is located, and the existence proof of x_n has been verified by validity verify. We extract the accumulator field STXO_C A_n from the block header, the latest block STXO_C A_m, timely proof $tf_m(x_n)$and the transaction x_n that needs to be verified as parameter inputs, as shown in Algorithm 3. The $\Delta tf_m(x_n)$ can be verified similarly, but the input block header is $block_{n-\Delta}$ rather than $block_n$.

5. COST ANALYSIS AND EVALUATION

5.1. Experiment Settings and Parameters

Based on the source code of RSA-Accumulator[9], Merkle Tree[10], and Merkle Mountain Range, we implemented a prototype of NVBD, DNS Bridge, and Clien[11]t with Python language. We use the

[9] https://github.com/oleiba/RSA-accumulator

[10] https://github.com/Tierion/pymerkletools

[11] https://github.com/jjyr/mmr.py/blob/master/mmr/mmr.py

RSA accumulator with 3072 bit-modulus, 128 bits prime representative, and the Merkle root is set to 32 bytes. All these parameters are considered to be safe enough in the field of cryptography. We run all our experiments on our desktop computer equipped with one 3.7 GHz Intel Core i9 processor, 64 GB RAM, and perform 10 runs and report their average for each data point of running time. We test the performance of the accumulator update, proof generation and update, proof verification. We also find the problem that timely proof generation time is a little high and we give our solution. It should be noted that our LEA-DNS is an extension based on DNS. We don't care about the specific performance or implementation of the public blockchain, such as throughput, confirmation time, network structure, etc. We only consider the additional consumption when the public blockchain supports timeliness verification, which may be a limitation of this paper.

We denote that the interval between block generation is T, the size of the block header is S_h, the average size of transactions is S_t. For convenience, we assume that each transaction will consume one input (UTXO) and generate two outputs (UTXOs). Denote m as the average number of transactions per block. The average number of UTXOs consumed per block will be $m/2$, and the average number of UTXOs generated per block will be m. Suppose n is total the number of UTXOs and L is the length of the current blockchain state.

5.2. NVBD Extra Cost

NVBD's full node requires additional work to add to the original Bitcoin node to update the accumulator STXO_C and update TXO_C. The accumulator needs to be updated for each transaction in the block, and the time complexity of the update is *O(m)*. The update of the accumulator in a block can be divided into the process of calculating the product of all transactions using the *HashToPrime* function and the process of product modular exponentiation. The experimental results are shown in Figure 6, where the time grows linearly with the number of transactions in the block.

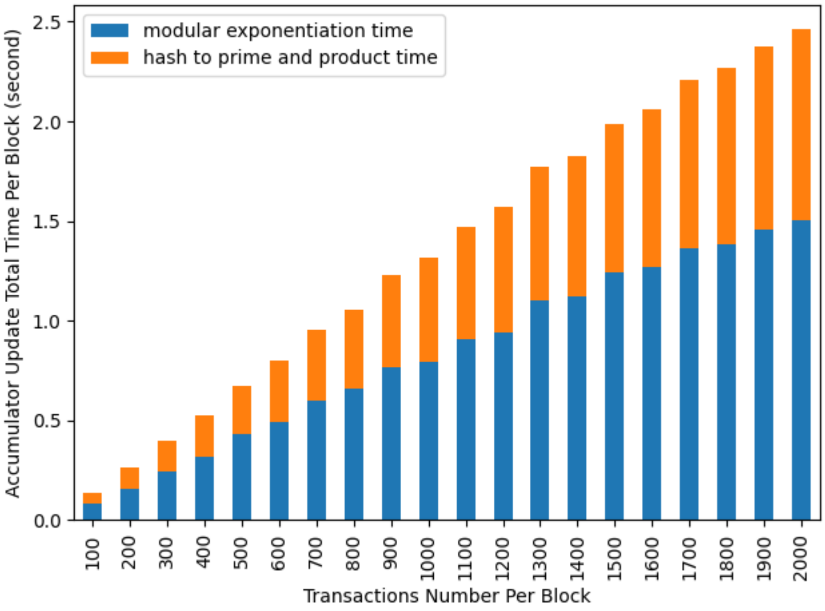


Figure 3. Performance of Accumulator Update Per Block.

The update of TXO_C is in two steps: the first step is to insert the new TMR to the MMR with a time complexity of $O(log(L))$. The second step is to construct the MMR Peaks into TXO_C via the Merkle structure. Since the number of MMR Peaks is $\lceil log(L) \rceil$, the time complexity of constructing TXO_C from MMR Peaks is $O(log(L))$. Thus the time complexity of TXO_C update will be $O(log(L))$. By the blue curve in Figure 6, even when the block height L grows to 2^{24}, the time to insert the TMR into the MMR is still less than 1ms.

5.3. DNS Bridge Cost

Firstly, DNS Bridge needs to synchronize all the data of NVDB with $S_t m + S_h$ scale each block generation, so the synchronization bandwidth between DNS Bridge and NVBD must be at least: $(S_t m + S_h)/T$. if we use the Bitcoin parameters, it only needs about 2Kb/s bandwidth. However, it needs to provide service for clients with high bandwidth.

Validity Proof Generation:

After data synchronization, DNS Bridge needs to construct a validity proof of transaction, construct a Merkle tree of new transactions to the TMR, and insert the TMR into the MMR. After the MMR tree is updated, the Bridge only needs to provide the Merkle path from the updated MMR each time clients request validity proof, and the time complexity is $O(1)$. Therefore, the time complexity for DNS Bridge to update the validity proof is $O(m) + O(log(L))$. By the orange curve in Figure 6, it takes only about 0.06ms to generate a TMR to TXO_C proof even when the block height grows to 2^{24}.

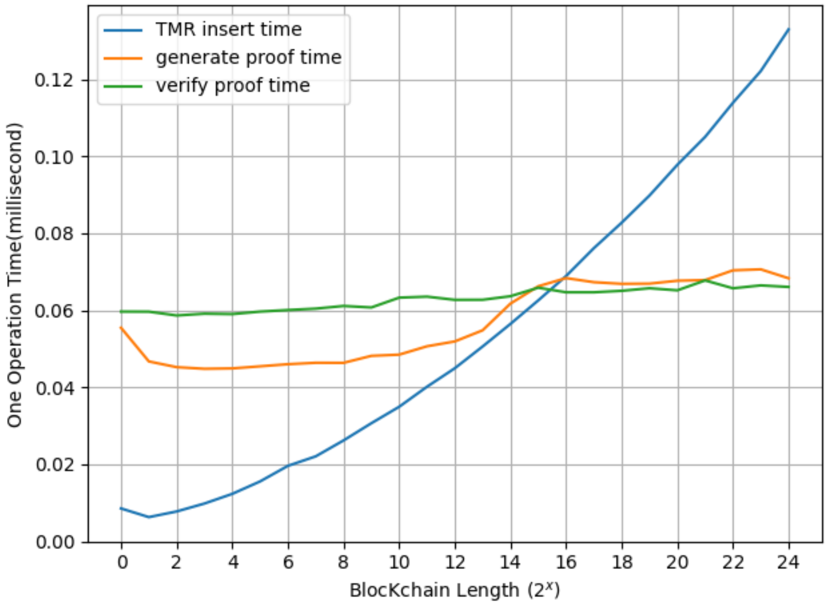


Figure 4. Perfomance of the MMR Operation With the Blockchain Growing.(Based on the first TMR).

Timely Proof Generation:

When the Client requests a timely proof of a DNS response, the timely proof needs to be generated if the transaction is newly generated. From Algorithm 1, the time complexity of timely proof generation is $O(m)$ for a block. And the time complexity of generating a $\Delta t f_m(x_n)$ is $\Delta O(m)$. If the transaction's timely proof already exists, then it needs to be updated or reconstructed. For each update from $\Delta tf_m(x_n)$ to $\Delta tf_{m+1}(x_n)$, the time complexity is $(\Delta + 1)O(m)$, which is the same as reconstructing a timely proof. However, we don't need to cache the previous STXO's prime products when using update operation. As we can see from the Figure 8, when the number of blocks (transactions) grows, the time it takes for timely proof to be generated will gradually increase. To keep the time within an acceptable range, we propose the idea of **phase validation**: the DNS Bridge provides only $\Delta tf_m(x_n)$, rather than full timely proof, where Δ has a limitation. A timely proof $\Delta tf_m(x_n)$ will be validated by many Clients and submitted to NVBD if an error is found, so outdated proofs($tx \notin STXO_{n:m-\Delta}$) can be omitted and we just need to keep the recent Δ range of proofs reliable if the error can be advertised to all the Clients during Δ blocks time. From the Figure 8, we can see that the timely proof generation will cost much time(second level), so DNS Bridge can also provide the $tf_{m-1}(x_n)$proof to ease the pressure of computing. The proof generation can also be easily accelerated by parallel computation.

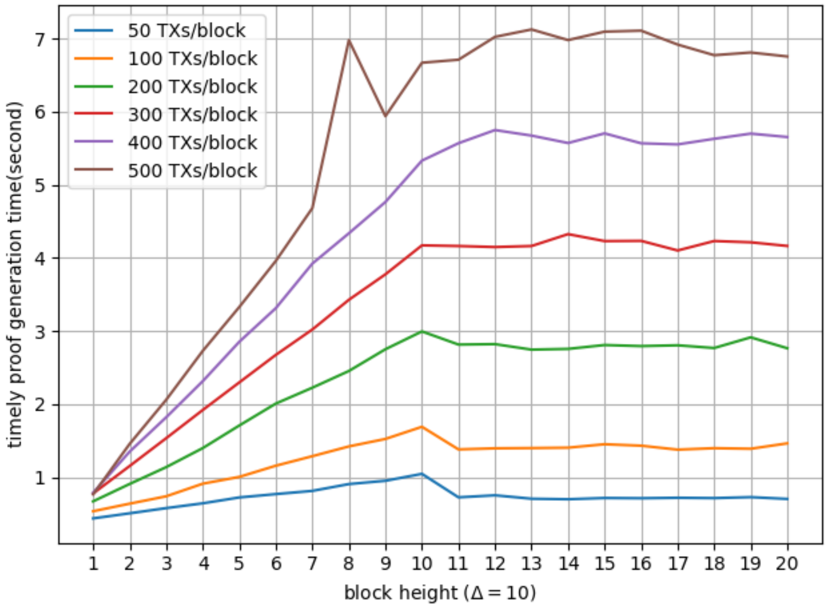


Figure 5. Performance of the Timely Proof Generation ($\Delta tf_m(x_n), \Delta = 10$).

5.4. Cilent Cost

Client will get <tx, blockheader$_n$, valid proof, timely proof> from DNS Bridge and blockheader$_m$ from NVBD as a response. Suppose $m < 2^{10}, L < 2^{20}, \Delta = 10$ and each transaction is assumed to be 300 bytes. Client also need to get the latest MMR Peaks when you first start, they are less than $32 * 20 = 640(bytes)$. The blockheader adds two accumulators (STXO_C and Pre_STXO_C) and TXO_C compared with the original Bitcoin, and its size is assumed to be 80(Bitcoin blockheader)+2*384(RSA accumulator)+32(TXO_C) = 880(bytes).The spatial complexity of validity proof is $O(log(m)) + O(log(L))$, so the size of validity proof size is at most 32 * (10 + 20) = 960(bytes).The size of timely proof is fixed to two constants, and the total size is 416 bytes. The blockheader$_m$ size is 496 bytes. Therefore, each time you interact with DNS Bridge, the size of the validation data is less than:

$$300 + 880 + 960 + 416 + 496 = 3052(\text{bytes})$$

The local data that needs to be maintained is all the MMR Peaks, with its spatial capacity scale being $log(L)$ and the latest blockheaders from $\mathrm{blockheader}_{m-\Delta}$ to $\mathrm{blockheader}_{m}$. At the first time when the client starts, the data size for synchronization will be less than:

$$640 + 10 * 880 = 9440(bytes)$$

Validity Verify Time:

Firstly, client need to insert TMR_m into MMR to update the local MMR Peaks, the time complexity is $O(log(L))$. Next, generate Merkle root from all MMR Peaks, and compare whether it is equal to $\mathrm{TXO\_C}_m$, whose time complexity is also $O(log(L))$. Verifying the path from transaction tx to TMR_nhas a time complexity of $O(m)$, followed by verifying the path from TMR_n to MMR root, which has a time complexity of O(log(L)). Therefore, the time complexity of each validity verify is $O(m) + 3O(log(L))$ which includes TMR insert time and verify proof time in Figure 7. The insert time is less than 0.14ms by the blue curve and the verify proof time is about 0.06ms through the green curve. So, the total time is less than 0.2ms when L grows to 2^{24}.

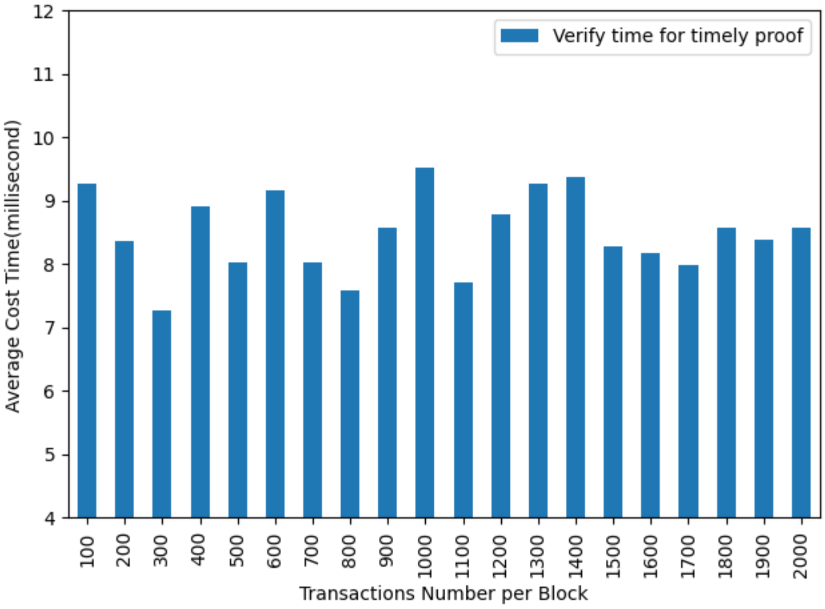


Figure 6. Performance of the Timely Proof Verify.

Timely Verify Time:

Timely proof only needs to perform a $HashToPrime$ and a modular exponentiation operation, thus the time complexity is $O(1)$. From experiments, the timely verify time is less than 10ms through the Figure 9, which is almost no burden to the Client. We conclude with a summary of the temporal complexity of three components and the time cost level, as show in Table 1.

Table 1. Time complexity and cost for NVDB full node, DNS bridge, and Client every block or transaction generation round.

Part		Time Complexity	Cost Time Level
NVBD	Acc Update	O(m)	second
	MMR Insert	O(log(L))	millisecond
DNS Bridge	Validity Proof	O(m)+O(log(L))	millisecond
	Timely Proof	$\Delta O(m)$	second
Client	Validity Verify	O(m)+O(log(n))	millisecond
	Timely Verify	O(1)	10 milliseconds
	Message Size	< 3.052Kbytes	
	Storage Size	< 9.44Kbytes	

6. CONCLUSIONS

This paper proposed a blockchain-based decentralized naming system called LEA-DNS to solve the centralization problem and data authenticity problem. We find the problem of record obsolescence in the blockchain when DNS <name, value> has been changed and propose our solution. LEA-DNS enables name owners to apply domain names and maintain authoritative server information on blockchain in a decentralized manner which mainly consists of NVBD, DNS, and Clients. The UTXO mechanism, RSA accumulator, and Merkle Mountain Range have been used for the blockchain design called NVDB. DNS Bridge will generate the validity proof and timely proof for the verifiable DNS request and the response size is only a few hundred bytes. Clients will verify the response with little time. Our simulated results show that the Clients will only need storage no more than 9.44Kb data locally, the overhead of verification message is less than 3.052Kb and the verification time is below 10ms. LEA-DNS is also compatible with current legacy DNS architectures. In the future work, we will deploy this system in a real network to further test its performance.

ACKNOWLEDGEMENTS

This paper is supported by the national Science Foundation of China, under the project No.61472476.

REFERENCES

[1] Boneh, D., B¨unz, B., Fisch, B.: Batching techniques for accumulators with applications to iops and stateless blockchains. In: Annual International Cryptology Conference. pp. 561–586. Springer (2019).

[2] Bunz, B., Kiffffer, L., Luu, L., Zamani, M.: Flyclient: Super-light clients for cryptocurrencies. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 928–946.

[3] Chen, H., Wang, Y.: Minichain: A lightweight protocol to combat the utxo growth in public blockchain. Journal of Parallel and Distributed Computing 143, 67 – 76(2020).

[4] Chepurnoy, A., Papamanthou, C., Zhang, Y.: Edrax: A cryptocurrency with stateless transaction validation. IACR Cryptol. ePrint Arch. 2018, 968 (2018).

[5] Fazio, N., Nicolosi, A.: Cryptographic accumulators: Defifinitions, constructions and applications. Paper written for course at New York University: www. cs. nyu.edu/nicolosi/papers/accumulators. pdf (2002).

[6] He, G., Su, W., Gao, S., Yue, J.: Td-root: A trustworthy decentralized dns root management architecture based on permissioned blockchain. Future Generation Computer Systems 102, 912 – 924 (2020).

[7] Huynh, T.T., Nguyen, T.D., Tan, H.: A decentralized solution for web hosting. In: 2019 6th NAFOSTED Conference on Information and Computer Science (NICS). pp. 82–87.

[8] Jiang, Y., Bai, H., Yang, H.: The messaging model design based blockchain and edge computing for the internet of things. In: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). pp. 604–608.

[9] Jin, Y., Fujikawa, K., Harai, H., Ohta, M.: Secure glue: A cache and zone transfer considering automatic renumbering. In: 2015 IEEE 39th Annual Computer Software and Applications Conference. vol. 2, pp. 393–398.

[10] Jones, B., Feamster, N., Paxson, V., Weaver, N., Allman, M.: Detecting dns root manipulation. In: International Conference on Passive and Active Network Measurement. pp. 276–288. Springer (2016).

[11] Kalodner, H.A., Carlsten, M., Ellenbogen, P., Bonneau, J., Narayanan, A.: An empirical study of namecoin and lessons for decentralized namespace design. In: WEIS. Citeseer (2015).

[12] Li, J., Li, N., Xue, R.: Universal accumulators with efficient nonmembership proofs. In: Katz, J., Yung, M. (eds.) Applied Cryptography and Network Security. pp. 253– 269. Springer Berlin Heidelberg, Berlin, Heidelberg (2007).

[13] Patsakis, C., Casino, F., Lykousas, N., Katos, V.: Unravelling ariadne's thread: Exploring the threats of decentralised dns. IEEE Access 8, 118559–118571 (2020).

[14] Ren, S., Liu, B., Yang, F., Wei, X., Yang, X., Wang, C.: Blockdns: Enhancing domain name ownership and data authenticity with blockchain. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6.

[15] Shi, P., Liu, H., Yang, S., Zhang, Y., Zhong, Y.: The inherent mechanism and a case study of the constructional evolution of the jointcloud ecosystem. IEEE Internet of Things Journal 7(3), 1561–1571 (2020).

[16] Trostle, J., Van Besien, B., Pujari, A.: Protecting against dns cache poisoning attacks. In: 2010 6th IEEE Workshop on Secure Network Protocols. pp. 25–30.

[17] Yoon, W., Im, J., Choi, T., Kim, D.: Blockchain-based object name service with tokenized authority. IEEE Transactions on Services Computing 13(2), 329–342 (2020).

AUTHORS

Ting Xiong received the bachelor's degree from National University of Defense Technology, Changsha, China, in 2019. He is currently pursuing the engineering degree with the National University of Defense Technology, Changsha, China. His current research interests include blockchain architecture, consensus algorithm, and application.

AN APPLICABILITY OF BLOCKCHAIN MODEL IN BUSINESS USE CASE - A TECHNICAL APPROACH

Anitha Premkumar

Department of Computer Science and Engineering, Presidency University
Rajankunde, Bangalore, Karnataka, India

ABSTRACT

Business network brings many organizations close together to achieve their desired goals and profit from it. People from different organizations may or may not know each other but still can be part of a business network. A major challenge with these business networks is how to provide trust among people and data security. Blockchain is another means through which many organizations in the current digital age are overcoming these problems with ease. Blockchains have also changed the way the business transactions with clients take place. Blockchain is a decentralized distributed ledger in a peer to peer network which can be public or private, and it enables individuals or companies to collaborate with each other to achieve trust and transparency between business and its clients. Many implementations of blockchain technology are widely available today. Each of them have their own strengths for a specific application domain. They can fundamentally alter electronic communications with a potential to affect all sorts of transaction processing systems. However, there are still many challenges of blockchain technology waiting to be solved such as scalability and adoptability. In this paper, we provide the knowledge on Blockchain technology and we present the applicability of blockchain in the business models and also discuss the relevant use cases for Banking and Supply Chain models.·

KEYWORDS

Blockchain, Secure Web Transaction, Decentralized Distributed Ledger, Peer to Peer Network.

1. INTRODUCTION

Blockchain is primarily defined as a shared immutable ledger, or just an “unchangeable record of who owns what”. It can be used to transfer and permanently records any changes to the assets like money, crypto currency, real estate, records of any kind, identities, personal property etc., between two or more parties without the need of intermediaries. Blockchain uses combination of existing technologies like Ledger, Cryptography, and Network Technology. Internet was a major milestone in technological evolution and it led to newer business models, distributed computing and many more changes to human lifestyle over the past 25 years. Experts believe that Blockchain [6][10] will become the next wave in Internet technology as it eliminates the need of intermediaries and provides trust between various business parties.

In cryptocurrencies, we have observed how Blockchain has enabled an online payment system without central authority unlike in Internet Banking. Blockchain-based system receives data, encrypts it and stores it as digital leger called a software leger in every node of the network. The software ledger is a collection of records which are immutable in nature. The key characteristic of Blockchain is that it provides trust between different parties who are part of the business and who

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 19-29, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110802

may or may not have known each other prior to the business transaction. This is the one of the reasons why Blockchain is making its entry into business world at a faster rate than other emerging technologies post 2016. In a typical business environment, multiple parities are involved and they communicate with each other to share data and perform business transactions. The business data between various parties needs to keep it in secured place and maintained for future purposes. Earlier generation of business systems kept all their data in central repository where it was being monitored and secured by a central authority. Malicious attack, data breach and data stealing could happen easily with central system. Hence security became a major concern to centralized systems in business environment. To address the above challenge, in Blockchain-based solutions all business data is stored at secured place that is on every node's location in the network. This makes it almost impossible for malicious attacker to hack the data.

2. BLOCKCHAIN TECHNOLOGY

Blockchain technology [4] [6] [16] is a decentralized digital ledger stored across all the nodes in the peer-to-peer network (P2P). P2P network is a computer network where nodes will be directly connected to each other without intermediaries. Digital ledger is a software ledger that holds encrypted data which is immutable and shared between nodes in the Blockchain network without a middleman. The structure of Blockchain technology is such that it contains series of blocks that hold business data in them. All nodes in the network holds a copy of the Blockchain records. Any changes in the data requires permission from all nodes in the network which means verification and validation is done by every node in the network. Every node should work cooperatively to achieve the business goal with help of consensus algorithm. Consensus algorithm is an algorithm embedded into every node in Blockchain network to help the network to achieve the agreement between the nodes on data sharing and new block creation. Decentralized application (DAPP) shown in figure 1, is an application model in which all the nodes are connected to each other to share a data and store it across the blockchain network without central server.

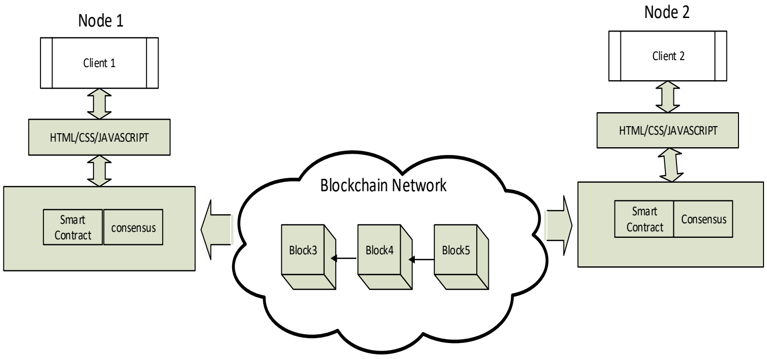


Figure 1. DAPP Model

Every blockchain node in DAPP model has front-end design for interacting with blockchain back-end network via web service calls. Commonly the front-end design is built using a programming languages like HTML, CSS and JAVASCRIPT via which user can access the back-end network. The back-end network consists of smart contract, consensus, and blockchain database. The Browser will initiate the transaction proposal, notify the result of transaction proposal. Blockchain contains a series of blocks attached together in a chain-like structure. The current block contains the address of previous block and previous block contains the address of its previous block and so on. Thus block integrity is maintained. The very first block in the blockchain is called genius block which does not contain address of the previous block. The actual block chain starts with genius block. Each block contains immutable and tamper proof record of transaction data. Data blocks are time stamped and stored in the order of transaction fashion strictly thereby supporting integrity, data transparency and data security. Indeed, blockchain provides immutable and tamper proof data, it is suitable for business to keep their data in secure manner and access by all nodes in the network. A data which is stored in blockchain is secured with help of cryptographic algorithms. Cryptographic algorithm [25][26] uses pair of keys for encryption and authenticate the data. They are namely public key and private key. For instance, in a crypto-currency blockchain, when a person purchases a blockchain wallet, he/she gets a private key. In addition to the private key, there is also a public key. As and when the person sends a transaction from blockchain wallet, the software checks and verifies the transaction and then signs it with his/her private key. This activity sends a message to entire blockchain network that the person has an account in the blockchain wallet and has the authority to transfer the fund on the public key that the person is sending from. Every asset can be converted into electronic asset and given a unique identifier to track, control, buy and sell on the blockchain. The system permits decentralized transactions of all types of e-assets between peers. Blockchain allows users to access the single version of truth to exist across the network as shown in the figure 2.

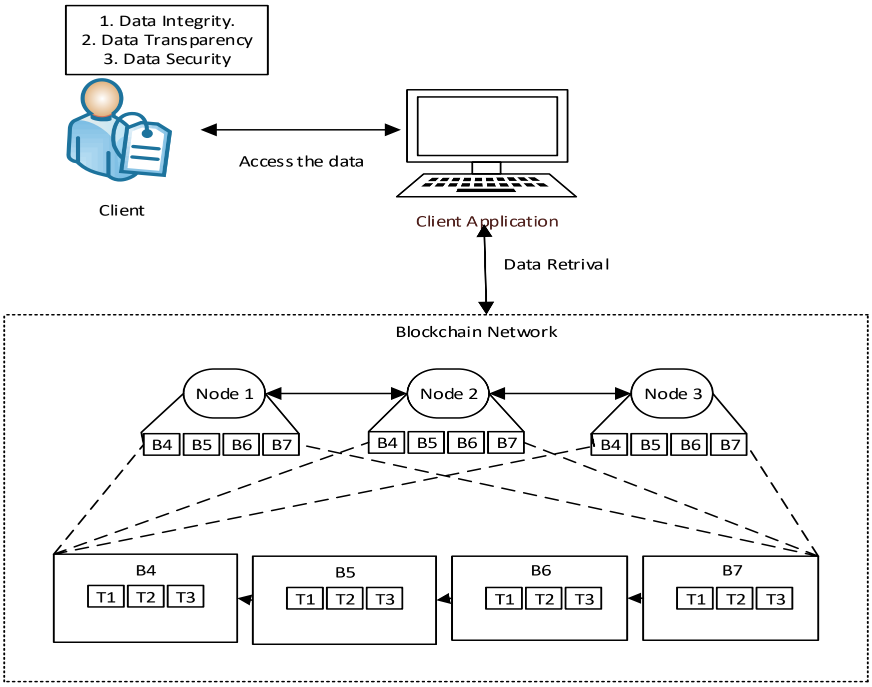


Figure 2. Accessing Single Version of Truth via Blockchain Network.

Blockchain is divided into categories based on the node’s participation into the network. If the network is of public type, then it is called public blockchain where any node can take part into the network at any time. If the network is of private type, then it is called private blockchain in which only certified nodes can be part of the network. Table 1 below highlights the differences between public and private blockchains [8][10].

Table 1: Differences between Public and Private Blockchain

Public Blockchain	**Private Blockchain**
Any node can participate	Only a certified node can participate
Decentralized such that no one has control over the network. Security enabled by the fact that data cannot be changed once validated by the blockchain.	One or more entities control the network leading to reliance on third-parties to transact. Only participating entities have knowledge of transaction and thereby there is additional security.
Transaction data is visible to all nodes in the network	Transaction data is visible only to the nodes who have got certified to be part of network.
Scalability is an issue	Scalability is not an issue
Speed – transactions per second is lesser due to higher number of nodes	Transactions per second is much higher
More secure since there are higher number of nodes in network and hackers cannot attack it easily and gain control.	More prone to attacks, data breaches and manipulation.
More chances of collusion among majority of participating nodes to gain control of the network.	No chance of collusion since each validator is known and identifiable.

3. FIRST IMPLEMENTATION OF BLOCKCHAIN

Bitcoin [3] was the first Blockchain project that got implemented using Proof of Work consensus algorithm. Consensus algorithm [2] [11] is a software agreement between nodes to work jointly and allows to take common decision to perform the task. Consensus algorithm is the backbone of Blockchain technology. There are many consensus algorithms available to implement Blockchain projects. In this section we discuss about first consensus algorithm PoW [5] [7] used in Bitcoin Blockchain Network. Bitcoin works based on crypto currency transactions. Crypto currencies are a form of electronic cash which can be used to trade assets between business parties that are connected via internet. How Proof of work algorithm works in Bitcoin [1] [3] Blockchain is described below in steps

Step1: Identity of a node is checked with help of membership service of decentralized application
Step 2: Once node's identity is validated, node can perform transactions with other nodes in the network
Step 3: Once node completes the transactions, transactions can be verified and validated by a special node in the network
Step 4: Special node create a new block by solving the complex puzzle to convert valid transactions into blocks
Step 5: Once new block is created, it gets broadcasted to other nodes in the network.
Step 6: Upon receiving a new block, nodes update their existing database called digital ledger with new block.
Step 7: Go to step 1

There are two nodes present in Bitcoin. They are called Peer node and Miner node. Peer nodes are normal nodes that execute the transactions with help of Consensus algorithm. Miner nodes are called special peer nodes that execute transactions and also validate the transactions. Miner nodes will solve complex mathematical puzzle in-order to validate the transaction and create new block. To solve complex puzzle, Miner nodes require heavy electricity and computing power. There can be many miner nodes in the Blockchain network who can compete with each other to solve the complex puzzle. The time duration to solve puzzle and create new block in Bitcoin is 10 minutes. Miners who completes task first within the duration, will be the winner of that round. He will be rewarded with some Bitcoin. He will then propagate a new block to whole network. All the nodes in the network receive a new block and update digital ledger stored at their location.

3.1. A Fork in Blockchain

What if two miners solve puzzle nearly at the same time and try to add new block into the previous chain of blocks in the network? This situation creates fork [9] in the network and it is described in Figure 3. Fork means deviation from previous history of records or suddenly a new rule is framed and followed by nodes at particular point. It means that two branches of chain will be created in the network by nodes who solves puzzle at the same time. This can happen with valid miners coincidentally who solves puzzle at same time or malicious attacker who purposefully wants to create a fork to take control over the network and hack the transaction data.

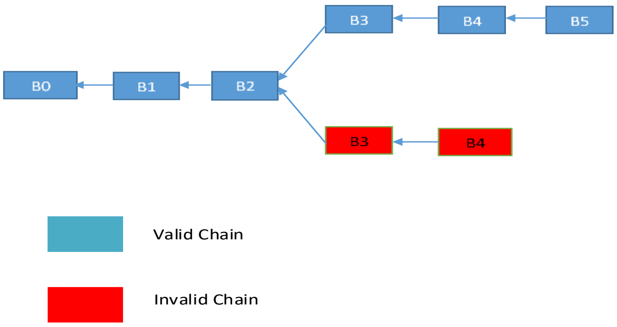


Figure 3: Fork situation in Blockchain Network

Solution: When nodes are encountered fork in the Blockchain network, the process of creating new blocks will not be stopped. Consensus algorithm allows miner to keep solving the complex puzzle and add new blocks into the branches of chain of Blocks. This process will continue and some more blocks will get added into the chain of blocks at every interval of time or each round of an algorithm executes. After some time, the longest branch of chain will be considered as valid chain and that will be considered for further processing and continued and smaller chain will be discarded.

4. Blockchain-Enabled Digital Business Models

Blockchain technology [17][18] [19][20] is becoming more and more popular in various business applications for healthcare , supply chain, education, government and banking. The reason behind this popularity is the ability to cater to constantly growing data in the distributed, decentralized digital ledger while providing ability to seamlessly share the data with all the nodes /peers in the organization network, shown in figure 4.

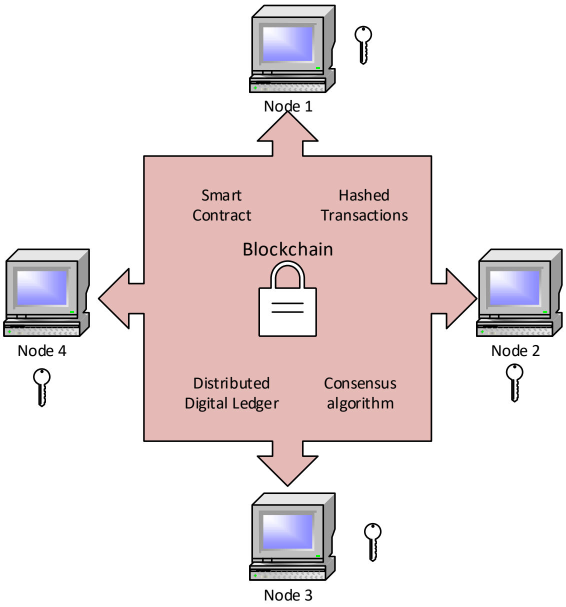


Figure 4. Seamless and Controlled data sharing between nodes within Blockchain Network

In case of healthcare system, the healthcare data is highly sensitive and it needs to be stored at a secured place with limited access. Operational/transactional data will be created and entered by authorized persons within the organization. Private Blockchain system is well suited for such an environment. In Pharma supply chain, manufacturer, distributor, retailer, transporter are the authorized people who are responsible for correctness and completeness of data. Once the drugs are manufactured and shifted to the distributor, drug details are recorded in the blockchain network. Thus every block in the network contains drug transaction details in it. Any authorized member can check for the authenticity of the drug at any time. Blockchain framework supports many platforms to provide a solution to business needs. A few popular platforms such as Ethereum, R3Corda, Ripple, Quorum are listed in Figure 5.

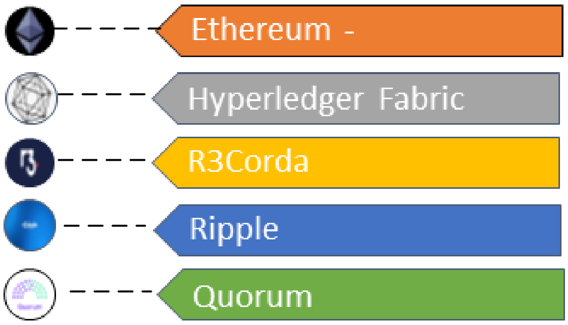


Figure 5. Popular Blockchain Platforms

When it comes to asset tracking with trust, transparency and more security, Hyperledger Fabric is most convincing framework among all. Hyperledger [21-24] from Linux foundation, is an open source permissioned distributed ledger technology platform for business enterprises. It is designed to support pluggable implementations of various components and helps us to solve variety of use cases. Hyperledger serves as a greenhouse that collaborates with the customer, developer, different vendors from various sectors to achieve its goal. It provides different consensus models that enable performance at larger scale while achieving data privacy. As mentioned previously, Consensus is an agreement between nodes in the network in-order to verify and validate the transactions and achieve the correctness of the set of transactions in the block. Blockchains can have different consensus models based on the requirements for different use cases. Use of Blockchain technology [14] [15] can provide enormous competitive advantage to businesses even though at present significant challenges are there in leveraging it in digital business models. In Figure 6, the key dimensions of Blockchain usage in business models is depicted.

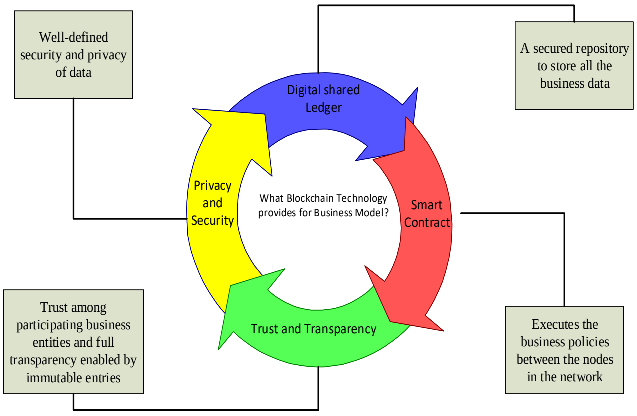

Well-defined security and privacy of data
A secured repository to store all the business data
Digital shared Ledger
Privacy and Security
What Blockchain Technology provides for Business Model?
Smart Contract
Trust and Transparency
Trust among participating business entities and full transparency enabled by immutable entries
Executes the business policies between the nodes in the network

Figure 6. Key Dimensions of Blockchain Usage in Digital Business models

4.1. Banking System [10] [12] [13]

Banking sector has been among the top to adapt blockchain to support their digital banking needs. Millions of customer transactions are processed via banking systems and it is very important to securely store this data in a secured place. Who gathers the data, who has access to the data and where it is stored are aspects critical to the banking system. In traditional systems, the transaction data gets stored in a central server for verifications, validations and maintenance. Consider a scenario where, Bob wants to sell his car to John. Bob sends his public key to John to transfer amount to his account using bank application. John uses Bob's public and his own private key of the bank application and transfers the required amount to Bob's account. Bob uses his public and private key to check the transaction account. In this scenario, Bob and John are connected via bank central system.so every transaction which is taking place via bank application, goes to the server which is maintained by bank's central authority. This server will capture all the data and stores on it. It is also responsibility of central server to maintain the data in proper way. Table 2, shows how Banks registers transaction data in their central databases.

Table 2: Bank Database without Blockchain

Sender Name	Sender Account	Receiver Name	Receiver Account	Transfer Amount	Date & Time
John	000XXXX20280	Bob	000XXXX08765	5,00,000	31/12/2019 18-00-30

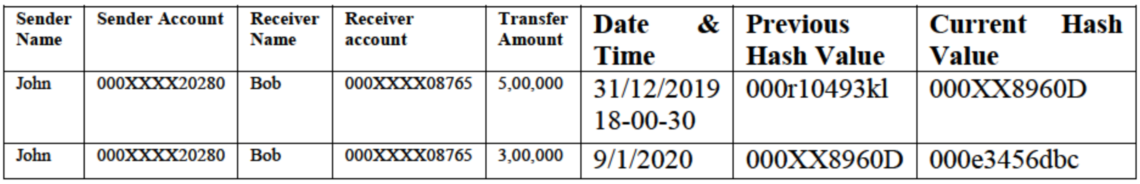
There are a few problems that are associated with centralized system –

i) Time to validate the user account and verify the account details is longer. This impacts the overall transaction time.
ii) Network is vulnerable to malicious attackers who can try to steal data or deny service to other users. Such attacks reduce the credibility of the bank in the eyes of the customers.
iii) Centralized systems are mostly monolithic in nature and require proprietary knowledge for maintenance. Thereby the time to make change is longer. Blockchain technology addresses these issues by eliminating completely a central server system and allows all the nodes to have all the transactional data in their computer in the form of a distributed digital ledger. The above mentioned example can be explained as below in a scenario where Blockchain is used.

Transaction: Bob is selling his car to John

Bob and John login to blockchain applications that are designed to work on a Peer-To-Peer network. Peer-to-Peer network means, Bob and John are connected directly without any intermediaries. Both uses their public and private key to perform transactions. Once transaction is completed, it is verified and validated by other nodes in the network and transaction data gets saved into a digital ledger. The data which is entered into ledger is immutable and is called as tamper proof data. This digital ledger will be shared with every node in the network. It is almost impossible to hack the data for network attacker as the data is stored at every single node in the network. Blockchain Digital ledger will contain the record of transaction data with hash value. Hash value is an unique value for each record of blocks in the digital ledger and shown in Table 3. In digital ledger, every record will contain two hash values.one is previous hash value and other one is current hash value. Previous hash value will have data about previous record details. Current hash value will have current transaction details.

Table 3.Bank Database with Blockchain



Sender Name	Sender Account	Receiver Name	Receiver account	Transfer Amount	Date & Time	Previous Hash Value	Current Hash Value
John	000XXXX20280	Bob	000XXXX08765	5,00,000	31/12/2019 18-00-30	000r10493kl	000XX8960D
John	000XXXX20280	Bob	000XXXX08765	3,00,000	9/1/2020	000XX8960D	000e3456dbc

4.2. Blockchain-Enabled Supply Chain [10][12]

Supply chain is at the heart of many businesses for effective product delivery. This is where the business models should have trust between different business parties and helps to supply the goods from suppliers to manufacturers to consumers. Supply chain is a complex connected network through which manufacturing companies source their raw materials and get their products to market, to sell and profit from them. It encompasses planning, controlling and execution of product flow from raw materials to finished goods. The goal of an efficient supply chain is to function in most effective and streamlined manner possible in order to achieve better performance throughout the network. The main parts of a Supply chain are

- **Planning-** Plan for resources that are required to satisfy the consumer needs
- **Sourcing-** Identify the supplier who provides goods and services needed for product
- **Manufacturing-** Activities including for manufacturing the product and testing it for quality and plan for delivery schedule
- **Delivery(Logistics)-** Collecting consumer orders, plan for safe storage and delivery, dispatching the load and receiving payments
- **Returning –** Providing return policy for goods in case of damage or unwanted product

Physical flows and information flows are two types of flow by which many organizations are linked together to form a supply chain which are i) Physical flow in supply chain deals with transportation, movement and storage of goods. It is actual flow of goods from manufacturing unit to marketplace. This is most visible flow in supply chain. ii) Information flow in supply chain allows organizations to share data about planning, controlling and execution of goods delivery. Here the information is stored in computer systems or in paper documents. In traditional supply chain models, multiple business partners are involved to supply the goods to consumers at right place and time. This scenario is given in figure 7. The business partners have to organize complex, trust-based systems to exchange and store such data with adequate security measures.

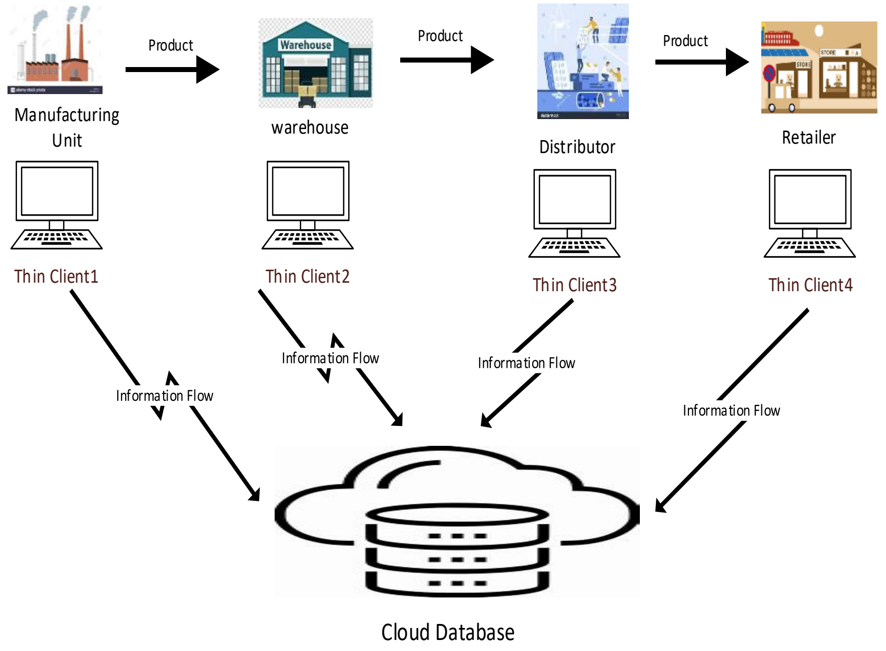


Figure 7. Supply Chain Model without Blockchain

On the other hand, Blockchain helps the business partners not only to exchange data but also have the required visibility throughout the chain. This model helps the business systems to verify the data at any time. This also helps prevent counterfeit products from getting into the supply chain.

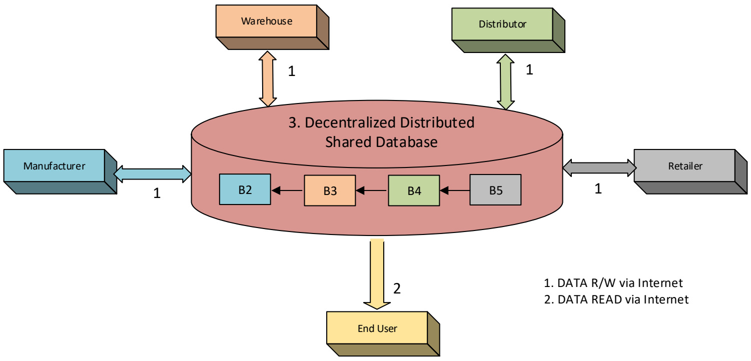


Figure 8. Blockchain Based Supply Chain Model

Blockchain-based supply chain helps businesses to track the product right from manufacturing unit to consumer location as given in Figure 8. Business data at various stages in the supply chain should captured and stored as blocks in the chain of blocks on a network. For example, details like ingredients, date of manufacturing, date of expiry etc., will be captured when goods are getting produced at the manufacturing unit. This information gets validated by other parties in the network, put into blocks and sent to all nodes in the network. When goods move from manufacturing unit to warehouse, warehouse details will be captured and stored as block which will then added into chain of blocks. This will continue till it reaches the consumer. Consumer can check their product simply by reading data from the ledger via a blockchain application. Smart contract, a piece of code embedded into the applications help the nodes to write data into ledger and read from ledgers.

5. Conclusion

Establishing trust and ensuring full transparency is the crux of business to business and business to consumer transactions. Securely storing the data and enabling role-based access to the same is another key goal. In traditional business systems, a complex system design is required to achieve these goals and the same can be quite inflexible and expensive to maintain. Blockchain is a technology that can help overcome these challenges by having a decentralized digital ledger to captures all the valuable data and corresponding transactions. The unique feature of this ledger is that it is immutable in nature which means data and transactions are not modifiable once it is stored in ledger. This helps the businesses to store data permanently in such a manner that it cannot be deleted or modified by other parties without the knowledge of the participants in the blockchain. Data in the blockchain is visible to participants in the network anywhere, anytime. Hence blockchain technology brings trust, transparency, security and visibility to all participants in the business network thereby making it the ideal choice for many businesses to adapt.

References

[1] Sathoshi Nakamoto “Bitcoin: A Peer-To-Peer Electronic Cash System”, 2008 [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] Giang-Truong Nguyen and Kyungbaek Kim,” A Survey about Consensus Algorithms Used in Blockchain” Journal of Information Processing Systems Vol.14, No.1, pp.101~128, February 2018.

[3] Florian Tschosch and Bjorn Scheuermann, “Bitcoin and Beyound: A Technical survey on Decentralized Digital Currencies” IEEE Communications Surveys and Tutorials,Volume 8, March 2016.

[4] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang, “ An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends”, 2017 IEEE 6th International congress on Big Data.

[5] Shijie Zhang, Jong-Hyouk Lee, “An Analysis of the main consensus protocols of blockchain”, ICT Express, August 20129.

[6] A white paper on " Deloitte's 2019 Global Blockchain Survey- Blockchain gets down to business"AvailableOnline[https://www2.deloitte.com/content/dam/Deloitte/se/Documents/risk/DI_2019-global-blockchain-survey.pdf

[7] Fan Yang, et.al. "Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm with Downgrade Mechanism" August 2019, Special section on Emerging Approaches to Cyber security, IEEE Access.

[8] Public-vs–private blockchain , Available online: https://www.blockchain-council.org/blockchain/public-vs-private-blockchain-a-comprehensive-comparison/

[9] Noe Elisa, Longzhi Yang, Fei Chao, Yi Cao. "A framework of blockchain-based secure and privacy-preserving E-government system" Wireless Networks, https://doi.org/10.1007/s11276-018-1883-0

[10] A Study paper on security aspects of a blockchain, TS Division, TEC Available Online: https://www.tec.gov.in/pdf/Studypaper/Security%20aspects%20of%20blockchain.pdf.

[11] Giang-Truong Nguyen and Kyungbaek Kim, " A Survey about Consensus Algorithms Used inBlockchain" J Inf Process Syst, Vol.14, No.1, pp.101~128, February 2018.

[12] Peter Verhoeven , Florian Sinn and Tino T. Herden, "Examples from Blockchain Implementations in Logistics and Supply Chain Management: Exploring the Mindful Use of a New Technology", Logistics 2018, 2, 20; doi:10.3390/logistics2030020.

[13] A white paper on "Blockchain in banking while the interest is huge, challenges remain for large scale adoption "April 18, 2017. Available Online: https://www2.deloitte.com/content/dam/Deloitte/in/Documents/strategy/in-strategy-innovation-blockchain-in-banking-noexp.pdf

[14] A white paper on "Blockchain and Suitability for Government Applications" 2018 PUBLIC-PRIVATE, Analytic Exchange Program", Available Online: https://www.dhs.gov/sites/default/files/publications/2018_AEP_Blockchain_and_Suitability_for_Government_Applications.pdf

[15] A white paper on "\Blockchain Technology in India – Opportunities and Challenges", available Online: https://www2.deloitte.com/content/dam/Deloitte/in/Documents/strategy/in-strategy-innovation-blockchain-technology-india-opportunities-challenges-noexp.pdf

[16] Michael Crosby (Google), Nachiappan (Yahoo), Pradan Pattanayak (Yahoo), Sanjeev Verma (Samsung Research America) ,Vignesh Kalyanaraman (Fairchild Semiconductor), "Blockchain Technology : Beyond Bitcoin" , Applied Innovation Review, Issuse 2, June 2016.

[17] A white paper on "Deloitte's 2019 Global Blockchain Survey". Available Online: www.deloitte.com.

[18] Swan, M. Blockchain: Blueprint for a New Economy; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.

[19] Leila Ismail and Huned Materwala," A Review of Blockchain Architecture and Consensus Protocols : Use cases, Challenges, and Solutions" , Symmetry 2019.

[20] Archana Prashanth Joshi et al, "A survey on security and privacy issues of Blockchain Technology", Mathematical Foundation of Computing, American Institute of Mathematical Sciences, May 2018.

[21] A white paper on "Hyperledger Fabric Framework". Available online: https://hyperledger-fabric.readthedocs.io/.

[22] A White paper on "An Introduction to Hyperledger". Available online: https://www.ibm.com/downloads/cas/0XMOQJNP.

[23] A White paper on "Hyperledger Architecture volume 1", Available on-line: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf.

[24] Sukhwani, H. Performance Modeling & Analysis of Hyperledger Fabric (performance Blockchain Network); Duke University: Duke, UK, 2018.

[25] Anjula Gupta et al, "Cryptography Algorithm: A Review", IJEDR, 2014.

[26] Lukman Adewale Ajao et al, "Crypto Hash Algorithm-Based Blockchain Technology for Managing Decentralized Ledger Database in Oil and Gas Industry", Multidisciplinary Scientific Journal, August 2019

DESIGN AND IMPLEMENTATION OF AN IOT BASED LPG AND CO GASES MONITORING SYSTEM

Otoniel Flores-Cortez[1], Ronny Cortez[2] and Bruno González[2]

[1]Department of Applied Sciences, Universidad Tecnologica
de El Salvador, San Salvador, El Salvador
[2]Associate researcher, Universidad Tecnologica de El Salvador,
San Salvador, El Salvador

ABSTRACT

Nowadays use of liquefied petroleum gas (LPG) has increased. LPG is an asphyxiating, volatile and highly flammable gas. In a LPG leak situation, potential health accidents are increased either by inhalation or by combustion of the gas. On the other hand, carbon monoxide (CO) is a toxic gas that comes mainly from combustion in car engines. Breathing CO-polluted air can cause dizziness, fainting, breathing problems, and sometimes death. To prevent health accidents, including explosions, in open or closed environments, remote and real-time monitoring of the concentration levels of CO and LPG gases has become a necessity. The aim of this work is to demonstrate the use of Internet of Things (IoT) techniques to design and build a telemetry system to monitor in real-time the concentration of GLP and CO gases in the surrounding air. To implement this work, as central hardware there is a microcontroller, CO and PLG sensors on the electronic station. Besides, Amazon Web Services (AWS) was used as an IoT platform and data storage in the cloud. The main result was a telematics system to monitor in real time the concentrations of both GLP and CO gases, whose data is accessible from any device with internet access through a website. Field tests have been successful and have shown that the proposed system is an efficient and low-cost option.

KEYWORDS

Internet of things, Microcontroller, Remote sensing, LPG, CO.

1. INTRODUCTION

In Latin American countries liquefied petroleum gas (LPG) is also known as “propane gas”, its use has increased in various applications: commercial, industrial and mainly residential [1]. Most home and restaurant kitchens used LPG to cook food on gas stoves. Metal cylindrical tanks were commonly used to store LPG and were placed inside the kitchen or in a nearby location. Another environment where this type of gas is found is within industrial facilities for the production and distribution of the gas itself. LPG is an asphyxiate, volatile and very flammable gas [2]. A leak can occur if it is not stored properly or because of a failure in the storage tank or because of poor handling of the circulation pipes and hoses [3]. In a gas leak situation, possible health accidents inside the house, restaurant or facility increase, either by accidental inhalation or by combustion of the gas [4]. In El Salvador country there is a history of explosive accidents caused by mishandling or leaks of PLG [5] [6] [7] [8]. Carbon Monoxide (CO) is poisonous, taste-, odour- and colourless gas derived from combustion processes of automobiles [9]. CO has fatal consequences if undetected. Intoxication caused by CO is frequent possibly leading to high

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021

DOI: 10.5121/csit.2021.110803

morbidity and mortality [10]. Symptoms of CO poisoning include dizziness, nausea, weakness, headaches, lethargy, and confusion [11]. Inside garages or closed-door parking lots, residential or commercial, are common places of concentration for CO gas [12]. People that work or user of this kind of spaces are more susceptible to suffer CO intoxication, even if they breathe CO-polluted air by a few minutes [13] [14]. In Latin America health accidents produced by CO gas has increased mainly by lack of monitoring system. [15] [16] [17] [18]. The measurement of LPG and CO gases is obtained in Parts Per Million (PPM) of concentration within the surrounding air. Threshold concentration levels for GLP to take care: 0 to 400 ppm = Normal -- 401 to 800 ppm = Hazardous -- more than 800 ppm = Explosive [19]. For CO gas: 0 to 50 ppm = Normal - 51 to 800 ppm = Dangerous - more than 800 = Deadly [20] [21]. Real-time monitoring for concentration levels of CO and LPG gases within a residential environment or in closed parking lots has become a necessity, as some places where these gases are found grows day by day. In order to help on prevention of health accidents derived from CO and LPG gases, such as poisonings and fires, it is useful to give close or open spaces with a monitoring and early warning system of the concentration levels of these gases. Previous work has related to developing LPG or CO gas monitoring stations. But these have focused on the use of high-cost technological tools [22] [23] [24] [25] or are not connected to a website in real-time [23] [26] [27]. The cloud platforms used for these previous implementation is restrictive or closed-source private or expensive The use of so-called free IoT clouds like Thingspeak or Ubidots is popular among the above studios, but they have some limitations like a limited quantity and frequency of telemetry data. [28] [29] [30] [31] [32]. Some of these previous works only focused only on monitoring one gas either LPG or CO [33] [34] [35]. This work proposed a low-cost electronic system based on IoT technologies, equipped with sensors that are capable of taking a reading of both CO and LPG gases in the surrounding air. The system is also capable of sending sensor data over the Internet and displayed on a website so security staff can aware in real-time possible leaks or high concentrations of these gases and alert users and take measures to avoid possible health damages. Rest of this paper is organized as follows. Section 2 summarizes the prototype development of the IoT system. Section 3 presents experimental results and discussion about the proposal, and Section 4 concludes and present some final comments and ideas to be tackled in future work.

2. DEVELOPMENT OF A GAS MONITORING IOT SYSTEM

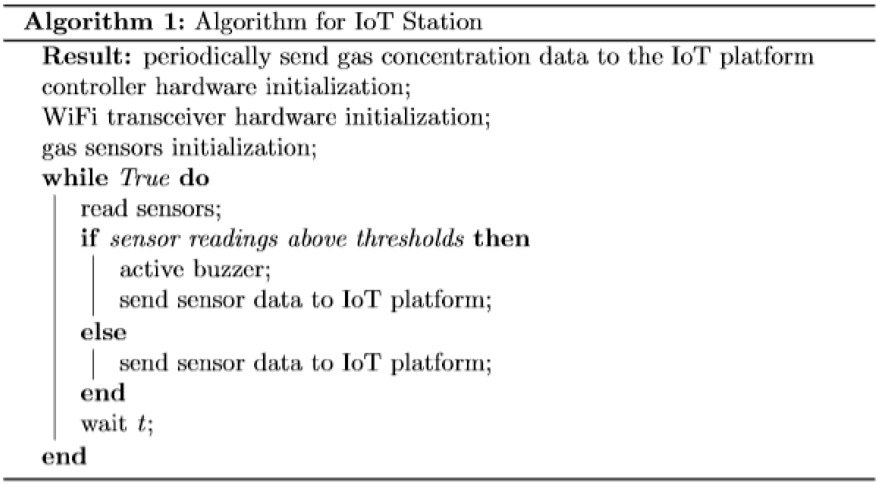
Methodological development of this proposal system was based on the IoT Architectural Reference Model [36].

2.1. Purpose & Requirements Specification

Purpose: automated PLG and CO gases concentration monitoring with Wi-Fi communication and real-time report via a web dashboard. Behaviour: electronics station with sensors capable to take measurement of gases concentration (PPM - part per million) in surrounding air, and a central digital controller programmed to do periodic reading of sensors and send collected data via Wi-Fi to a platform on the Internet. Requirement for management: the system can be monitored via the internet; programming management and configuration of the sensor station can be locally through a USB port provided at the station itself. Requirement for data analysis: the data collected by the sensor is processing in the station itself then send its payload with formatted values to the service in the "cloud". Deployment of applications: firmware or control software is within microcontroller’s flash memory inside the station, to monitor the data produced by the station an IoT platform with a web site dashboard. Security requirements: the system must have basic user authentication for change and access to the IoT platform, however, will be of public access for the visualization of measurements.

2.2. Process Specification

Define a single case of operation in a repetitive loop through the firmware in the digital controller: when the system boots, it executes actions to set up internal and external hardware of the microcontroller, then reads the sensors for PLG, CO and Temperature, active an on-board buzzer if reading are above thresholds, finally sends readings to the IoT service through the Wi-Fi network, this entire process is periodic. Figure 1 shows a pseudocode algorithm about described process.



```
Algorithm 1: Algorithm for IoT Station
  Result: periodically send gas concentration data to the IoT platform
  controller hardware initialization;
  WiFi transceiver hardware initialization;
  gas sensors initialization;
  while True do
      read sensors;
      if sensor readings above thresholds then
          active buzzer;
          send sensor data to IoT platform;
      else
          send sensor data to IoT platform;
      end
      wait t;
  end
```

Figure 1. Single case algorithm for electronic station process.

2.3. Domain Model Specification

Physical entity: the surrounding air, whose concentration of GLP and CO gases will be read. Virtual entity: represent the physical entity in the digital world. So, we define only one for the surrounding air. Device: programmable digital controller with connected LPG and CO gases sensors. Resource: firmware that runs on the device and a setup script that runs in the IoT cloud. Service - The station service runs natively on the device.

2.4. Functional View Specification

The functional view defines functional groups (FG) for the different functions of the IoT system. Each functional group provides functions to interact with instances of concepts defined in the domain model or information related to those concepts. Device FG: includes the programmable controller, PLG and CO gases sensors. Communications FG: protocols used are 802.11 link layer, IPv4 network layer, TCP transport layer, HTTP application layer; to send data payload to the IoT platform system use JSON format. Services FG: There is only one service running within the IoT station controlling service. Administration FG: is performs by the firmware resource. Security FG: security mechanism is a single user credential for IoT cloud configuration. Application FG: the user interface for monitoring the values produced by the IoT system is in the "cloud" as a web page.

2.5. Operational View Specification

The Options for deployment and operation of the IoT system are defined. IoT electronic station: mains components are a microcontroller, a Wi-Fi transceiver to internet access, a sensor for LPG gas, a sensor for CO gas. As a visual representation, a led screen for displaying text on the station and a Buzzer to play a sound alarm onsite. Communication API: Amazon Web Services API. Communication protocols: 802.11, IPV4 / 6, TCP and HTTP. Services: controller service hosted on the device written on C programing language and running as a native service. Applications: Web and database Application – AWS web toolbox. Administration: device – Arduino IDE for electronics station and AWS for cloud applications. Figure 2 shows functional blocks for the proposed IoT system.

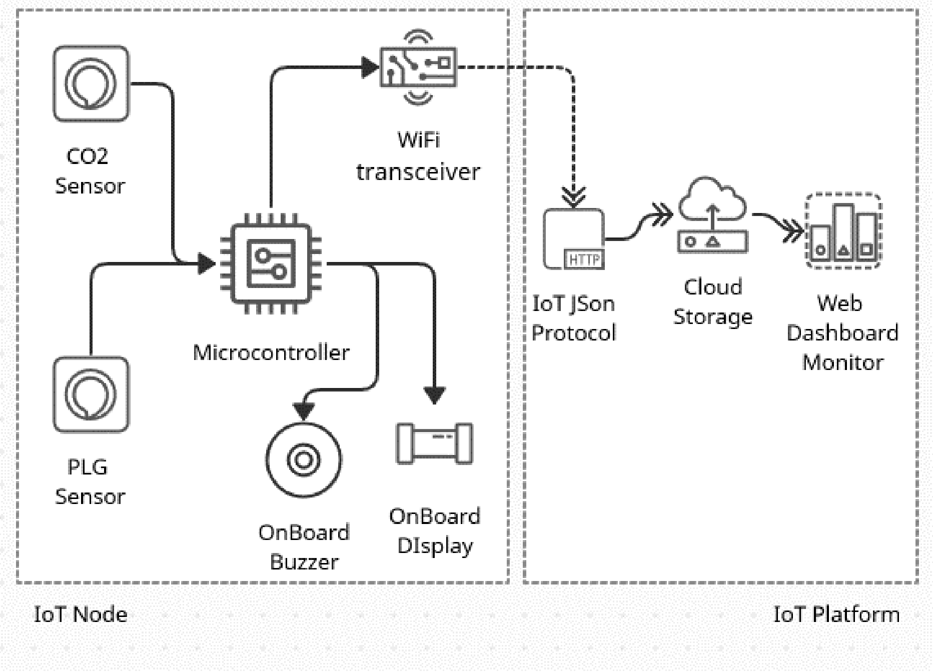


Figure 2: Overview of functional blocks architecture of the proposed IoT system.

2.6. Device & Component Integration

Figure 3 shows a schematic diagram with component integration of the IoT electronic station. Mains components used are an Arduino development board for Atmega2560 microcontroller, a ESP8266 chip as Wi-Fi transceiver, a SSD1306 Oled display, MQ-5 LPG sensor, a MQ-9 CO gas sensor and a sound buzzer. Sensors are connected via ADC pins, Wi-Fi transceiver used the UART port pins and I2C port pins are used to handle Oled display, buzzer use one GPIO of microcontroller.

2.7. Application Development

The Applications developed to run by the IoT the system are: 1) Device firmware: written in C programing language, the program follows a one loop structure and specific tasks: a. Read sensors values concentrations for PLG and CO gases. b. Store these values locally c. display readings on LCD on device d. Compare readings values to threshold levels of each gas, if current values are above sound buzzer on the station. e. Packet and send the data payload to IoT cloud. g. Wait until the next reading. 2) Service script in the "cloud": developed in JavaScript language hosted in Amazon Web Services (AWS) cloud. The telemetric protocol JavaScript Object Notation (JSON) is used to send and receive data between sensor station and IoT Platform. AWS services were selected for their low cost, high reliability, and availability versus other similar services. In addition to having a relatively short learning curve. 3) Web application: it was developed using the AWS hosting services, using web-toolbox we setup the web site with data tables and graphical dashboard to display the data generated by the sensor

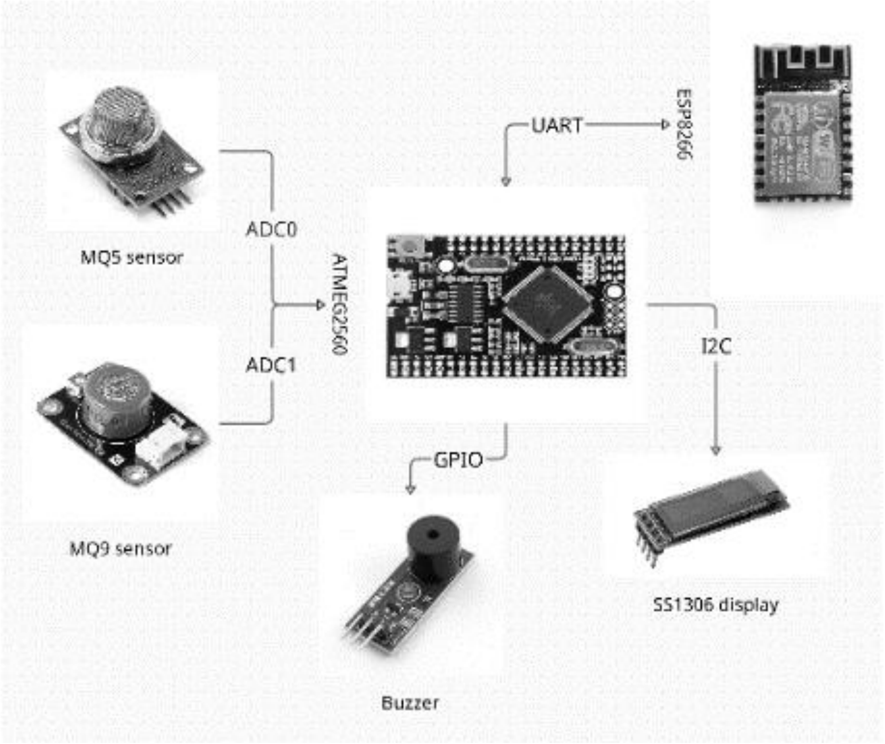


Figure 3: Electronics components integration for sensor station of proposed IoT system.

3. RESULTS AND DISCUSSION

3.1. IoT monitoring station

One main result of this work was an electronic sensor station prototype that allows take measurements of PLG and CO gases concentrations and send it to the IoT cloud platform, figure 4 shows some photos of the prototype. It is a design that takes on mind needs for a Salvadoran condition, were based on the state-of-the-art, affordable and efficient electronic components. The design of the system allows to add more sensors to the station to increase the gases to be measured. Station reports to the website two values of sensed magnitudes every 10 minutes or when the values raise above threshold levels. The 10 minutes' period is configurable via firmware on microcontroller. Among electrical characteristic of the station prototype, we have: Operation voltage: 110VAC, Current consumption: 0.4W Max, working temperature: +60°C Max. Measurement operation: range PLG and CO 1 to 1000 part per million (ppm) Communication operation: Link: 802.11 Wi-Fi Transmission power: 14 dB tip. Physical installation of station is simple, it can be embedded in a wall of a building or structure, with a height between 1.5 to 2 meters from the ground level. The technical requirements for the place of installation are: access to electric power and Wi-Fi network coverage, the stations are configured for network access by DHCP. The commissioning only requires defining, via the firmware, the network access credentials and the time between reports to the IoT platform, in the case of study it is used 10 minutes.

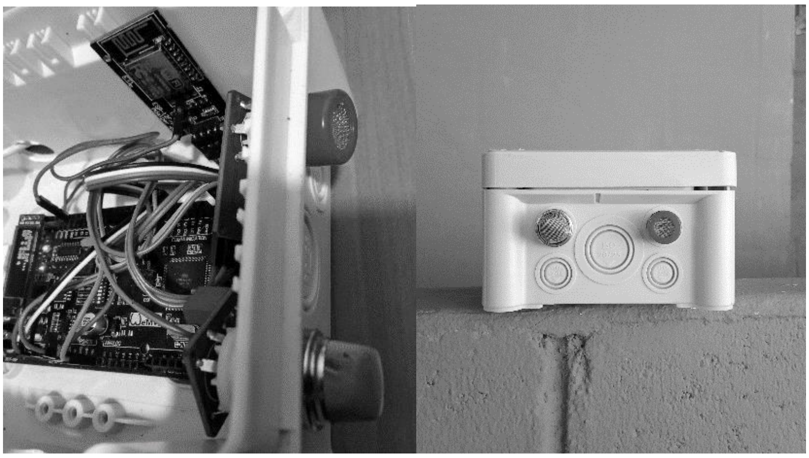
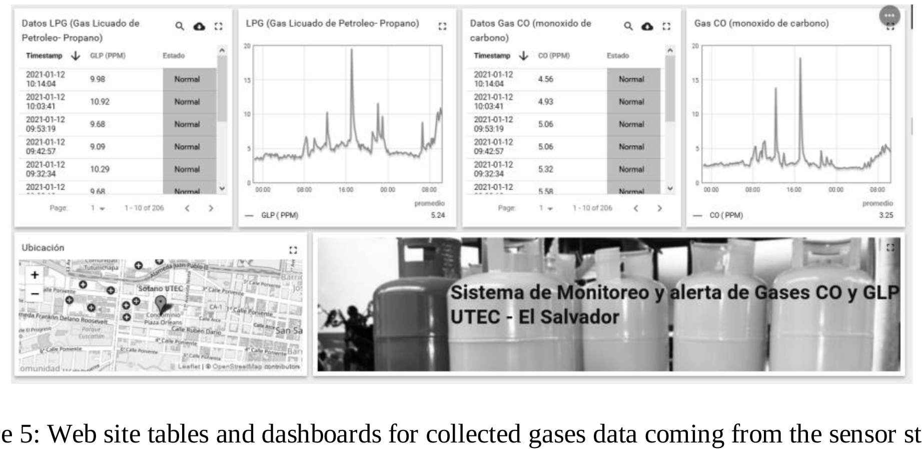


Figure 4: Assembled electronic prototype (left) and outdoors enclosure (right) for the IoT sensor station.

3.2. Web Site and Field Test

As a field test designed system was implemented with one node. The station was placed in the center of San Salvador, within the campus of the Universidad Tecnologica de El Salvador, a sector of the capital city with high traffic, specifically between 19 Avenida and Calle Arce. This point was selected to observe its performance and verify operation of the system and to make the necessary adjustments. To monitoring the data collected user can access through any device with Internet access to website with the URL: https://bit.ly/2JBBRkd. This website, figure 5, includes tables and dashboards to view the history of values for PLG and CO gases reported by the station. System performance so far has been satisfactory. The telemetry link has not suffered losses and has remained stable. The website has been available and has not been down. Regarding the data collected, a growth trend has been observed in the concentration of gases that coincides with the rush hours of automotive traffic. It is on early hours of the day and late at night when the tendency on gases concentration is low. Significant effects were observed in gas concentration values in response to climatic conditions such as rainy and windy seasons. The concentration values decrease to their lowest values during the weekdays or after a rain or a gust of wind. This is consistent with the assumption that vehicular traffic is a major source of air pollutant by PLG and CO gases.



Datos LPG (Gas Licuado de Petroleo- Propano)

Timestamp ↓	GLP (PPM)	Estado
2021-01-12 10:14:04	9.98	Normal
2021-01-12 10:03:41	10.92	Normal
2021-01-12 09:53:19	9.68	Normal
2021-01-12 09:42:57	9.09	Normal
2021-01-12 09:32:34	10.29	Normal
2021-01-12	9.68	Normal

Page: 1 ▾ 1 - 10 of 206 ‹ ›

Datos Gas CO (monoxido de carbono)

Timestamp ↓	CO (PPM)	Estado
2021-01-12 10:14:04	4.56	Normal
2021-01-12 10:03:41	4.93	Normal
2021-01-12 09:53:19	5.06	Normal
2021-01-12 09:42:57	5.06	Normal
2021-01-12 09:32:34	5.32	Normal
2021-01-12	5.58	Normal

Page: 1 ▾ 1 - 10 of 206 ‹ ›

Figure 5: Web site tables and dashboards for collected gases data coming from the sensor station.

4. Conclusion and Future Work

Development of an IoT station to monitor the level of LPG and CO gases concentration in surrounding air is a fundamental step for study of behaviour, impacts, and actions for care of the environment, reduction of expenses in health and focus resources on possible solutions for problems that affect quality of human life. The proposed system was developed using state-of-art techniques on electronics, programming, and internet of things, which allowed to produce a piece of low-cost equipment that works according to the expected requirements. Tools such as Atmega microcontroller together with C programming language allows development of efficient IoT prototypes at a low-cost, with short development times and high performance. In addition, using the ready-to-use tools of AWS has allowed fast and simple development of a platform and web site to data monitoring from any device and in real-time. The scientific contribution of this work was to show new and innovative techniques for the use of hardware and software components in the implementation of Internet systems of things. These can be applied in new developments, allowing for fast and efficient prototyping. In the future, this research has the task of developing more stations for different locations within the national territory, conducting validation experiments with additional sensors. In future, we seek to implement a monitoring network through radio frequency links, and analyse massive data or forecasts from the data produced by stations. Also, result of this work can be use in development of new lines of applied research, in areas such as: analysis of aquifers, monitoring in agriculture and livestock fields, analysis of sports performance, etc.

Acknowledgements

This work was supported by applied research office of Universidad Tecnologica de El Salvador.

References

[1] N. Di Sbroiavacca, «Rol y perspectivas del gas natural en la transformación energética de América Latina: aportes a la implementación del Observatorio Regional sobre Energías Sostenibles,» CEPAL, 2019.

[2] B. Yuliarto, M. Silvia, M. Iqbal and Nugraha, "Fabrication of LP Gas Leakage Detector Systems Based on Modified Nanostructured ZnO Thin Film", Advanced Materials Research, vol. 364, pp. 206-210, 2011. Available: 10.4028/www.scientific.net/amr.364.206.

[3] Amhsr.org, 2021. [Online]. Available: https://www.amhsr.org/articles/modelling-the-consequences-of-explosion-fire-and-gas-leakage-in-domestic-cylinders-containing-lpg.pdf. [Accessed: 08- Sep-2019].

[4] J. Chang and C. Lin, "A study of storage tank accidents", Journal of Loss Prevention in the Process Industries, vol. 19, no. 1, pp. 51-59, 2006. Available: 10.1016/j.jlp.2005.05.015.

[5] "Dirección de Hidrocarburos y Minas evalúa origen de la explosión en planta de Tropigas en Soyapango - Noticias de El Salvador y el Mundo", Noticias de El Salvador y el Mundo, 2021. [Online]. Available: https://diario.elmundo.sv/direccion-de-hidrocarburos-y-minas-evalua-origen-de-la-explosion-en-planta-de-tropigas-en-soyapango/. [Accessed: 10- Dec- 2020].

[6] "Dos empleados de empresa de gas detenidos como presuntos responsables de explosión en local del mercado de Santa Ana | Noticias de El Salvador - elsalvador.com", Noticias de El Salvador - elsalvador.com, 2021. [Online]. Available: https://www.elsalvador.com/noticias/nacional/santa-ana-explosion-mercado-municipal-empleados-empresa-de-gas-detenidos-pnc/792563/2021/. [Accessed: 05- Jan- 2021].

[7] "El Salvador: 3 muertos y 7 lesionados por explosión de gas", AP NEWS, 2021. [Online]. Available: https://apnews.com/article/c60a24c91dc14e52b44ff09cbf625225. [Accessed: 21- Jul- 2020].

[8] "Incendio en planta de Tropigas | Bomberos controló incendio y confinó fuego en un área, autoridades evacúan alrededores", Noticias de El Salvador - La Prensa Gráfica | Informate con la verdad, 2021. [Online]. Available: https://www.laprensagrafica.com/elsalvador/Reportan-incendio-

de-grandes-proporciones-y-explosiones-en-planta-de-Tropigas-20201205-0033.html. [Accessed: 05-Dec- 2020].

[9] S. Ragsdale, "Life with Carbon Monoxide", Critical Reviews in Biochemistry and Molecular Biology, vol. 39, no. 3, pp. 165-195, 2004. Available: 10.1080/10409230490496577.

[10] G. Reumuth et al., "Carbon monoxide intoxication: What we know", Burns, vol. 45, no. 3, pp. 526-530, 2019. Available: 10.1016/j.burns.2018.07.006.

[11] C. Henry, "Myocardial Injury and Long-term Mortality Following Moderate to Severe Carbon Monoxide Poisoning", JAMA, vol. 295, no. 4, p. 398, 2006. Available: 10.1001/jama.295.4.398.

[12] P. Sittisak, T. Charinpanitkul and B. Chalermsinsuwan, "Enhancement of carbon monoxide removal in an underground car park using ventilation system with single and twin jet fans", Tunnelling and Underground Space Technology, vol. 97, p. 103226, 2020. Available: 10.1016/j.tust.2019.103226

[13] N. Hampson, J. Holm and T. Courtney, "Garage carbon monoxide levels from sources commonly used in intentional poisoning", Undersea and Hyperbaric Medicine, pp. 11-15, 2017. Available: 10.22462/1.2.2017.3.

[14] "Morirse sin darse cuenta", EL PAÍS, 2021. [Online]. Available: https://elpais.com/elpais/2007/11/26/actualidad/1196068621_850215.html.

[15] S. Mundo, "Un asesino silencioso: el monóxido de carbono cobra vidas en América Latina", Mundo.sputniknews.com, 2021. [Online]. Available: https://mundo.sputniknews.com/america-latina/201806181079683679-muertes-co-intoxicacion-casos/.

[16] D. (www.dw.com), "Cinco muertos en Ecuador por inhalar monóxido de carbono en confinamiento | DW | 18.05.2020", DW.COM, 2021. [Online]. Available: https://www.dw.com/es/cinco-muertos-en-ecuador-por-inhalar-mon%C3%B3xido-de-carbono-en-confinamiento/a-53474709.

[17] "Ante la baja de las temperaturas, especialistas advierten sobre el riesgo de intoxicaciones con monóxido de carbono", infobae, 2021. [Online]. Available: https://www.infobae.com/salud/2020/05/14/ante-la-baja-de-las-temperaturas-especialistas-advierten-sobre-el-riesgo-de-intoxicaciones-con-monoxido-de-carbono/.

[18] "Los gráficos animados que muestran los 15 países que más CO2 emitieron en los últimos 20 años (y cuánto emitió América Latina) - BBC News Mundo", BBC News Mundo, 2021. [Online]. Available: https://www.bbc.com/mundo/noticias-internacional-50811389

[19] Selvapriya C, Sathya Prabha, Abdulrahim , Aarthi K C, “LPG Leakage Monitoring and Multilevel Alerting System”, international journal of en-gineering sciences & research Technology (IJSERT), [Selvapriya, 2(11): November, 2013]

[20] "Carbon monoxide Levels that Sound the Alarm | Kidde", Kidde, 2021. [Online]. Available: https://www.kidde.com/home-safety/en/us/support/help-center/browse-articles/articles/what_are_the_carbon_monoxide_levels_that_will_sound_the_alarm_.html.

[21] M. Shaw, "Carbon Monoxide (CO) - Gas Hazards & Workplace Safety", Cacgas.com.au, 2021. [Online]. Available: https://www.cacgas.com.au/blog/carbon-monoxide-co-toxic-gas-workplace-safety.

[22] R. Salmani, "LPG Gas Leakage Detection & Control System", Bonfring International Journal of Software Engineering and Soft Computing, vol. 6, no., pp. 73-77, 2016. Available: 10.9756/bijsesc.8246.

[23] "AUTOMATIC GAS BOOKING, LEAKAGE AND DETECTION USING GSM", International Journal of Recent Trends in Engineering and Research, vol. 3, no. 3, pp. 219-224, 2017. Available: 10.23883/ijrter.conf.20170331.043.qeunb.

[24] P. Spachos and D. Hatzinakos, "Real-Time Indoor Carbon Dioxide Monitoring Through Cognitive Wireless Sensor Networks", IEEE Sensors Journal, vol. 16, no. 2, pp. 506-514, 2016. Available: 10.1109/jsen.2015.2479647.

[25] R. Pitarma, G. Marques and B. Ferreira, "Monitoring Indoor Air Quality for Enhanced Occupational Health", Journal of Medical Systems, vol. 41, no. 2, 2016. Available: 10.1007/s10916-016-0667-2.

[26] W. Sung and Y. Hsu, "Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee", Expert Systems with Applications, vol. 38, no. 4, pp. 4522-4529, 2011. Available: 10.1016/j.eswa.2010.09.126.

[27] T. Wen, J. Jiang, C. Sun, J. Juang and T. Lin, "Monitoring Street-Level Spatial-Temporal Variations of Carbon Monoxide in Urban Settings Using a Wireless Sensor Network (WSN) Framework", International Journal of Environmental Research and Public Health, vol. 10, no. 12, pp. 6380-6396, 2013. Available: 10.3390/ijerph10126380.

[28] K. S. S. Ram and A. Gupta, “IoT based Data Logger System for weather monitoring using Wireless sensor networks,” International Journal of Engineering Trends and Technology, vol. 32, no. 2, pp. 71–75, 2016.

[29] C. Xiaojun, L. Xianpeng, and X. Peng, “IOT-based air pollution monitoring and forecasting system,” 2015 International Conference on Computer and Computational Sciences (ICCCS), 2015.

[30] S. Ravichandran, "Cloud Connected Smart Gas Cylinder Platform Senses LPG Gas Leakage Using IOT Application", International Journal of MC Square Scientific Research, vol. 9, no. 1, pp. 324-330, 2017. Available: 10.20894/ijmsr.117.009.001.038.

[31] A. Karumbaya and G. Satheesh, "IoT Empowered Real Time Environment Monitoring System", International Journal of Computer Applications, vol. 129, no. 5, pp. 30-32, 2015. Available: 10.5120/ijca2015906917.

[32] M. Shahadat, A. Mallik and M. Islam, "Development of an automated gas-leakage monitoring system with feedback and feedforward control by utilizing IoT", Facta universitatis - series: Electronics and Energetics, vol. 32, no. 4, pp. 615-631, 2019. Available: 10.2298/fuee1904615s.

[33] S. Kristiyana and A. Rinaldi, "Air Quality Monitoring System in Thingspeak-Based Applications Using Internet of Things (IOT)", WSEAS TRANSACTIONS ON COMPUTER RESEARCH, vol. 8, pp. 34-38, 2020. Available: 10.37394/232018.2020.8.6.

[34] A. Mari and A. Raghunath, "Air Quality Monitoring System using Raspberry Pi and Web Socket", International Journal of Computer Applications, vol. 169, no. 11, pp. 28-30, 2017. Available: 10.5120/ijca2017914826.

[35] C. Balasubramaniyan and D. Manivannan, "IoT Enabled Air Quality Monitoring System (AQMS) using Raspberry Pi", Indian Journal of Science and Technology, vol. 9, no. 39, 2016. Available: 10.17485/ijst/2016/v9i39/90414.

[36] Bahga, A., & Madisetti, V. (2014). Internet of Things: A Hands-On Approach (Edition: 1). S.l.: Vpt

AUTHORS

Otoniel Flores-Cortez received his Engineer degree in electrical engineering from the Universidad de El Salvador, San Salvador, El Salvador, in 2005 Since 2005 he works as a lecture and researcher for the electronics department in the Universidad Tecnológica de El Salvador, San Salvador, El Salvador. His research interests include embedded systems, IoT systems.

Ronny Cortez received his engineering degree in computer science from the Universidad Tecnologica de El Salvador in 2012. Since 2013 is working as a associate professor and research in the computer science department in the same university. His research interests are data mining, cloud computing and machine learning.

Bruno Gonzalez received his engineering degree in computer science from the Universidad de El Salvador in 2012. Since 2011 is working as a professional developer, he worked on different industries like airlines, and services. His research interests are the application of IoT technologies and Machine Learning techniques for natural language processing and computer vision.

STUDYING THE APPLICABILITY OF PROOF OF REPUTATION(POR) AS AN ALTERNATIVE CONSENSUS MECHANISM FOR DISTRIBUTED LEDGER SYSTEMS

Oladotun Aluko[1] and Anton Kolonin[2]

[1]Novosibirsk State University, Novosibirsk, Russia
[2]Aigents Group, Novosibirsk, Russia

ABSTRACT

Blockchains combine several other technologies like cryptography, networking, and incentive mechanisms in order to support the creation, validation, and recording of transactions between participating nodes. A blockchain system relies on a consensus algorithm to determine the shared state among distributed nodes. An important component underlying any blockchain-based system is its consensus mechanism, which determines the characteristics of the overall system. This thesis proposes a reputation-based consensus mechanism for blockchain-based systems which we term Proof-of-Reputation(PoR) that uses the liquid rank algorithm where the reputation of a node is calculated by blending the normalized ratings by other nodes in the network for a given period with the reputation values of the nodes giving the ratings. The nodes with the highest reputation values eventually become part of the consensus group that determines the state of the blockchain.

KEYWORDS

Consensus, Distributed Ledger Technology, Blockchain, Reputation, Social Computing.

1. INTRODUCTION

In the last couple of years, blockchain has received a significant amount of attention from the industry and academia alike and quite rightly so due to the success of cryptocurren- cies. While cryptocurrencies are the most popular use case for blockchain technology, there is a plethora of application domains. A blockchain system is, fundamentally, a distributed system that relies on a consensus algorithm to determine shared state among distributed nodes. In blockchain speak, this shared state called a chain is a public or private record of all transactions or digital events that have been created and shared among participating nodes. A blockchain system is, fundamentally, a distributed system that relies on a consensus algorithm to determine shared state among distributed nodes. In blockchain speak, this shared state called a chain is a public or private record of all transactions or digital events that have been created and shared among participating nodes [1].

The main objective of any blockchain-based system is to maintain a live decentralized transaction ledger while defending against attacks from malicious Byzantine actors that may try to game the system. The reliability and the integrity of the entire blockchain system as a whole depend largely on the consensus model employed. The applicability of any consensus mechanism is based on three key properties: safety, liveness and fault tolerance [2].

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 41-53, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110804

Distributed consensus among nodes geographically distributed has been a widely studied research topic in distributed systems, however, with the advent of blockchain, it has received more attention as blockchains are a type of distributed system. Most blockchain-based systems targeting different application domains with an array of unique requirements have introduced a corresponding consensus mechanism suited for its particular uses. As a result of this, several consensus algorithms have emerged with different properties and capabilities.

The common approach to building consensus among participants is evidence-based. The basic concept of a proof-based consensus algorithm is that among many nodes joining the network, the node that performs sufficient testing is given the right to add a new block to the chain and gets a reward [3]. A large number of these methods are still vulnerable to the games of the participants on the network. The most popular proof-based algorithm is the PoW (Proof of Work) consensus algorithm, which powers the Bitcoin cryptocurrency, where each participant in the system votes by the total amount of computing power that the participant controls at the time of the vote. The obvious disadvantage of this approach is that anyone with the most computing power can essentially take over a significant portion of the system. In addition, it has been known to consume a significant amount of resources. a lot of electricity [4]. Proof of Stake (PoS) is an energy-saving alternative to PoW. PoS requires nodes to demonstrate ownership of a particular stake as it is believed that nodes with more coins are less likely to attack the network.

As the nature of peer-to-peer (P2P) networks is open and dynamic, the security risk within that environment is greatly increased mostly because nodes can join and leave the network at will. Thus, it is important to have a system that can check against malicious behaviour. One way to minimize risks associated with this type of open communities is to use community-based reputations. Historically, reputation systems have been employed to facilitate trust between entities [5]. The reputation of a node defines an expectation about its behavior, which is based on other nodes’ observations or information about the node’s past behavior within a specific context at a given time.

The Proof of Reputation (PoR) consensus algorithm is about decentralizing reputation such that the reputation dynamics of each member in the system can be measured and tracked [6]. This consensus algorithm can be seen as the application of a reputation systems model to the blockchain. PoR adopts the concept of the balance of power and designs a decentralized incentive system that ensures that new and existing users have the same opportunity to receive rewards from the system. PoR can prevent the distribution of power from being centralized due to incomplete incentive designs. It is particularly useful in the design of social decentralized applications.

The main contributions of this work are described as follows:

- First, in our reputation-based consensus mechanism, the reputation of a node is not simply calculated by the value of the direct rating given by other nodes but by blending together a normalized set of ratings and the corresponding reputation values of the node providing the rating at a given period in time. The behaviour of a node affects its overall reputation value
- Second, our reputation-based consensus mechanism is based on the following principles: 1) The liquid nature of the reputation values. The reputation value computed for a node is based on the reputation value of the node providing the rating. 2) The temporal scoping of reputation so that reputation values collected by members in the past are less contributing to the current reputation value. 3) The openness of all reputation values to all members in the community so that audits can be performed.

- Third, we use a side chain to store the reputation values of all the nodes without the use of a third party to manage reputation
- Finally, we develop an experimental implementation and evaluate its performance in terms of security and the throughput of the system.

2. REVIEW OF RELATED WORK

2.1. Consensus Algorithms

Blockchains solve the Byzantine Generals Problem [7] which is a problem associated with distributed systems. This problem is addressed with the method of verifying the transactions by many distributed nodes. The data can be delivered between different nodes usually through a broadcast message. However, some nodes may be maliciously attacked, which could lead to a situation where changes are made to communication contents. Every node in the network needs to distinguish the information that has been tampered and obtain the consistent results with other normal nodes. This is usually done through a consensus algorithm.

Consensus is central to the Blockchain Technology [8], [9]. It has been studied for well over three decades. Consensus protocols have been historically known to enable consensus to be reached about a shared among a set of distributed nodes. The design of a consensus protocol is a challenging task and so it is usually common to make assumptions under which the protocol is proven to function properly. These assumptions eventually influence the characteristics of the consensus protocol. In fact, it's the case that most applications of the Blockchain Technology usually roll out their corresponding consensus algorithm to fit the specific use case for the Technology [10], [11], [12].

Proof-of-Work (PoW) is by far the most widely used consensus mechanism for blockchains introduced by Bitcoin [2], [13]. In PoW nodes acting as miners vote by the amount of computing power they possess by trying to solve a computational challenge. The first node to solve the challenge validates and adds a new block of transactions to the blockchain and gets a reward for this action. However, the generation of blocks requires the use of a huge amount of computational power and introduces delay for block confirmation, resulting in low efficiency and low transaction throughput.

Proof-of-Stake (PoS) was proposed as an alternative to PoW. With PoS, nodes who like to participate in the block creation process must prove ownership of a certain amount of coins. In addition, they are required to lock a certain amount of currency, called stake, to participate in the block creation process. A variation of PoS is the Delegated-Proof-of-Stake(DPoS), in which miners are elected by other nodes. The stake of the nodes are used as the weighting parameter for votes.

2.2. Reputation-Based Consensus Algorithms

Reputation has been defined as a quantity derived from the underlying social network which is globally visible to all members of the network [14], [15], [16]. Reputation systems have historically been known as a means of harnessing reputation data in some form. They work by facilitating the collection, aggregation and distribution of data about an entity. This data can thereafter be used to characterize and predict that entity's future actions [17], [18]. Essentially, by referring to the reputation data, users within a network are able to decide whom they will trust, and to what degree. In addition to above, a reputation system is a socially corrective mechanism, as the incentive of positive reputation and the disincentive of negative reputation will generally encourage good behavior over the longer term. Upon the collection of reputation data by a reputation system, it can be shared amongst users which in turn can be used to evaluate other users before making decisions about intended or future interactions, without ever having to have previously interacted. Examples of the practical application of this system can be found on eCommerce websites like Amazon or eBay where reputation attributed to a seller is influenced by ratings through previous transactions. Another use case is in government where a country like China incentives the behavior of the citizens through a social credit score system. Earlier works proposed possibilities of applying a reputation-based model to distributed computing. One such was described by [19]. The downside was that the approach was not completely decentralized.

Recent studies have introduced reputation systems into the blockchain space to improve efficiency and reliability. [20] proposed a reputation-based consensus mechanism for peer- to-peer networks where reputation serves as the incentive for good behaviour and the node with the highest reputation gets to publish a new block. At the end of each interaction between two nodes, feedback is generated by the service requester and broadcast to the entire network. On reaching the set threshold, nodes start to calculate a ranking list after which the node with the highest ranking publishes the new block and other nodes verify the integrity of the newly published block. [21] also proposed a reputation-based consensus mechanism based on the proof-of-work consensus algorithm. In their approach, a miner's voting power is given by its reputation. The reputation for each miner is computed based on the total amount of valid work a miner has contributed and also the regularity of that contribution over a given period. [22] proposed the Blockchain Reputation-Based Consensus(BRBC) mechanism in which a node in the network must have a reputation score higher than a set threshold to be able to publish a new block. Also, a judge is randomly selected that is responsible for updating node reputation values. However, none of these address the behaviour of a node on a transactional basis as it interacts with other nodes in the network.

3. SYSTEM OVERVIEW AND THREAT MODEL

In our approach, we consider that the network is untrustworthy and unreliable, which means messages in the network can be delayed, duplicated or lost in some cases. Furthermore, the nodes are heterogeneous and failure at a node does not cause the failure of another node.

We also assume that there's a social community that affords nodes the ability to vote about different aspects of the system. Each node i is identified by a public key pki similar to a wallet in regular blockchains like Bitcoin. This public key has a corresponding secret key ski with which it can use to append its signature to transactions. Each transaction group is made up of two nodes [i, j]. Node j gives the rating while node i is the recipient of this rating usually with respect to an interaction between them. The transaction is said to be completed only after it is appended to the blockchain.

During the consensus phase, a node can be a leader of the consensus group or simply a member of the consensus group. For smaller networks, all the nodes in the network can be part of the consensus group. For larger networks, it's impractical to have all nodes as members of the consensus group. In those instances, a subset of nodes within the network that have the highest reputation corresponding to at least two-thirds of the entire reputation values in the network should be used. The leader for the consensus round can then be selected at random.

In addition, we assume that there is a malicious node within the network that may cause the failure or misbehaviour of a number of nodes. In order for the system to be safe and live, we assume that if F nodes eventually become faulty, at least 3F + 1 nodes remain honest.

4. The Consensus Mechanism

4.1. Consensus Group

In our scheme, we assume *N* nodes in the network, an individual node is represented as p_i, $i \in N$. The computation performed by network nodes happens in rounds during which a node sends messages, receives them and thereafter performs some local computation on the received message [23]. Each node *i* is identified by a key pair, pk_i which is the public key and sk_i is the corresponding secret key. At the end of every interaction between nodes, rating values are generated with respect to the service. A node will usually function in one of two possibilities: either as the recipient node or as the rater node for the particular interaction. Whenever a rater node gives a rating, it broadcasts the details of that transaction to the entire network. We denote this interaction where rater node is *i* and node *j* is the recipient node as follows:

$$Transaction = (E_{sk_i}, pk_j, r) \qquad (1)$$

$$\{r : 0 < r < 1\}$$

where pkj is the public key of the recipient node, r is the rating given by the rater node which is a value between 0 and 1, and Eski is the encrypted transaction data signed using the rater node's secret key. Transactions generated between nodes for a round k are added to a list of pending transactions waiting to be appended to the chain during the consensus phase.

At the start of every consensus round, consensus group members need to be selected and added into a consensus group. We denote this consensus group for a round k as Gk. The consensus group members are selected from the nodes with the highest reputation values for which their collective reputation scores are over 50% of the total reputation values of the entire network. With this approach, the size of Gk will vary depending on the reputation distribution in the network. A node that is part of the consensus group is denoted as:

$$pk_i \in G_k \qquad (2)$$

To proceed, a new leader L_k for the round *k* is selected. After the leader for the round *k* is selected, it serves the following functions:

- Packaging all valid transactions from the list of pending transactions to a $Block_k$
- Calculating the new reputation values for all network nodes for $Reputation_k$ the round *k* using data from transactions in the transaction list
- Broadcasting the commit message to the consensus group G_k

4.2. Leader Selection

We use a random function to select the leader for the consensus group for the round G_k. By doing this, there is no deterministic guarantee for which node will be selected as leader for the consensus round *k*. As such, all nodes that have been selected as members of the consensus group for the round *k* have equal chance of being selected. The leader's public key is broadcasted to the consensus group before the start of the consensus phase. The leader L_k packages all the transactions T_i for a recent time window in a specific order and adds them into a block. Afterwards, the leader sends a commit message to the consensus group G_k. This message contains the newly packaged *Block*$_k$, the leader's public key *pk*$_L$, reputation list for the round *k*, and also a hash of the *Block*$_k$ generated using the leader's secret key:

$$< Block_k, pk_L, Hash(Block_k), ReputationList > \quad (3)$$

4.3. Block Publication

After a commit message is sent by the leader L_k to the consensus group G_k, the consensus group determines the final block to be published for the current round *k*. Each node $pk_i \in G_k$ checks the commit message sent by the leader L_k which contains both the *Block*$_k$ and *Hash(Block*$_k$*)*. First, the node checks the pkL in the commit message to see if it matches the pkL that was broadcasted upon the leader selection earlier; otherwise, it can ignore the commit message. It then proceeds to check the integrity of the hash using the pkL. Afterwards, it checks the validity of the transactions within the *Block*$_k$. If this process completes, it sends a new commit message back to the consensus group G_k. This process continues with every node in the consensus group G_k. Upon completion, each node pk_i that successfully verifies the *Block*$_k$ and the *ReputationList* sends a verification commit back to the consensus group G_k.

$$< VERIFY, Block_k, Hash(Block_k), ReputationList > \quad (4)$$

The consensus group waits until at least a certain amount of consensus members sends in this message. This message constitutes a consensus group vote. We formalize this using a social choice function. For a set of nodes in the consensus group for round G_k, each node has an associated weight *w* assigned which is equivalent to its reputation value from the previous round $k - 1$. During the consensus process, there's a minimum quota which has to be reached for decisions to be made. We set this quota at two-thirds of the total weight in the consensus group:

$$d(G_k) = \begin{cases} 1 & \sum_{i \in G_k} w_i > q \\ 0 & \text{otherwise} \end{cases}$$

where *d(G*$_k$*)* represents the decision of the consensus group. Whenever the *d(G*$_k$*)* is 1, it means consensus for round *k* has been reached.

5. Reputation System

A node's reputation is defined by an evaluation of the ratings it receives from others in the past. These ratings reflect the degree of trust that other nodes have on a specific node based on their past interactions. Reputation-based systems generally rely on feedback to evaluate a node. This feedback is generally in terms of the amount of satisfaction a node receives by interacting with another node in the network. [24] pointed out that when considering reputation information, the source of information and the context need to be accounted for. We define the reputation principles for approach adapted from [25] as follows:

- The liquid nature of the reputation values. The reputation value computed for a node is based on the reputation value of the node providing the rating.
- The temporal scoping of reputation so that reputation values collected by members in the past are less contributing to the current reputation value.

- The openness of all reputation values to all members in the community so that audits can be performed.

Let s_i denote the reputation for a recipient node *i*. All nodes in the network start with a default reputation value determined on system initialization. During node interactions, a node's reputation value is determined by the liquid rank algorithm [6]. This approach can be used as a predictive metric to evaluate a node's behaviour. For each round, a node can receive multiple unique ratings:

$$s_{i1\ldots n} = \{s_{i1}, \ldots, s_{in}\} \tag{5}$$

where the range of s_i is *[0, 1]*. Values s_i are then normalised as follows:

$$S_{i,n} = \frac{S_{i,n} - min_i(S_{i,n})}{max_i(S_{i,n}) - min_i(S_{i,n})} \tag{6}$$

We slightly modify the normalization of the rating values to prevent null values from the set of ratings as follows:

$$S_{i,n} = \frac{(S_{i,n} - min_i(S_{i,n})) + 1}{(max_i(S_{i,n}) - min_i(S_{i,n})) + 1} \tag{7}$$

Furthermore, we define the ratings matrix *S* to be $[s_{ij}]$. After each round, these ratings will be generated for all nodes in the network. To compute new reputation values for a node for the round *k*, we blend these ratings with the rater reputation values from the previous round $k - 1$. We denote this as:

$$P = \overrightarrow{S} * \overrightarrow{R} \tag{8}$$

where $\overrightarrow{S} = [s_{ij}]$ and $\overrightarrow{R} = [r_{in}]$. r_{in} corresponds to the rater node providing the rating.

To compute the reputation value for the round *k*, we then blend the initial node's reputation value with the current rank generated from the ratings.

$$R_{i,k+1} = \alpha * P + (\alpha - 1) * R_k \tag{9}$$

where α is a constant determined on system initialization. The value is set between 0 and 1. It determines what portion of the equation to give more priority to. If the value is set closer to 1, it means that the newly generated reputation value will give more priority to the ratings *P* and less priority to the previously generated reputation value. This is what we want as this aligns with the reputation principles stated earlier. It helps to reduce the impact of nodes that change behaviour over time. Further, to prevent reputation values from hopping, we clamp the values using a sigmoid function as follows:

$$R'_{i,k+1} = \frac{R_{i,k+1}}{\sqrt{1 + (R_{i,k+1})^2}} \tag{10}$$

6. REPUTATION STORAGE

Most existing reputation mechanisms use a central server for the storage, management and sometimes distribution of reputation values among network nodes. In our approach, we do not require a central server but the reputation value for each node in the network is managed through a reputation side chain connected to the main transaction chain. The structure of the reputation chain is such that it has a header which contains meta information about a specific block and then the reputation values for all the network nodes:

$$ReputationBlock_i = (ReputationBlockHeader_i, ReputationList_i) \tag{11}$$

The *ReputationList*$_i$ contains a list of all network nodes with their associated reputation values for the most recent round. This serves as a lookup data structure for future uses. The *ReputationBlock*$_i$ as well as the transaction block use the standard blockchain block structure with a hash of all the transactions for the round, a previous hash, timestamp and transactions.

A new reputation block is created along with a normal transaction block during the consensus phase. So for a consensus round k, a *Block*$_k$ which is added to the transaction chain corresponds to a *ReputationBlock*$_k$ which is added to the reputation chain. As stated in section 4, part of the duties of the consensus group is to validate the reputation calculation generated by the leader L_k. After the consensus is reached, the leader broadcasts the new *ReputationBlock*$_k$ to the entire network and as such the reputation value of all the nodes in the network is visible to all other nodes.

7. EXPERIMENTS AND RESULTS

For our prototypes, we built an experimental protocol that implements the protocol. Thereafter, the nodes were deployed on AWS EC2 remote server running on 16GB RAM with Amazon’s t3 processor. In the experiment, we set up 1,000 nodes. We used a default initial reputation value of 0.2. We set the value of α to 0.6, the effect of that is that we give priority to recently generated reputation values.

To simulate the effect of a Wide Area Network, we impose a round trip latency of 200ms. While it’s unlikely that this will be the case in reality, the average of network delays across the entire network will average out to a close enough value.

In terms of the throughput, Figure 1 shows how the throughput values change as we vary the number of network nodes from 500 to 1,000 nodes. As the network size increases, so does the throughput because as more nodes join the network, more messages are being transferred in the network.

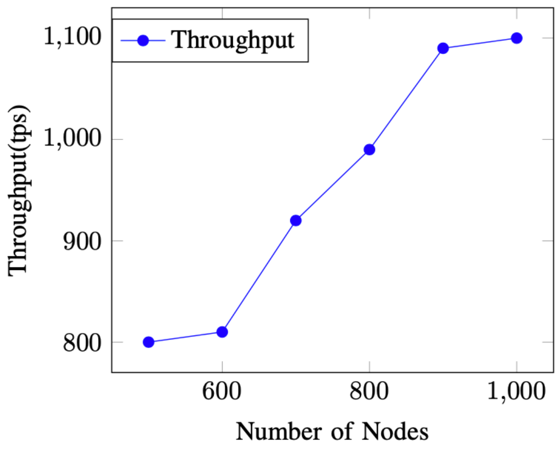


Figure 1: Throughput vs Number of network nodes

Also, we vary the number of transactions in a single block, we vary them between 100 and 500 to measure the average time it takes for a new block to be produced. Figure 2 shows that it takes more time for a new block to be produced as the transactions in a block increase. This is so because more transactions are now in a single block and so it takes more time for those transactions to be processed.

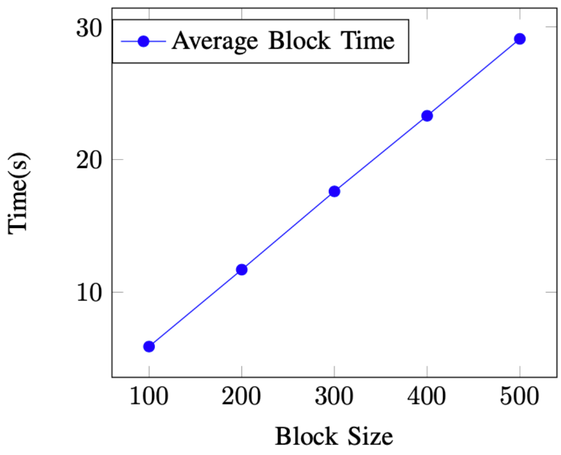


Figure 2: Average Block Time as the number of transactions in a single Block is varied

In our final experiment, we measure the consensus time in relation to the block size for each block. We observed that when the block size is relatively small around 100 transactions in each block, consensus takes about 2 seconds. As we increase the block size, so does the consensus time increase as well.

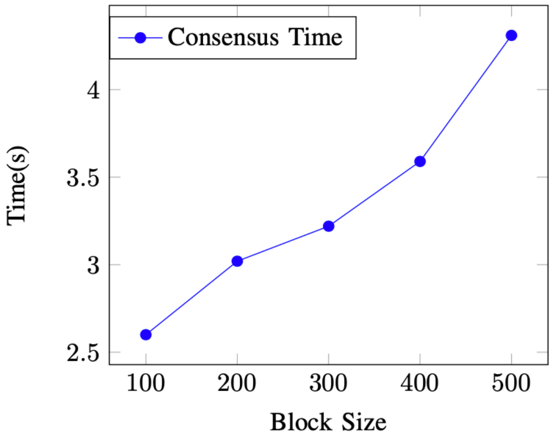


Figure 3: Consensus Time as the number of transactions in a single Block is varied

8. Security Analysis

In our reputation consensus mechanism, the right to generate a new block is reserved to the leader *Lk* for a round and the consensus group members for that round, in each round, only the highest ranking nodes are selected to be members of the consensus group. Unlike PoW where an adversary can attack the system only by having sufficient compute power. In addition, the leader election is based on random selection and there's no way for an adversary to deterministically predict the outcome of the selection process. As it takes time for reputation values to grow, an adversary will need to spend a lot of time doing honest work before it can be added as a consensus group member. For cases where an adversary becomes the leader for a round, all consensus group members still need to vote as regards the Block that will eventually be appended to the chain. Only when an adversary controls a significant number of members in the consensus group, at least two-thirds can the security of our approach be tampered with.

8.1. Selfish Mining Attacks

Selfish Mining attacks [26], [27], [28] is a mining strategy where a group of miners collude to exert power over the entire blockchain in order to increase their revenue. In selfish mining attacks, two groups exist side-by-side: an honest group of miners following the standard protocol and a colluding group that follows the selfish mining strategy. The selfish miners mine blocks while keeping them secret, they continue this process until the fork created from the main chain is longer than the main chain. In our approach, since blocks are not mined based solely on the compute power a node possesses or a group of nodes collectively possess, this kind of attack is impossible. Furthermore, there is no way for a node to know the nodes that will be involved in the consensus for a round or which node will be selected as the leader for that round.

8.2. Eclipse Attack

An eclipse attack [29] happens whenever a node in the network is occluded from the rest of the network. Most of the external contact for that node is controlled by the malicious node that launched the attack. This attack is a serious threat to any blockchain. In the case of our approach, the effect of this type of attack is only noticeable if an attacker is able to simultaneously isolate multiple consensus group members which is highly unlikely.

8.3. Flash Attacks

Flash attacks [30] happen whenever an attacker can pur- chase or rent compute power for a short period with the intention of using this compute power to its advantage. This type of attack is only feasible with network types like PoW that require the use of compute power. In our approach, an attacker with a sufficiently large amount of compute power cannot simply launch an attack on the basis of its compute power.

9. CONCLUSIONS

In this work, we proposed a reputation-based consensus mechanism for distributed ledger systems. The consensus scheme uses a social choice function where the weight of nodes that are responsible for consensus is equivalent to the reputation value for that node. In addition, approach uses the liquid rank algorithm where the reputation of a node is calculated by blending the normalized ratings by other nodes in the network for a given period with the reputation values of the nodes giving the ratings. Finally, we built an experimental prototype to show the potential of this approach. We remark that there are several other parts of this system which can be improved and are left to future work.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support and generosity of The Stream Data Analytics and Artificial Intelligence of the Novosibirsk State University without which the present research could not have been carried out.

REFERENCES

[1] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanara- man. (2016) Blockchain technology: Beyond bitcoin. [On- line]. Available: https://j2-capital.com/wp-content/uploads/2017/11/AIR- 2016-Blockchain.pdf?forcedefault=true.

[2] A. Baliga, "Understanding blockchain consensus models," 2017.

[3] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *J. Inf. Process. Syst.*, vol. 14, pp. 101–128, 2018.

[4] Quantaloop.io. (2020) Types of consensus algorithms in blockchain. [Online]. Available: https://quantaloop.io/proof-of-work-vs-proof-of-stake-101.

[5] F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *Journal of Parallel and Distributed Computing*, vol. 75, pp. 184–197, 2015.

[6] A. Kolonin, B. Goertzel, D. Duong, and M. Ikle, "A reputation system for artificial societies," *arXiv preprint arXiv:1806.07342*, 2018.

[7] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 203–226.

[8] V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Generation Computer Systems*, vol. 107, pp. 760–769, 2020.

[9] S. Azouvi, P. McCorry, and S. Meiklejohn, "Betting on blockchain consensus with fantomette," *arXiv preprint arXiv:1805.06786*, 2018.

[10] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensuses algorithms: A survey," *arXiv preprint arXiv:2001.07091*, 2020.

[11] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 1545–1550.

[12] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[13] A. Gervais, G. O. Karame, K. Wu ̈st, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.

[14] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[15] L. Kleinrock, R. Ostrovsky, and V. Zikas, "Proof-of-reputation blockchain with nakamoto fallback," in *International Conference on Cryptology in India*. Springer, 2020, pp. 16–38.

[16] J. Horton and J. Golden,"Reputation Inflation An Online Marketplace," *New York I*, vol. 1, 2015.

[17] G. Swamynathan, K. C. Almeroth, and B. Y. Zhao, "The design of a reliable reputation system," *Electronic Commerce Research*, vol. 10, no. 3, pp. 239–270, 2010.

[18] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, pp. 1–31, 2009.

[19] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer- to-peer networks," in *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, 2003, pp. 144–152.

[20] F. Gai, B. Wang, W. Deng, and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 666–681.

[21] J. Yu, D. Kozhaya, J. Decouchant, and P. Esteves-Verissimo,"Repucoin: Your reputation is your power," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1225–1237, 2019.

[22] M. T. de Oliveira, L. H. Reis, D. S. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. Mattos, "Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications," *Computer Networks*, vol. 179, p. 107367, 2020.

[23] P. Berman, J. A. Garay, K. J. Perry *et al.*, "Towards optimal distributed consensus," in *FOCS*, vol. 89. Citeseer, 1989, pp. 410–415.

[24] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[25] A. Kolonin and S. SingularityNET, "Reputation systems for human- computer environments," *Complexity, Informatics and Cybernetics*, 2019.

[26] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

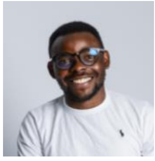
[27] C. Grunspan and R. Pe ́rez-Marco, "On profitability of selfish mining," *arXiv preprint arXiv:1805.08281*, 2018.

[28] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in *International Conference on Financial Cryptography and Data Security*. Springer, 2020, pp. 61–78.

[29] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 2015, pp. 129–144.

[30] J. Bonneau, "Why Buy When You Can Rent?" in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 19–26.

AUTHORS



Oladotun Aluko received his BSc(2017) in Computer Science and Engineering from Obafemi Awolowo University, Nigeria. He is currently working on his MSc in Big Data Analytics and Artificial Intelligence at Novosibirsk State University, Novosibirsk, Russia. His research interests Distributed Computing, Blockchain Technology, Machine Learning and Cloud Databases.

Anton Kolonin received his Ph.D in 1998 after he independently developed a software-algorithmic complex for processing geophysical data, introduced into production in many CIS countries. He has also participated as a leader or lead architect in many projects to develop algorithms and software, including those related to the use of AI, including the recognition of static text, moving objects, music, extracting information from texts and identifying events on financial markets – in Russian and foreign companies. Since 2017, he is also a software architect for AI and blockchain in the Singularity NET project, leading work care of unsupervised language learning and reputation systems.

Blockchain Decentralized Voting for Verified Users with a Focus on Anonymity

Piotr Pospiech, Aleksander Marianski and Michal Kedziora

Department of Computer Science and Management, Wroclaw University of Science and Technology, Wroclaw, Poland

Abstract

The paper presents decentralized voting scheme for verified users while maintaining their anonymity. A blockchain network was applied, which is a decentralized and distributed database based on the Peer-to-Peer architecture. During the implementation, the Ethereum network was used. Thanks to this, it is possible to code the terms of the contract required to perform the transaction. Ethereum and the use of smart contracts were also discussed in paper. The implementation uses the blind signature protocol by David Chaum and encryption with the Rivest-Shamir-Adleman (RSA) algorithm. Presented in this paper scheme for blockchain decentralized voting for verified users with focus on anonymity is then fully implemented and identified potential issues are analysed and discussed.

Keywords

Blockchain, e-voting, Ethereum.

1. Introduction

With the development of blockchain technology, more decentralized systems and applications began to emerge. Bitcoin, as the first project to gain global popularity, introduced a Peer-to-Peer payment system where every user is equally important and has the same rights [8]. The lack of a transfer authorization unit increases the credibility of the system for users [9]. Nowadays, commonly used electronic storage and payment systems are those provided by banks. There is a central unit here which is self-interested and has unlimited power in the system. As a result, it can impose additional restrictions or new account rules on users who, due to the lack of alternatives, have to agree to them. An additional problem is the lack of trust in institutions. Central unit can lead to fraud [1].

User anonymity and the lack of a central unit to manage the system have gained popularity over time [13]. More blockchain-based systems have emerged that offer a variety of possibilities [15][18]. Bitcoin is no longer attractive for transactions due to high transaction fees and speed of transactions. In addition, methods for analyzing the network in search of the owner of the transaction were found, which enabled user identification [2]. The use of blockchain systems for payments is not the only possible use [20]. Many decentralized applications have been created that use the advantages of blockchain networks and get rid of the disadvantages of centralized systems [19]. The development of such applications made it possible to introduce smart contracts, which were used for the first time in the Ethereum network [16]. Decentralized applications are used where the central unit can be a weak point of the system. For example, electronic voting

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 55-63, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110805

may be controlled and falsified by its organizer [7]. The topic of electronic voting itself is a very complex issue that is not easy to implement on a larger scale [10-12].

One of the first countries that introduced electronic voting was Estonia [3]. The system enabled remote voting, so users required verification. Citizens have owned modern ID cards. They had a chip containing electronic data, certificates and private keys. This information were protected with PIN codes. It allows confirming identity of voters online. Remote voting had an advantage over the traditional approach because it was possible to change the ballot. However, analysis of the voting system has identified many potential security gaps [5]. One server has stored data of users with their encrypted ballot. Another server was responsible for counting ballots. It was able only to decrypt the ballots without revealing the personal data of the voters. Notwithstanding, the cooperation of these two systems could potentially reveal voters. In this case, maintaining anonymity depends on the Trusted Third Party (TTP) and can be a vulnerable point of the entire system. Currently, many voting systems have been developed that use blockchain technologies [14]. One of them is the Voatz app [16]. Developers talk about the use of blockchain as follows: "All ballots are secured on multiple, restricted-access, geographically-distributed servers running on blockchain technology to ensure all election data remains tamperproof" [6]. It is a solution that is being introduced to an increasing number of elections in the United States. In 2018 it was used in the US Midterm Election in West Virginia, and in 2020 it made it possible to vote in the presidential election in Utah.

The objective of the paper is to develop and create a voting scheme. The purpose of the scheme and proof-of-concept application is to create and conduct voting. Participation in voting will require user verification, but the user himself will remain anonymous in the system and it will not be possible to check what a given user voted for. The results of the research can be used to verify the methods of maintaining user anonymity using the blockchain network.

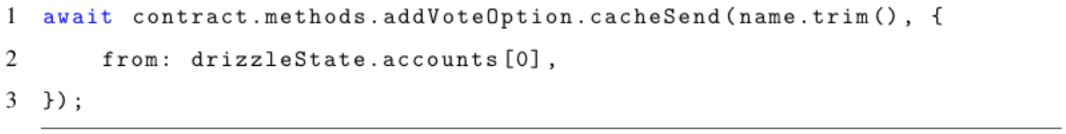
Structure of the paper is as follows. The introduction contains a brief history of blockchain networks, cryptocurrencies and electronic voting. Research papers that deal with the same issue are discussed in the related works section. Some of the solutions proposed there were applied in this research. Then there is a section on blockchain technology, which introduces the topic of blockchain and its architecture, the Ethereum project and the blind signature protocol. The next section presents the technologies that were used in the application development process. Chapter Voting procedure presents in chronological manner the entire procedure of creating voting by its organizer, sending ballots, their verification and finally the presentation of the results. Then there is a chapter on the application itself. It presents the result of paper. The last chapter is a summary of the paper along with an indication of the direction of further work.

2. RELATED WORK

In order to increase the anonymity of users, many works introduced a blockchain network to the voting protocol [14]. The decentralization of the system solves the problem of entrusting all responsibility to one entity. It also enables transparency of the elections but imposed the necessity to implement the protocol responsible for anonymity. The authors of paper [4], propose to use the blind signature protocol. A similar solution is in the paper of Yi Liu and Qi Wang [6]. The blockchain network is used to maintain transparency and the ability to verify the voting procedure by users, and the blind signature protocol enables hiding the identity of voters. The authors divide the participants of the voting process into three groups: voters, organizers and inspectors. Organizers verify voters and are responsible for collecting ballots. Inspectors are limiting the control of the organizers by additional interaction with the voters in the voting process. The system is resistant to ballot manipulation and forgery. As long as there is one honest organizer or inspector, an attack will not succeed. In the paper [17] by Qixuan Zhang, Bowen Xu, Haotian Jing and Zeyu Zheng, the authors presented a solution that is a simplified version of the previous paper. The system has only two types of users: voters and voting organizer. Each vote has only one organizer. The Ques-Chain smart contract acts as an inspector. It has a public key, a judge function to verify ballots and a ballot box to store valid votes. Our research also uses a blind signatures protocol and is improvement due to Ques-Chai work. In addition, the system was based on the Ethereum network due to the use of smart contracts.

3. E-VOTING SCHEME

Voting scheme starts with voters list preparation. The organizer prepares a list of public keys of cryptocurrency wallets of users who will be entitled to participate in the vote. This is an activity that is not performed in the scheme and the method of user verification depends on the organizer and the type of voting. The user list will be needed when verifying user blinded ballot, therefore this process can be done until then.



```
await contract.methods.addVoteOption.cacheSend(name.trim(), {
    from: drizzleState.accounts[0],
});
```

Listing 1. Adding vote option

Second step is generating keys by organizer using RSA algorithm. Before the organizer starts voting, the public and private key must be generated. The keys will be used for digital signatures of ballot and for their subsequent verification.

Third step is vote creation. The organizer can create a vote by creating a list of voting options. Candidates are signed up on a smart contract, which makes them visible to all users of both applications.

Fourth step is Sending a ballot for verification. The user selects an option to vote and votes. The ballot is then encrypted with the public key. The user receives a scrambled ballot and a blinding factor that will be needed later to decrypt the signed ballot. The next step is to send your ballot to the organizer.

```
const keyE = await Organization.methods.keyE().call();
const keyN = await Organization.methods.keyN().call();

const { blinded, r: keyR } = BlindSignature.blind({
    message: this.state.selectedOption,
    N: keyN,
    E: keyE,
});

const userAddress = drizzleState.accounts[0];

await Organization.methods
    .sendBlindedBallot(userAddress, blinded.toString(), date)
    .send({
        from: userAddress,
    });
```

Listing 1. Ballot signing and sending it to the Organization contract

Fifth step is ballot verification. The organizer must verify if the user is allowed to vote, having access to a list of all sent encrypted ballots. For this purpose, it compares the address from which the ballot was sent with the addresses on the list of users that was created earlier. After positive verification, the ballot is signed by the organizer and sent back to the user.

Sixth step is Sending a ballot. The user, using part of the public key and the blinding factor, can decrypt his ballot already signed by the organizer. The user then changes their wallet address. Thanks to that, the user remains anonymous after sending the ballot.

```
await QuesChain.methods
    .sendBallot(
        drizzleState.accounts[0],
        unblindedBallot,
        localStorage.getItem("voteOption")
    )
    .send({
        from: drizzleState.accounts[0],
    });
```

Listing 2. Sending a ballot to the Ques-Chain contract

Seventh step is results verification. The organizer has the opportunity to count the ballots and share the results. All verified valid ballots will be recorded on the Ques-Chain contract.

```
const verifiedBallots = ballots
    .filter((ballot) =>
        BlindSignature.verify({
            unblinded: ballot.unblindedSignature,
            key,
            message: ballot.content,
        })
    )
    .map((ballot) => ({
        id: ballot.id,
        userAddress: ballot.userAddress,
        unblindedSignature: ballot.unblindedSignature,
        content: ballot.content,
    }));
```

Listing 3. Verifying and filtering ballots

In final step the user has access to all verified and approved ballots on the Ques-Chain contract. It can display the number of ballots cast for each voting option. In addition, he can search for his ballot from the list of ballots taken into account in the election results. To do this, it searches for a ballot in terms of the address from which it was sent.

4. Proof of Concept E-Voting Implementation

The project consists of two separate applications dedicated to two groups of users. First is Voter which is a person taking part in voting. The second is voting organizer which is a person responsible for creating and conducting a vote. The application for voters will be available at the selected internet address. Any user can use the application and vote. Only verified users' ballots will be counted towards the voting results. The application for the voting organizer may be run locally and may not be available from any internet address. Two smart contracts were created for the purpose of voting. First is organization contract - used to create voting and verify users ballots, second is Ques-Chain contract - collects verified users ballots and allows you to check the voting result. The voting organizer can create a new vote by adding voting options to the Organization contract. Then the voter sends his ballot for verification. Also, the verification and return of the signed ballot takes place under the same contract. Only sending the verified ballot is done through the Ques-Chain contract. The organizer has access to them and can count the ballots and share the voting result.

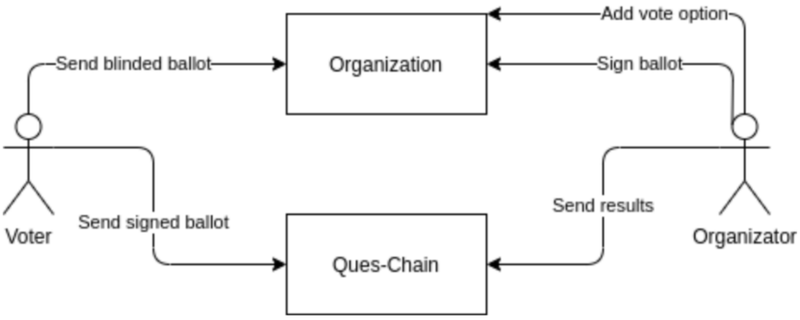


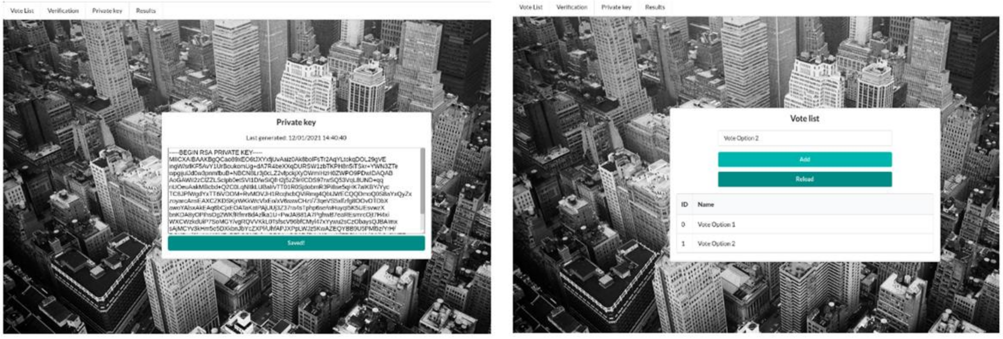
Figure 1. Traffic sample

The Organization contract was written in the 0.5.16 version of Solidity. The contract has three implemented data structures: First one is ballot - a voter's ballot that is created when an option is selected, and the ballot is sent for verification. It has a unique identifier, address of the person sending the ballot, content, i.e., the selected option, date of creation and information whether the ballot has already been verified. Second one is SignedBallot - Verified ballot of the voter, signed by the organizer. It has the same data as a regular voter's ballot, but without verification information, and the content is already encrypted information. Third is VoteOption - A voting option that is added by the voting organizer. It has a unique identifier and name to be displayed to the users of the application. Voting options are stored in the mapping structure. The function of adding options to a contract takes one parameter - the name of the option. The contract stores the organizer's public key data that is needed during the ballot-blinding process. There is one function that sets both used variables. Blinded ballot are held in the mapping structure. The function to send a ballot to a contract has three arguments: the address of the user who sends the ballot, the blind ballot, and the sending date. Ballots verified and signed by the organizer are kept in the mapping structure. The ballot signing function has four arguments: selected ballot identifier, user address, blind ballot, and date of signature.

```
handleSaveButton = async (event) => {
    event.preventDefault();

    const generationDate = moment()
        .format("DD/MM/YYYY HH:mm:ss");

    localStorage.setItem("privateKey", this.state.privateKey);
    localStorage.setItem("keyDate", generationDate);

    const key = new NodeRSA(this.state.privateKey);

    const { drizzle, drizzleState } = this.props;
    const contract = drizzle.contracts.Organization;

    const e = key.keyPair.e.toString();
    const n = key.keyPair.n.toString();

    await contract.methods.setKeyData(e, n).send({
        from: drizzleState.accounts[0],
    });

    this.setState({ saved: true });
};
```

Listing 4. Generating RSA key by organizator

The contract has one voting structure. It contains a unique identifier, user address, unblinded ballot and the selected option. User ballots are stored in the mapping structure. The user sends a ballot by specifying three parameters: user's address, unblinded signature and the selected option in voting. The voting results are serialized to a string variable. The share result function takes only one argument. The "Private" key page is used to generate the RSA key. The key can be generated using the PEM string. Above the field for entering the private key, the date of generating the current key is displayed. The date is generated and formatted using the moment library. The date and private key are saved in local memory. An RSA key is then generated using the key entered. The two components of the RSA key (e and n) will be used to encrypt the ballots. Therefore, they are included in the Organization contract.



Private key

Last generated: 12/01/2021 14:40:40

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCao89xEO6tJXYxfjUvAaiz0Ak8bolFsTr2AqYLtckqDOL29gVE
ingW/srlKF5AvY1UrBoukomUg+dA7R4beXXqDURSW1zbTKPH8n5iTSkr+YWN3ZTe
opgjul3d0w3pnmfbuB+NBCN8Lr3j0cLZ2vlpckpXyDWmIHzH0ZWPO9PDwIDAQAB
AoGAWl2zClZZLSclpb0etSVl1D/wSiQfH2j5zZ9/lCDS97rwSQ53VqL8UND+qq
nUOeuAakMBcbd+Q2C0LqNltkLUBaIVTT01R0SjdobmR3Pi8se5qHK7alKBY/Yyc
TC8JPfWgdYxTT8VOOM+RvMDVJH1RcqhcbQViRmg4Q6UWfECQQDmoQ0S8aYxQyZx
zoyarcAmiFAXCZKDSKjrWKkWcVbdFolxV6ezwCHz/73qeVS5sEzfg8OOvOTDbX
awoYAlsxAkEAq6bCjxEOATaKstPAjUUj3Z37ra4s1phpfisel/wHuyqt5K5UEsvwzX
bnKDA8yOPlhsOgZWKf8fmr8dAzlka1U+PwJA881A7Pghwt87eoREsmrcOjt7H4xi
WXCWzkdUlP7SoMGYlvgRQVVKkL0TsfscV96bfCMyl47xYywu2sCzObaysQJBAImx
sAjMCYv3kHm5e5DXkbnJbYcZXPlUhMAPJXPpLWJz5KwAZEQY8B0U5PMBz/YrH/
```

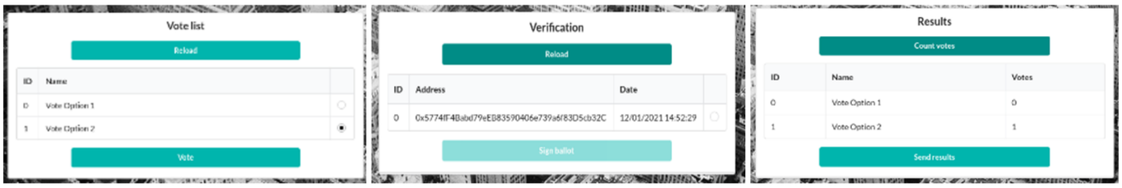
Vote list

Vote Option 2

ID	Name
0	Vote Option 1
1	Vote Option 2

Figure 2. GUI of implemented solution. Page to set RSA key and Voting list

Adding a new voting option requires an Ethereum transaction. For this purpose, a wallet in the MetaMask application was used. There is a transaction fee to complete the transaction. When you press the "Add" button, the current voting options from the contract are downloaded and then checked if the option name is unique. If the option name is not unique, an error message will be displayed. Otherwise, the name is sent to the contract and a new voting option is added to the list. If the list does not contain any options from the beginning, a message informing about it is displayed. The "Reload" button is used to refresh the list after adding a new voting option. The data is taken from the Organization contract. The "Verification" page is used to confirm the identity of voters and sign their encrypted ballots. The "Reload" button is used to download the entire list of ballots requiring confirmation. Each record has information about the wallet address from which the transaction of sending the ballot was made and the transaction date. After verifying the address, the organizer can choose a ballot and sign it. The ballot is then added to the list of verified ballots on the Organization's contract. When the "Count votes" button is pressed, all ballots from the Ques-Chain contract are downloaded. Then they are verified, i.e. the decrypted ballot is compared with the selected voting option. All verified ballots are added up and listed in the scoreboard. The organizer can publish the results after clicking the "Send results" button.



Vote list

ID	Name	
0	Vote Option 1	○
1	Vote Option 2	●

Verification

ID	Address	Date	
0	0x5774fF4Babd79eEB83590406e739a6f83D5cb32C	12/01/2021 14:52:29	○

Results

ID	Name	Votes
0	Vote Option 1	0
1	Vote Option 2	1

Figure 3. GUI of implemented solution. sending a ballot, ballot verification and vote results with verified ballot.

As for the voter implementation. The user can view the current voting list that is available on the Organization contract. After selecting an option from the list and pressing the "Vote" button, the ballot is encrypted and sent for verification. The "Signature" page displays information about the user's ballot verification process. When the ballot is verified by the organizer, the information will be displayed along with the exact approval date. Additionally, the verified ballot is checked if it has not been counterfeited and still holds information about the same ballot.

If the organizer has made the results available, the user can view them. The result table contains information about the name of a voting option and the total number of ballots cast for that option. In addition, the user can check if his ballot is on the list of ballots that make up the score. If the ballot has been counted, its data is displayed. Thanks to this, the user can see if his ballot is cast for the option chosen by him.

5. CONCLUSIONS

The research required an analysis of existing solutions in the field of anonymity and blockchain technology. The research also concerned theoretical works that had not yet been implemented. The result of the analysis was the creation of an architecture that was based on the solution proposed in the paper [17]. During creation of the paper, a blockchain-based voting scheme was presented and proof of concept application was developed. It uses a blind signature protocol to encrypt messages and keep voter anonymity.

Currently, a transaction fee is payable when making a transaction. If this is not a problem for the voting organizer, the user should not be charged with an additional fee for using this service. In addition, the user would have to have Ether on two wallets, which makes the voting process very difficult. The solution to this problem would be to transfer the obligation to pay the transaction cost to the voting organizer.

The application requires the user to send the verified ballot from one wallet and then send the verified ballot from the other wallet. The goal is to have a different address so that the user remains anonymous. The application does not automate this process and it is the user's responsibility to change the address. The application could use MetaMask to simplify this process.

The implemented ballot encryption protocol uses RSA encryption. Only the public key and the selected voting option are needed for the blinding process. The ballot encrypted in this way can be easily decrypted. Just compare it with the encryption results of all voting options. It would be necessary to introduce an additional method of securing an encrypted message.

The application allows only one election. Additionally, it does not provide the ability to edit voting options before publishing them. Currently, the organizer ends the voting when the ballots are counted, and the results are made available. The application should allow the creation of many different votes by different organizations for commercial use. Also, a system should be implemented that prevents voting after a given time. The results would be made available automatically without the intervention of the organizer.

The application fully implements the proposed architecture and enables voting. However, in order to be able to implement this system, many improvements for users should be introduced. These improvements, possible fixes and changes are listed below. The application requires some changes to be commercially applicable. Many of them facilitate the use of the application for users or adapt the use of Ethereum transactions to reduce the requirements for the user.

REFERENCES

[1] Li Y. Alvarez R., Levin I. Fraud, convenience, and e-voting: how voting experience shapesopinions about voting technology, May 2018.

[2] Chaum D. Blind signatures for untraceable payments, 1983.

[3] Maaten E. Towards remote e-voting: Estonian case, January 2004.

[4] Akram R. Markantonakis K. Hardwick F., Gioulis A. E-voting with blockchain: An e-votingprotocol with decentralisation and voter privacy, July 2018.

[5] Chowdhury M. Jabed M. Comparison of e-voting schemes: Estonian and norwegian solutions,September 2013.

[6] Wang Q. Liu Y. An e-voting protocol based on blockchain, 2017.

[7] Embele K. Ndu A. Awodola O. Mawutor J., Enofe A. Fraud and performance of deposit moneybanks, May 2019.

[8] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system, October 2008.

[9] Chen M. Jia C. Liu C. Wang Z. Shao W., Li H. Identifying bitcoin users using deep neural network,December 2018.

[10] KSHETRI, Nir; VOAS, Jeffrey. Blockchain-enabled e-voting. IEEE Software, 2018, 35.4: 95-99.

[11] Specter, Michael A., James Koppel, and Daniel Weitzner. "The ballot is busted before the blockchain: A security analysis of voatz, the first internet voting application used in us federal elections." 29th {USENIX} Security Symposium ({USENIX} Security 20). 2020.

[12] Park, Sunoo, et al. "Going from bad to worse: from internet voting to blockchain voting." Journal of Cybersecurity 7.1 (2021):

[13] Kedziora, Michal, and Wojciech Wojtysiak. "Practical Analysis of Traceability Problem in Monero's Blockchain." ENASE. 2020.

[14] Yang, Xuechao, et al. "Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities." Future Generation Computer Systems 112 (2020): 859-874.

[15] Trojanowska, Natalia, et al. "Secure Decentralized Application Development of Blockchain-based Games." 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC). IEEE, 2020.

[16] Lam, Peter. "From Helios to Voatz: Blockchain Voting and the Vulnerabilities It Opens For The Future.", 2017

[17] Jing H. Zheng Z. Zhang Q., Xu B. Ques-chain: an ethereum based e-voting system, May 2019

[18] Kedziora, Michal, et al. "Anti-Cheat tool for detecting unauthorized user interference in the unity engine using Blockchain." Data-Centric Business and Applications. Springer, Cham, 2020. 191-209.

[19] Kedziora, Michal, Patryk Kozlowski, and Piotr Jozwiak. "Security of Blockchain Distributed Ledger Consensus Mechanism in Context of the Sybil Attack." International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, Cham, 2020.

[20] Dimitriou, Tassos. "Efficient, coercion-free and universally verifiable blockchain-based voting." Computer Networks 174 (2020): 107234.

DETECT TEXT TOPICS BY SEMANTICS GRAPHS

Alex Romanova

Melenar, LLC, McLean, VA, US, 22101

ABSTRACT

It is beneficial for document topic analysis to build a bridge between word embedding process and graph capacity to connect the dots and represent complex correlations between entities. In this study we examine processes of building a semantic graph model, finding document topics and validating topic discovery. We introduce a novel Word2Vec2Graph model that is built on top of Word2Vec word embedding model. We demonstrate how this model can be used to analyze long documents and uncover document topics as graph clusters. To validate topic discovery method we transfer words to vectors and vectors to images and use deep learning image classification.

KEYWORDS

Graph Mining, Semantics, NLP, Deep Learning, CNN Image Classification.

1. INTRODUCTION

Nowadays data volumes are growing exponentially. For organizations that are daily getting huge amounts of unstructured text data, analyzing this data is too difficult and time consuming task to do manually. Topic analysis can solve document analysis problems as well as support other NLP problems such as search, text mining, and documents summarization.

Most common traditional approaches for topic analysis are topic modelings and topic classifications. Topic classifications as supervised machine learning techniques require topic knowledge before starting the analysis. Topic modelings as unsupervised machine learning techniques such as K-means clustering, Latent Semantic Indexing, Latent Dirichlet Allocation can infer patterns without defining topic tags on training data beforehand [1]. In this study we will introduce method of finding document topics through semantic graph clusters.

Word embedding methods such as Word2Vec [2], are conceptually based on sequential, logical thinking. These methods are capable of capturing context of a word in a document, semantic and syntactic similarity, and therefore solving many complicated NLP problems. However word embedding methods are missing capabilities to 'connect the dots', i.e. determine connections between entities. Understanding word relationships within documents is very important for topic discovery process and graph techniques can help to feel this gap.

In this article we will introduce a semantic graph model Word2Vec2Graph. This model combines word embedding and graph approaches to gain the benefits of both. Based on this model we will analyze long documents and uncover document topics as graph clusters. Document topics defined as semantic graph clusters will not only uncover sets of keywords, but will show relationships between words in topics.

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 65-72, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110806

Our novel Word2Vec2Graph model, a semantic graph built on top of Word2Vec model is created on Spark - a powerful open source analytic engine [3] with libraries for SQL (DataFrames), graphs (GraphFrames), machine learning, and NLP [1]. Until recently there were no single processing framework that was able to solve several very different analytical problems in one place. Spark is the first framework for data mining and graph mining right out of the box.

Finding text document topics within semantic graph can be done using various community detection algorithms. In this paper we will use a simple community detection method - graph connected components - subgraphs where any two nodes are connected to each other by paths, and which are not connected to any additional nodes.

To validate topic correctness through method independent on semantic graph topic discovery method, we will transform word vectors to images and use Convolutional Neural Network image classification technique. Please see Figure 1 that shows the data flow diagram for the process of finding and validating document topics.

In this paper we propose a new, graph-based methodology, which has the following original contributions:

- Introduced a novel Word2Vec2Graph model that combines analytic thinking and holistic thinking functionalities in semantic graph.
- Established an ability of the Word2Vec2Graph model to analyze long documents and discover document topics as graph clusters with relationships between words in topics.
- Proposed CNN image classification method for topic validation.

In the pages that follow, we will show:

- Studies related to semantic graph building methods and algorithms of finding text topics based on semantics graphs.
- Process of building Word2Vec2Graph model by getting document pairs of words, training Word2Vec model and building a graph for pairs with high cosine similarities.
- Topic discovery method through calculating connected components and top PageRank words within components.
- Topic correctness validation method by deep learning CNN image classification.

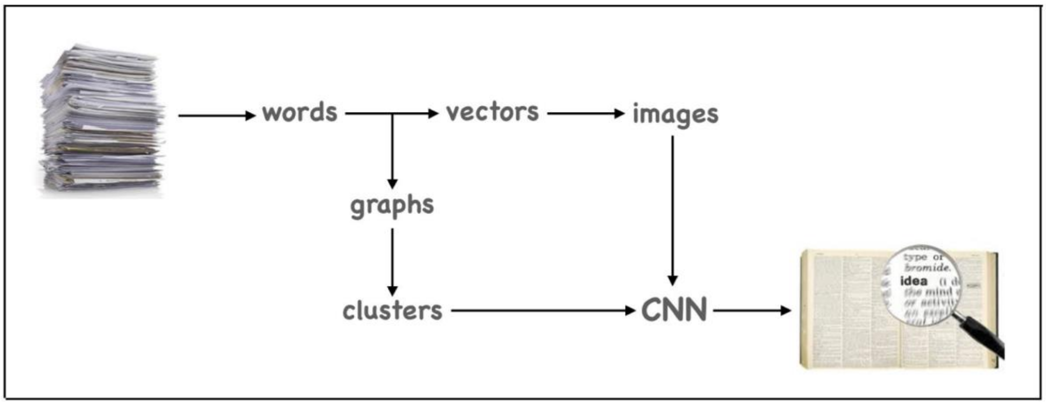


Figure 1. Finding text topics through a Word2Vec2Graph model and validating topics via CNN classification

2. RELATED WORK

There are various methods of building semantics graphs. Some of these methods are based on more traditional deep syntactic text analysis like RDF triples (subject–predicate–object) [4], other methods are based on unsupervised key phrase extractions and identifying statistically significant words [5] or on structuring asynchronous text streams [6].

Recently because of enormous progress of word embedding methods such as Word2Vec [2] some methods of building semantic graphs are based on word embeddings. For example, WordGraph2Vec method [7] is a semantic graph built on top of Word2Vec model that enriches text by adding target words for a specific context word in a sliding window.

Our Word2Vec2Graph model is similar to the WordGraph2Vec model [7] as in both models semantic graphs are built on top of Word2Vec. However in our semantic graph model we use pairs of words located next to each other in the document and mapping these words to vectors through Word2Vec model. For pairs of words we are calculating cosine similarities between words and building a graph based on threshold of pair similarities.

In recent years, there are some studies trying to integrate semantic graph structures with topic modeling. These models apply different methods of combining text with semantics graphs. Some studies integrate topic mining and time synchronization into a unified model [6] or combine semantic graphs with the textual information for topic modeling to estimate the probabilities of topics for documents [8]. Other studies are looking for topics through semantic graphs built on semantic relatedness between entities and concepts based on Wikipedia metadata [9]. In this paper to find topics we use a simple community detection method - graph connected components.

3. BUILD SEMANTIC GRAPH

To demonstrate our topic discovery method as data source we will use a document that consists of data about Creativity and Aha Moments that was manually extracted from several Wikipedia articles.

To build Word2Vec2Graph model and find document topics we will use Spark framework: Machine Learning and DataFrame libraries for Word2Vec model training and GraphFrame library for graphs. To process these methods, we will do the following:

- Retrain Word2Vec model.
- Extract pairs of words and calculate cosine similarities based on Word2Vec model.
- Build Word2Vec2Graph model.

Spark code is described in several posts of our blog[10].

3.1. Train Word2Vec Model

There are different approaches of using Word2Vec model for word embedding: using pre-trained model or training model on domain-specific corpus. Based on our observations, for topic finding Word2Vec models trained on domain-specific corpus work much better than pre-trained generic models. This observation corresponds with a study [11] that shows that domain-specific training corpuses work with less ambiguity than general corpuses for these problems.

To prove the difference, we trained two Word2Vec models. The first model was trained on generic corpus (News) and the second model was trained on combination of generic corpus and data about Stress extracted from Wikipedia (News + Wiki). In Table 1 you can see the differences of synonyms to words ‘Stress’ and ‘Rain’. As the word ‘Stress’ belongs to Stress corpus, the synonyms on these models are very different, but for a neutral word ‘Rain’ synonyms taken from these models are very similar.

Table 1. Examples of synonyms based on word2vec model corpuses: 'News' is word2vec model trained on generic corpus and 'News + Wiki' is word2vec model trained on combination of generic corpus and 'Stress' related corpus.

Stress		**Rain**	
News	**News + Wiki**	**News**	**News + Wiki**
risk	obesity	snow	snow
adversely	adverse	winds	rains
clots	systemic	rains	winds
anxiety	averse	fog	mph
traumatic	risk	inches	storm
persistent	detect	storm	storms
problems	infection	gusts	inches

Based on these circumstances, for topic discovery we will train the Word2Vec model on domain specific data corpus. Spark code for training and analyzing Word2Vec model can be found in our blog post [12].

3.2. Build Word2Vec2Graph Model

To build Word2Vec2Graph model, semantic graph on top of Word2Vec model, we will do the following steps:

- We will train Word2Vec model on the corpus that combines generic data (News) and domain specific data about Creativity and Aha Moment.
- From Creativity and Aha Moment data we will exclude stop words and tokenize other words.
- To discover document topics, instead of using a bag of words, we will look at pairs of words located next to each other in the document. To extract such pairs of words {word1, word2} we will use Spark Ngram function.
- For every word from word pairs we will get word vectors from Word2Vec model, i.e. for {word1, word2} pair we will map word1 to [word1 vector] and word2 to [word2 vector].
- Then we will calculate cosine similarities for word pairs, i.e. for {word1, word2} pair we will calculate cosine between [word1 vector] and [word2 vector].
- Finally, we will build a graph on word pairs with words as nodes and cosine similarities as edge weights. We will take only pairs of words with cosines higher than cosine similarity threshold 0.8.

Spark code for steps of building Word2Vec2Graph model can be found in our blog post [9].

4. UNCOVER AND VALIDATE DOCUMENT TOPICS

4.1. Uncover Document Topics

To detect document topics we will examine units of semantic graph that are separated from each other - graph connected components. Within each of these components we will find the most highly connected word using graph PageRank function.

For topic discovery we will do the following steps:

- Calculate connected components using Connected Components function from Spark GraphFrame library.
- Calculate graph PageRank scores by Spark PageRank function.
- For each connected component find the word with highest PageRank score and use this word as a topic class word.
- Map words to vectors and label vectors with topic class words.
- Transform vectors to images for CNN classification.

Spark code for topic finding and vector labelings can be found in our blog post [13].

4.2. Validate Topics

To validate topic correctness we will apply CNN image classification method. Vectors from uncovered topics will be converted to images with topic class words labels. Based on CNN image classification we will compare topics with image classes. This validation method does not fully prove topic modeling technique because clusters will have some noise: if two words are getting into the same image cluster it does not mean that they are highly connected. But if two words are in different image clusters they obviously do not belong to the same topic.

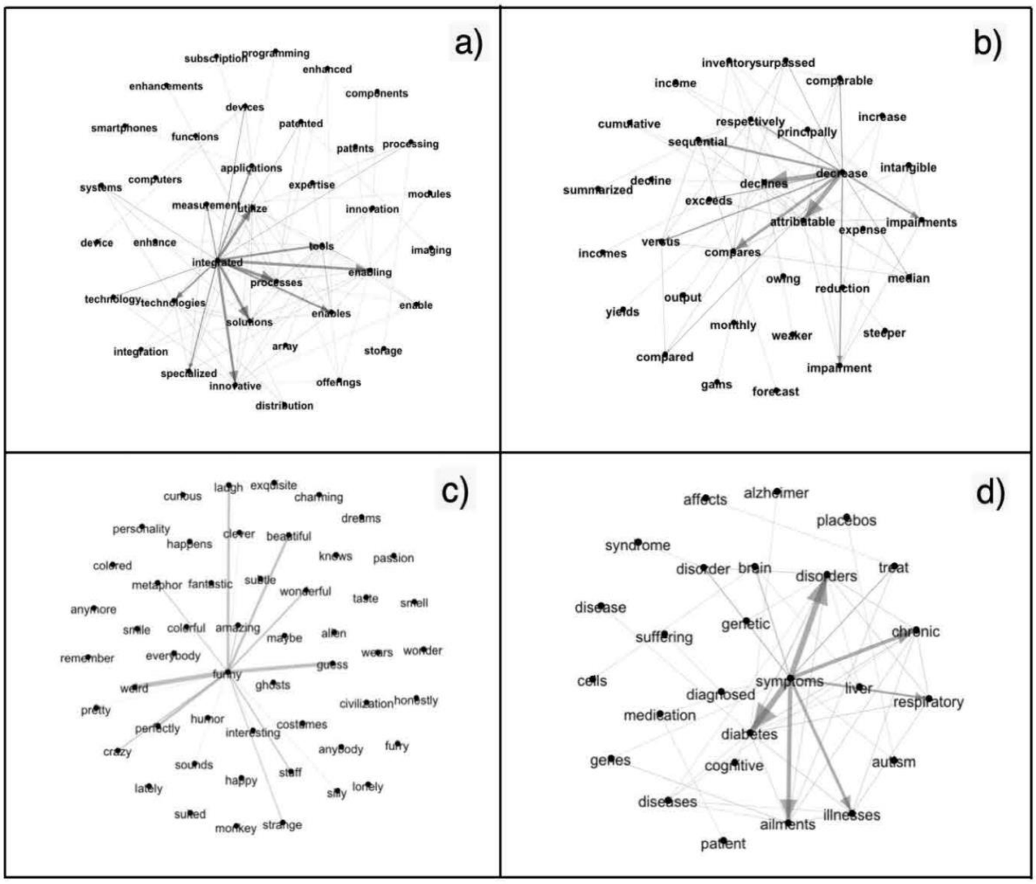


Figure 2. Subgraph topic examples: top PageRank words of topics: a) "integrated"; b) "decrease"; c) "funny"; d) "symptoms".

To convert vectors to images we will use Gramian Angular Field (GASF) - a polar coordinate transformation. The method was suggested by Ignacio Oguiza as a method of encoding time series as images for CNN transfer learning classification based on fast.ai library [14, 15]. To convert arrays to images and classify images we used open source code created by Ignacio Oguiza [16].

As usual, many graph connected components have very small sizes. For that reason for topics validation we used connected components with size bigger than 12 nodes. Our image classification model achieved accuracy about 91 percent.

4.3. Topic Examples

Topic examples are displayed in Figure 2. For each topic as a center of graph representation we use a topic class word and calculate a list of two degree neighbors (‘friends of friends’) around topics class words. Here are two degree neighbors for class word ‘symptoms’:

- symptoms -> brain; brain -> cells
- symptoms -> disorders; disorders -> cognitive

To find two degree neighbors we used Spark GraphFrame 'motif' technique [17] and transform the results to DOT language [18]. For graph visualization we used Gephi tool [19]. Spark code for graph visualization can be found in our blog post [13].

Topic visualization demonstrates an additional strength of using semantics graphs to uncover document topics: graph clusters that not only reveal sets of keywords in topics, but also demonstrate word relationships within topics.

5. CONCLUSION AND FUTURE WORK

In this paper we introduced a novel Word2Vec2Graph model that combines analytic thinking and holistic thinking functionalities in semantic graph. We demonstrated an ability of the Word2Vec2Graph model to analyze long documents and discover document topics as graph clusters that not only reveal sets of topic keywords, but also show word relationships within topics. For topic validation we suggested a novel CNN image classification method independent on semantic graph techniques.

In the future we are planning to do the following:

- Use more advanced word embedding models, like BERT, in particularly, examine phrase embedding process. Evaluate new Spark NLP library [1] that allows to fine tune various word embedding models and combine them with graph and machine learning models in Spark.
- Apply Word2Vec2Graph model to NLP problems that benefit from graph capacity to examine relationships between objects, such as entity disambiguation, semantic similarity, question answering, and others.
- Experiment with mapping words to vectors and vectors to images and classifying words and sequences of words through CNN image classification methods.

REFERENCES

[1] Alex Thomas (2020) *Natural Language Processing with Spark NLP*, O'Reilly Media, Inc.

[2] T Mikolov & I Sutskever & K Chen & GS Corrado & J Dean, (2013) "Distributed representations of words and phrases and their compositionality", Neural information processing systems.

[3] Bill Chambers &Matei Zaharia (2018) *Spark: The Definitive Guide: Big Data Processing Made Simple*, O'Reilly Media, Inc.

[4] Jurij Leskovec & Marko Grobelnik & Natasa Milic-Frayling, (2004). "Learning Sub-structures of Document Semantic Graphs for Document Summarization", LinkKDD 2004

[5] Juan Martinez-Romo & Lourdes Araujo & Andres Duque Fernandez, (2016). "SemGraph: Extracting Keyphrases Following a Novel Semantic Graph-Based Approach", Journal of the Association for Information Science and Technology, 67(1):71–82, 2016

[6] Long Chen and Joemon M Jose and Haitao Yu and Fajie Yuan, (2017) "A Semantic Graph-Based Approach for Mining Common Topics from Multiple Asynchronous Text Streams", 2017 International World Wide Web Conference Committee (IW3C2)

[7] Matan Zuckerman & Mark Last, (2019) "Using Graphs for Word Embedding with Enhanced Semantic Relations", Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13).

[8] Long Chen & Joemon M Jose & Haitao Yu & Fajie Yuan & Dell Zhang, (2016). "A Semantic Graph based Topic Model for Question Retrieval in Community Question Answering", WSDM'16

[9] Jintao Tang & Ting Wang & Qin Lu Ji & Wang & Wenjie Li, (2011). "A Wikipedia Based Semantic Graph Model for Topic Tracking in Blogosphere", IJCAI'11

[10] "Sparkling Data Ocean - Data Art and Science in Spark", http://sparklingdataocean.com/

[11] Yoav Goldberg & Graeme Hirst (2017) *Neural Network Methods in Natural Language Processing,* Morgan & Claypool Publishers.

[12] "Word2Vec Model Training", http://sparklingdataocean.com/2017/09/06/w2vTrain/

[13] "Word2Vec2Graph to Images to Deep Learning", http://sparklingdataocean.com/2019/03/16/word2vec2graph2CNN/

[14] Jeremy Howard, Sylvain Gugger (2020) *Deep Learning for Coders with Fastai and PyTorch*, O'Reilly Media, Inc.

[15] Zhiguang Wang & Tim Oates, (2015) “Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks”, Association for the Advancement of *Artificial* Intelligence (www.aaai.org).

[16] “Practical Deep Learning applied to Time Series", https://github.com/oguiza

[17] “Motifs Findings in GraphFrames”, https://www.waitingforcode.com/apache-spark-graphframes/motifs-finding-graphframes/read

[18] “Drawing graphs with dot”, https://www.ocf.berkeley.edu/~eek/index.html/tiny_examples/thinktank/src/gv1.7c/doc/dotguide.pdf

[19] “Visual network analysis with Gephi”, https://medium.com/@EthnographicMachines/visual-network-analysis-with-gephi-d6241127a336

AUTHOR

Alex Romanova Holds MS in mathematics from Faculty of Mechanics and Mathematics, Moscow State University and Ph.D. in applied mathematics from Faculty of Geography, Moscow State University, Moscow, Russia. She is currently a data scientist in Melenar, an expert in Knowledge Graph, NLP, Deep Learning, Graph Mining and Data Mining. Sharing her experience in technical blog: http://sparklingdataocean.com/

A DATA-DRIVEN INTELLIGENT APPLICATION FOR YOUTUBE VIDEO POPULARITY ANALYSIS USING MACHINE LEARNING AND STATISTICS

Wenxi Gao[1], Ishmael Rico[2], Yu Sun[3]

[1]University of Toronto, Toronto, ON M5S, Canada
[2]University of California, Berkeley, Berkeley, CA, 94720
[3]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

People now prefer to follow trends. Since the time is moving, people can only keep themselves from being left behind if they keep up with the pace of time. There are a lot of websites for people to explore the world, but websites for those who show the public something new are uncommon. This paper proposes an web application to help YouTuber with recommending trending video content because they sometimes have trouble in thinking of the video topic. Our method to solve the problem is basically in four steps: YouTube scraping, data processing, prediction by SVM and the webpage. Users input their thoughts on our web app and computer will scrap the trending page of YouTube and process the data to do prediction. We did some experiments by using different data, and got the accuracy evaluation of our method. The results show that our method is feasible so people can use it to get their own recommendation.

KEYWORDS

Machine Learning, data processing, SVM, topic prediction.

1. INTRODUCTION

Nowadays, with advanced technology, people can get the latest news quickly from the Internet. People try to catch on the trend in order to keep themselves from being left behind. On the Internet, people can broaden their horizons. They can keep abreast of current affairs and news, and get all kinds of latest knowledge and information online as well. For example, we can watch videos via YouTube [4]. The trending in YouTube shows the most popular videos in a period of time. It is a great platform for viewers. For people who create YouTube videos, YouTube can bring them not only popularity but also monetary encouragement to make creative videos. However, youtubers may sometimes have no idea of the video theme. Hence, this project is for youtubers who have trouble in deciding the content of the video. Youtubers can find similar trending videos through their own ideas to see if their topic is popular at that time. It provides ideas or materials for making new videos.

For some of the searching systems, like searching in YouTube, users can get popular videos based on the cumulative data of the whole site but not the recent data. Namely, people can get the videos that contain the keywords they entered but these videos might not be the hottest topic at that time. There is a list of trending videos but it will take some time to find what people want to get because they need to go through titles and contents.

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 73-78, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110807

Other techniques such as Twitter [5] can provide ideas but we cannot make sure that these ideas are advancing with the times. People share their thoughts and feelings on the platform with some tags, and different people have different ideas so sometimes they argue. For those who hesitate, it does not help, but instead makes them more entangled in choices of topics.

The difference between our method and other techniques is that we do both recommendation and trend at the same time while others can only meet one condition. In order to combine two functions, searching through trending videos and providing the new ideas, our method is to do searching and make analyses to trending videos. There are some advantages of our project. First of all, the result will only show one of the most related videos. It reduces many choices but provides the best one for reference. Secondly, the video for sure is trending. There is no doubt that the video is very popular at present because our analyses are all based on trending videos. And the third feature is that the result will provide the direct link to the related video so people can quickly get basic information. It saves the time of searching the video on YouTube. The efficiency is improved and our result is also the best.

In our experiment part, we are going to check the accuracy of our result: changing different datasets to check if the method is feasible and changing to another algorithm to compare the results between these two algorithms. First, we use different dataset as our training data and to test another class of data. Second, we make an experiment to check whether another algorithm is better than our method or not. After comparing the result with the fact, we get our percentage of accuracy. The higher of the percentages, the better of accuracy.

The rest parts of the paper are organized as follows: Next section gives the details that challenges and problem we met; Section 3 is about our solutions to the challenges we mentioned in the previous section; Section 4 shows the experiment we did to check how well our method works; Section 5 presents some related work; Section 6, which is the last part of the paper, gives the conclusion and discuss the future work.

2. CHALLENGES

2.1. Challenge 1: Thinking of a Video Topic

Nowadays, high technology is developing rapidly but some challenges come with it. Take YouTube as an example, we know that YouTube is an online platform where people share videos. It is useful for users to search the video but for youtuber, there is no practical tool for youtubers to produce ideas. Here is the first challenge how youtuber can think of an idea for their videos. For example, new youtubers come to this platform and want to make the first video to expand their influence, the first thing is to choose the right topic. However, inspiration comes suddenly, people seldom have any ideas immediately. Even though there are many types of videos in YouTube, this will be a burden for youtubers because there are millions of videos waiting for them.

2.2. Challenge 2: Keeping up the Pace of Trending

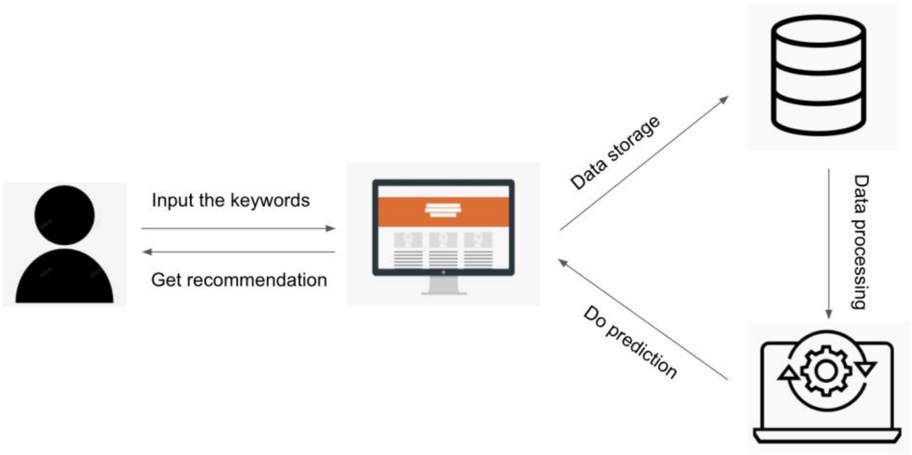
Media workers want to get a lot of attention from the public, but the content is hard to please all people. Sometimes, it is much more efficient to talk about hot spots. In our daily life, we can get news from social media but it changes frequently and fast, so we may have difficulty catching up. Some people may notice the beginning of the event on the news, and only some of them may know the end. Due to the fast speed of updating, people often miss important steps. Therefore, using a browser-like application is very important for people to understand and search popular trends. Moreover, this application has to be easy and concise so that people of all over the ages who know how to surf the Internet can operate easily.

2.3. Challenge 3: Trending Recommendation

There are a lot of videos introducing how to make your video topic on YouTube to attract more fans, they always tell you to choose a hot topic. But now it seems that few browsers can give ideas and catch up the trending simultaneously. For instance, there are many posts on the Internet that teach you how to make your first video but when you read the article, it may have been posted a long time ago. Those topics mentioned in the article might be popular when the article was just released, but now it is not a trend anymore. Another example is that we can see worldwide trending in Twitter but we cannot get recommendations for the contents of tweets. Therefore, the platforms can mostly only do the recommendations of contents or show trends but neither of them at the same time.

3. SOLUTION

This system works in two parts: one part is to search from the trending videos on YouTube, and another part is to analyze the data and give back the class that matches the input the most relatively. Users first type the main topic or some keywords of their video in the input box, and this system will use tfidfvectorizer to convert the text to a matrix with their frequency. Then, the support vector machine (SVM) [6, 7, 8, 9] will find the most related trending videos based on this matrix. Finally, it shows the result of the prediction which is the recommendation of the video.



From the overview diagram, we can see that these four steps are indispensable. Users input the keywords and get their recommendation in the web app. The web app needs YouTube scraping and SVM, and the SVM part needs YouTube scraping and preprocessor to filter the data.

There are four main components in this recommendation system. The web app, where users input their thoughts and get the result back, is one of the main components. In our webpage, there is a predict button, and users can click that button to get recommendations and also, users can see the trending video list. The result will be shown at the bottom.

We need YouTube data so YouTube web scraping is important. It needs to extract useful data from the YouTube website. We use "json" [10, 11] to get the HTML information and from those HTML [12] text, we can get video IDs, short descriptions and titles of the list of trending videos.

The third step is to vectorize the data. Because the raw data cannot be used directly, we need to process the data before our prediction. From scraping the YouTube web, our data is in string type but when we go further prediction, the trained data must be numbers. Therefore, we need to process the data before we use them. After that, our data is ready to make the model.

The last step is SVM. In this part, we are going to use the dataset in the previous part and make a model to predict.

In our web app, we use python flask to make a link to do prediction so that we can use it in our JavaScript. Also, we write DOCTYPE to build the webpage of our prediction program and a style CSS file to make it look functional and nice. There is a trending video list on the page for people to view the trending conveniently. The result will show under the prediction button. That is how our webpage looks like.

For YouTube data scraping, we use some libraries such as requests to get HTML text from YouTube website and json to convert JSON objects into the python dictionary. These two libraries allow us to successfully obtain the information we want from the trending list. We write two text files to save the HTML document and video links in the trending list. From the text files, we can find the information conveniently and cut them out for processing.

In our data processing part, we use TfidfVectorizer [13, 14] from sk-learn library [15] to convert the strings into numbers. Also, in order to make the data frame, pandas help us create the data frame into five columns: video ID, title, length seconds, keywords and short description.

Last but not least, our project needs SVM from the sk-learn library to make the model and do prediction. In this model, X data is vectorized short descriptions and Y data is vectorized titles of videos. When doing the prediction, the text that users enter cannot be used directly, so we need to vectorize the text as well. By these four parts, we can do the prediction.

4. EXPERIMENT

One of the most important things for this program is to give back an accurate result based on input content. To evaluate the accuracy, we need to test if the video ID is the most relevant to what we enter. Therefore, we conduct experiments with two directions: the first one is to test the data we have and the second is to change another algorithm.

For experiment 1, we use video descriptions and ID as our training data and video titles as our testing data. The number of training data in the experiment is the same as the number of training data in the original prediction. The reason why we choose titles as our test data is that each video title corresponds to an ID respectively, we can compare the fact to the outcome to check whether the prediction is correct or not. After that, we can get a percentage of the accuracy.

From the testing, the score that is based on video titles is about 0.6923. In other words, it is 69.23% accuracy to get the video ID if we enter video titles. This result shows that these predictions can still be justified.

In our second experiment, we use a new algorithm called the “KNN (K-Nearest Neighbor) [16]” algorithm. We want to compare the result of SVM and KNN which is more accurate. In our experiment, we take short descriptions and video id as training data and still use videos’ titles to predict. The training data and the test data are the same. We compute the percentage of correct results.

The result of experiment 2 reveals that KNN algorithm is better. From the outcome, we get the score of the prediction using KNN algorithm is about 0.7033. From testing of prediction using the SVM algorithm, we have 69.23% accuracy and for KNN one, it is about one percent more accurate (70.33%).

5. Related Work

Paek, Hye-Jin, et al [1] showed a machine learning approach in the analysis of antismoking video contents on YouTube by using four characteristics of 934 antismoking videos. We are using only trending videos which have different contents on YouTube to analyze the popular video keywords or contents.

Covington, Paul, et al [2] presented a nice deep machine learning approach on YouTube videos recommendation system for viewers. In this work, we know how the recommendation system works. The difference between our project and theirs is that we apply recommendation systems for youtubers, but theirs is faced to viewers.

Victor Roman [3] explored some methods to make the model for the data in machine learning. In this article, we consider the optimization of the training model. Therefore, in our experiment, we tried another model to see if it is more accurate.

6. Conclusion and Future Work

In conclusion, this recommendation web app is for people who want to make YouTube videos but do not have any ideas. After people type their keywords in the box, the website starts to do the prediction. The web app first collects the data from YouTube, and saves the data into files. The data needs to be processed to use as training data. The last step is to do prediction by the data we collect and to display the information so that people can see the trending video and get the recommended trending video. And we apply our method to do the experiment with different data and algorithms, the result is acceptable. These good results of our experiments demonstrate the good accuracy of our method and it is feasible to search through the trending list and do recommendation at the same time by our project.

Because the system is based on the limited data of trending videos, the keywords entered by people may not match the data at all, resulting in inaccurate results. However, we have no training data if we start a brand-new recommendation system. Moreover, people who want to make YouTube videos always struggle with the content of the video and our recommendation system is for those who do not have ideas on making videos. They can use the system well. Tfidf calculates the word frequency which very relies on the database, but the disadvantage is that keywords do not appear frequently and it cannot reflect the importance of the word in context. Therefore, the optimization of using tfidf is needed to improve our system.

Since the trending video list is changing every time, we could save the data of trending in a period of time so that we have more data to recommend. We can add some numbers to do the classification so that the result by tfidfvectorizer will be more accurate.

REFERENCES

[1] Paek, Hye-Jin, et al. "Content Analysis of Antismoking Videos on YouTube: Message Sensation Value, Message Appeals, and Their Relationships with Viewer Responses." OUP Academic, Oxford University Press, 5 Oct. 2010, academic.oup.com/her/article/25/6/1085/660720.

[2] Covington, Paul, et al. "Deep Neural Networks for YouTube Recommendations." RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems, Sept. 2016, doi.org/10.1145/2959100.2959190.

[3] Roman, Victor. "How To Develop a Machine Learning Model From Scratch." Medium, Towards Data Science, 2 Apr. 2019, towardsdatascience.com/machine-learning-general-process-8f1b510bd8af.

[4] Burgess, Jean E. "YouTube." Oxford Bibliographies Online (2011).

[5] Murthy, Dhiraj. Twitter. Cambridge: Polity Press, 2018.

[6] Jakkula, Vikramaditya. "Tutorial on support vector machine (svm)." School of EECS, Washington State University 37 (2006).

[7] Wang, Lipo, ed. Support vector machines: theory and applications. Vol. 177. Springer Science & Business Media, 2005.

[8] Noble, William S. "What is a support vector machine?" Nature biotechnology 24, no. 12 (2006): 1565-1567.

[9] Ma, Yunqian, and Guodong Guo, eds. Support vector machines applications. Vol. 649. New York, NY, USA: Springer, 2014.

[10] Nurseitov, Nurzhan, Michael Paulson, Randall Reynolds, and Clemente Izurieta. "Comparison of JSON and XML data interchange formats: a case study." Caine 9 (2009): 157-162.

[11] Pezoa, Felipe, Juan L. Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. "Foundations of JSON schema." In Proceedings of the 25th International Conference on World Wide Web, pp. 263-273. 2016.

[12] Duckett, Jon. HTML & CSS: design and build websites. Vol. 15. Indianapolis, IN: Wiley, 2011.

[13] Kumar, Vipin, and Basant Subba. "A TfidfVectorizer and SVM based sentiment analysis framework for text data corpus." In 2020 National Conference on Communications (NCC), pp. 1-6. IEEE, 2020.

[14] Subba, Basant, and Prakriti Gupta. "A tfidfvectorizer and singular value decomposition-based host intrusion detection system framework for detecting anomalous system processes." Computers & Security 100 (2021): 102084.

[15] Feurer, Matthias, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. "Auto-sklearn: efficient and robust automated machine learning." Automated Machine Learning (2018): 113-134.

[16] Peterson, Leif E. "K-nearest neighbor." Scholarpedia 4, no. 2 (2009): 1883.

AUTHOR

Wenxi Gao now is an undergraduate student in University of Toronto Scarborough in Canada. She studies in the specialist program in Mathematics - Statistics Stream.

WHAT ARE THE ASPECTS OF ADOPTING COMPUTER-BASED EXAMS AND DO THEY IMPACT NEGATIVELY ON STUDENTS?

Rabea Emdas[1] and Ahmed Alruwaili[2]

[1]Faculty of Science, Engineering and Technology, Swinburne University
of Technology, Hawthorn, Victoria 3122, Australia
[2]Department of Computer Science and Information Technology,
La Trobe University, Bundoora, Victoria 3086, Australia

ABSTRACT

Computer-based exams (CBEs) have been used in various courses, such as schools, universities and other training centres. As there are many educational institutions which have chosen to convert from paper test system to computer- based exam. However, adopting computer tests may lead to some difficulties for the students, which relates to technical defects and lake of computer skills of some students when they applying the computer based exams. The purpose of the essay was to determine negative and positive effects on the students of using computer-based exams and focus on some of suggesting solutions to the negative effects, such the exams to make continuous use of computer- based possible. In the first section the computer test, which could cause negative effects on students due to various levels of skills to use computer and some technical problems was examined. The design of the computer examination system requires careful planning and study from several aspects before becoming officially accepted, the computer-based exams still have a few problems which may lead to difficulties in using computer exams. Then the many benefits which could be gained by using computer-based exams, such as the student will be more independent with computer test were described. In addition, the students have accessible to the exams through the internet network. Finally, the effectiveness of certain strategy to solve the negative effects of computer-based exams were argued. developing the solutions of the technical problems are required for computer test, where improving the input methods questions and corrections. It was concluded that the computer exam, with adjustments, is more suitable for students.

KEYWORDS

Computer-Based Exams, Computer test, computer system examinations, computer test model.

1. INTRODUCTION

The computer-based exam is a system designed for the students' to do exams via computer. Therefore, electronic devices which are providing developed software has changed to be more suitable in educational operation [1, 2]. All these developments of the schooling devices have made it possible for students to do computer tests with better performance. On the other hand, adopting computer tests may lead to some difficulties for the student, which is related to technical defects and computer skills when applying the computer-based exam. These difficulties prevent the application of the computer test easily. For example, unequal levels of computer skills among the users could be a barrier to apply this kind of test equally. In addition, technical problems that are related to hardware and software are considered as one of the practical problems [3-5].

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 79-83, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110808

However, there are some advantages for the computer-based exam compared with the paper tests and that represented in easy access and accuracy of the assessment of the questions. As a result, the students become more independent in the study, due to the student not receiving help from anyone. This essay will discuss negative and positive effects on the students when the computer-based exams been used and focus on suggesting some solutions to the negative effects of such exams to make continuous use of the computer based exams possible [6, 7].

2. BACKGROUND

Computer-based exams (CBEs) are software of computer that support test papers to present electronically. According to Bennett, Braswell, Oranje, Sandene, Kaplan and Yan [6], electronic tests are powered by a database containing all the subjects' data of the student. In addition, electronic tests have function which are correcting the questions accurate and showing individual results for each question. According to Goldberg and Pedulla [5], computer-based exams are supported the materials and topics differently, such as questions of mathematics test would use many functions more than history questions. The test which is done by the computer includes questions that are connected by database. Therefore, computer-based tests might have less error than the paper examinations. The computer exam is highly bases on the new technology, in order to accurately assess the performance of the students during the exam. For example, supporting of the new functions and developed versions of the software were used in the exam [8,9]. The electronic system test is designed to be done by computer which is provided with many options. For example, the computer test could have direct messages which can displayed as a result for each question [7]. In addition, the exam which using the computer based is more accessible for students even through the internet, while the paper examination could not support like this option [7, 10].

3. THE NEGATIVE IMPACTS OF THE ELECTRONIC TEST ON STUDENTS

The use of computer tests may have significant negative impacts on the student when the student is doing computer exams. According to Frankel, Altschuler, George, Kinsman, Jimison, Robertson and Hsu [11], the computer-based exam has a high accuracy when calculate the errors of the students during the exams. Consequently, lack of computer skills might have a negative impact on students when doing the computer-based exams, because of the concentrating of student to answer questions and how to use the computer. However, the computer test is accreted and gives the specific time to the student for reviewing errors during the examination, while in the paper exam could not deal with time accurately same as and cheque errors. Table 1 depicts elements of the Input- Skill for the student, focusing on accuracy and speed [6].As can be seen from Table 1, accuracy skills are 17 maximum points. While the speed skills are 22 points, which cloud means that the students may already release some of his errors after he knew the calculation marks in the computer test. Therefore, the problems of computer exams could hinder the student to continuing in the exam, which may impact on the student's performance. Due to of these similar problems the concentrating of the student could really be shifted from answering questions to technical problems. Shacham [1] stated that the computer problems which are related to hardware and software could affect on the performance of student, which is possibly making the student return the exam from beginning. These two major areas of problem might be negatively affecting the level of student performance [7, 12].

4. THE POSITIVE EFFECTS OF THE ELECTRONIC TEST ON STUDENTS

Using computer-based exams could have a positive effect on students, due to electronic tests provide many options for the students which were not available in the paper examinations.

However, the electronic test system focuses on the time factor and quick access to records. The computer test system is provided on electronic records that are different from paper records [1, 10]. In addition, these kinds of questions could include multiple choices for the student, such as inputting maths expressions and numerical answers. Moreover, the computer-based exam calculates the time of the test accurately, and displays student's mistakes at the same time. The questions show on the screen sequentially. As a result, the students could understand each question individually. Computer Based Exams (CBEs) have been used in various courses [1, 2]. In such tests, all the questions are answered via individual interactive work with a personal computer. Immediate feedback on errors is provided and the rating is typically done by the computer. The time frame of the computer-based exams could be set so that the largest number of students able to redo the exam. There is no assistance to the student from anyone, which could make the student more independent. This means that the student would take responsibility for study [3]. In addition, the computer-based exam supports simulation tests for some special applications, such as computer and physics experiments. This means that the computer test can be used to simulate some practical experiments which are requiring high cost and hardware. Thus, the students can obtain a hands-on experience before commencing to do some real exam. All these steps could assist to increase the student's performance [4].

Table 1. Elements of the Input- Skill

Variable Accuracy	Max. Points	Variable Speed	Max. Points
Typing and editing		Typing and editing	
Accuracy typing a brief given passage	2	Time to type a brief passage	2
Accuracy inserting a word	2	Time to insert word	2
Accuracy changing a word	2	Time to change word	3
Navigating the test		Navigating the test	
Accuracy pointing and mouse clicking	2	Time to point and click	3
Accuracy scrolling	2	Time to scroll	2
Accuracy clicking on the “Next” icon	2	Time to click on “Next”	3
Accuracy clicking on the “Previous” icon	2	Time to click on “Previous”	2
Entering responses		Entering responses	
Accuracy filling in a mixed number	2	Time to fill in mixed number	2
Using the calculator		Using the calculator	
Accuracy in performing a given operation	1	Total time to complete the calculator tutorial	3
Total Of Accuracy	17	Total Of Speed	22

5. THE STRATEGIES TO SOLVE THE NEGATIVE EFFECTS OF COMPUTER - BASED EXAM SYSTEMS

The electronic test system may have a few errors as does any system which is built to solve a problem or develop the old system. The computer-based exam system is relatively recent compared with the paper examination system. According to Shacham [1] the computer based exams require extra development to support multiple functions and some tools. Therefore, the computer test should be more suitable to the students. For example, voice technology is the technique used in some international tests, such as the TOEFL exam, which gives the student a chance to display conversation skills. All these improvements in the computer system test could have a positive effect on the students whom would increase their performance. According to Mary [3] solving of technical problems can be caused to develop computer test for example, improve the input method of questions and corrections, which should make the computer exam more simple for the students. Taking into account the student's needs considering the sensitive issue of the technical problems in the computer test, which could affect skills of the students more than traditional exams. Such as the rebooting computer system for various reasons which often leads to wasting time and causes frustration. This complaint of students about computer exams is related to the technical problems, these two major strategies could solve problems of the computer test.

6. Conclusions

Design of the computer examination system requires a lot of planning and study from several aspects before becoming certified. Using electronic tests may have benefits to students. On the other hand, the computer-based test still has a few problems; these problems may lead to difficulties in using computer tests, such as technical problems and computer skills. As a result, the student should have sufficient time to gain good experience in using computer tests, where this test system focuses on the time factor and quick access to records, the computer- based exams calculate the time of the test accurately. Therefore, the test could assist performance of student, which make test more suitable for the students, such as input method questions and corrections, and using technology by various ways, where is making the computer-based exam support voice input, editing texts and easily update. However, technical problems with the hardware could be decreased by using new and well-maintained computers. It is expected that with the increase of using the computer-based exam at educational institutes, while the using of the paper examination may decrease.

Acknowledgements

The authors would like to thank Dr Abd Alhamid R. Sarhan (Faculty of Science Engineering and Technology - Swinburne University of Technology) for his support.

References

[1] Shacham, M.: 'Computer-based exams in undergraduate engineering courses', Computer Applications in Engineering Education, 1998, 6, (3), pp. 201-209.

[2] Wingenbach, G.J.: 'Agriculture Students' Skills and Electronic Exams', Journal of Agricultural Education, 2000, 41, (1), pp. 69-78.

[3] Mary, P.: 'The Effect of Using Item Parameters Calibrated from Paper Administrations in Computer Adaptive Test Administrations', The Journal of Technology, Learning and Assessment, 2007, 5, (7).

[4] Blumenstein, M.: 'Synergies of Learning Analytics and Learning Design: A Systematic Review of Student Outcomes', Journal of Learning Analytics, 2020, 7, (3), pp. 13-32.

[5] Goldberg, A.L., and Pedulla, J.J.: 'Performance Differences According to Test Mode and Computer Familiarity on a Practice Graduate Record Exam', Educational and Psychological Measurement, 2016, 62, (6), pp. 1053-1067.

[6] Randy Elliot, B., James, B., Andreas, O., Brent, S., Bruce, K., and Fred, Y.: 'Does it Matter if I Take My Mathematics Test on Computer? A Second Empirical Study of Mode Effects in NAEP', The Journal of Technology, Learning and Assessment, 2008, 6, (9).

[7] Zilles, C., West, M., Mussulman, D., and Bretl, T.: 'Making testing less trying: Lessons learned from operating a Computer-Based Testing Facility', in Editor (Eds.): 'Book Making testing less trying: Lessons learned from operating a Computer-Based Testing Facility' (IEEE,2018, edn.), pp. 1-9.

[8] Morrison, B.B., Margulieux, L.E., Ericson, B., and Guzdial, M.: ‘Subgoals help students solve Parsons problems’, in Editor: ‘Book Subgoals help students solve Parsons problems’ (2016,edn.), pp. 42-47.

[9] Zilles, C., Deloatch, R.T., Bailey, J., Khattar, B.B., Fagen, W., Heeren, C., Mussulman, D., and West, M.: ‘Computerized testing: A vision and initial experiences’, age, 2015, 26, pp.1.

[10] Hainey, T., Connolly, T.M., Boyle, E.A., Wilson, A., and Razak, A.: ‘A systematic literature review of games-based learning empirical evidence in primary education’, Computers & Education, 2016, 102, pp. 202-223.

[11] Frankel, R., Altschuler, A., George, S., Kinsman, J., Jimison, H., Robertson, N.R., and Hsu, J.: ‘Effects of exam-room computing on clinician-patient communication: a longitudinal qualitativestudy’, J Gen Intern Med, 2005, 20, (8), pp. 677-682.

[12] Zilles, C.B., West, M., Herman, G.L., and Bretl, T.: ‘Every University Should Have a Computer-Based Testing Facility’, in Editor (Eds.): ‘Book Every University Should Have a Computer-Based Testing Facility’ (2019, edn.), pp. 414-420.

CLOCK GATING FLIP-FLOP USING EMBEDDED XOR CIRCUITRY

Peiyi Zhao[1], William Cortes[1], Congyi Zhu[2] and Tom Springer[1]

[1]Fowler School of Engineering, Chapman University, Orange, CA, USA
[2]Nangjing University, Nanjing, China

ABSTRACT

Flip flops/Pulsed latches are one of the main contributors of dynamic power consumption. In this paper, a novel flip-flop (FF) using clock gating circuitry with embedded XOR, GEMFF, is proposed. Using post layout simulation with 45nm technology, GEMFF outperforms prior state-of-the-art flip-flop by 25.1% at 10% data switching activity in terms of power consumption.

KEYWORDS

Dynamic power, low supply voltage, flip-flop.

1. INTRODUCTION

While the human brain consumes a little energy but can work with remarkable speed and response at real time, processors in data centers and personal computers would consume considerable energy. The System on Chip design integrates billions of transistors on one chip. However, current cooling equipment has a limited capability for removing the excess heat. Hence, one of the major challenges for digital systems is to minimize power consumption.

The clocking system consisting of clock distribution network and flip-flops (FFs) [1-7] consumes a large portion of dynamic power. In server processors, flip-flops take up to 20% of the total dynamic power of the processors. Hence FFs have become one of the dominant contributors to dynamic power [1].

In FFs, the dynamic power consumption is highly related clocking power consumption. There are different low power flip flops in the literature, for example, Transmission Gate FF (TGFF) [2], Topologically Compressed FF (TCFF) [3], Change Sensing FF(CSFF) [4], Static Contention Free FF (SC^{2}FF) [8]. TGFF uses two phase clocks and consumes clocking power even when data does not change. TCFF cannot work at low voltage due to reduced voltage swing at internal nodes. SC^{2}FF has a dynamic node which consumes power. CSFF reduces clocking power consumption by sensing whether input changes or not. When input does not change, there will be no redundant transition in CSFF. However, CSFF uses many stages which causes large DQ delay and there is still room to further reduce clocking power.

To further reduce power consumption, a change detect FF is proposed in this paper.

This paper describes the proposed fine-grained clock-gate FF in section II. Section III presents simulation results and an overall conclusion is presented in section IV.

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 85-89, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110809

2. PROPOSED CLOCK GATING FLIP-FLOP

We propose a fine-grain clock-gate flip-flop using embedded XOR circuitry, GEMFF, as shown in Figure 1.

Clock gating control of GEMFF is built by employing a ‘compare’ block, consisting of NMOS N5, N6, N7, and N8, implementing a logic XOR function D⊕Q in terms of transistor ON/OFF operation, embedded in the pull down clock network.

When D does not change, Q remains the same as D, the ‘compare’ block turns off and stops clock propagation, the output will keep its value.

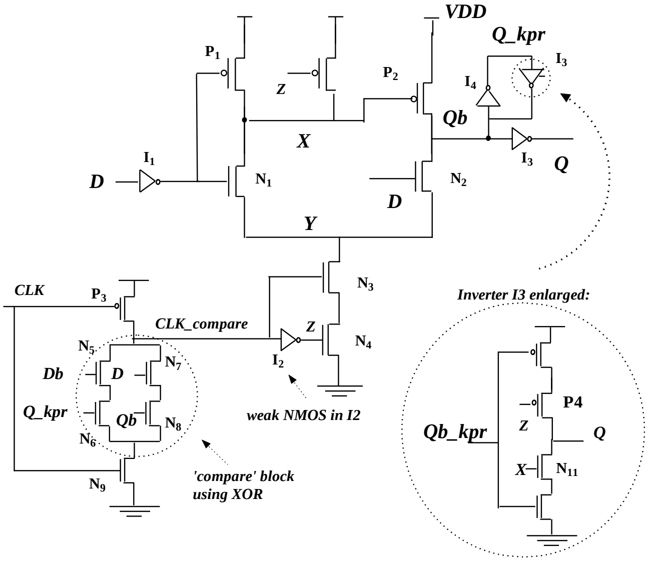


Figure 1. Proposed flip-flop using clock gating circuitry with embedded XOR, GEMFF

When D changes, XOR block turns ON, so ‘CLK_compr’ discharges to ‘0’ by the XOR network and sets Z to be ‘1’ and turns N4 ON.

Following that when CLK falling edge comes, ‘CLK_compr’ rises to ‘1’ and turns N3 ON while N4 will stay ON for a short time until Z falls to ‘0’, hence both N3, N4 will be ON for a short time which produces the pulse during which D is sampled.

The pulse width is determined by the delay of inverter, I2. A weak NMOS transistor is used in I2 to produce longer delay in order to have a wide pulse width. A larger gate length(multi times of 50nm) is used in that NMOS in I2. When that NMOS is weak, it will turn the transistor N4 off slowly. GEMFF is a negative clock edge triggered flip-flop.

3. SIMULATION RESULTS

The simulation results have been obtained from post-layout simulation in 45nm CMOS technology at room temperature using Hspice. The parasitic capacitances were extracted from the layouts. VDD is 1.0 V, and the clock frequency is 500 MHz. The setup used in our simulations is as follows. In order to obtain accurate results, we have simulated the circuits in a real environment, where the flip-flop inputs (clock, data) are driven by the input buffers, and the output is required to drive an output of 4 standard sized inverters. Standard values are used for the sizes of PMOS and NMOS transistors, respectively.

The total power consumption includes the power consumption of flip–flop as well as the power consumption of the clock driver and the data driver.

Table 1 shows comparison of the flip-flop characteristics in terms of the number of total transistors, number of clock related transistors, number of clocked transistors switching during data idle time in a flip-flop, low voltage operation, redundant- transition-free, the transistor width, CQ delay, setup time, hold time and power consumption under switching activity of 5%, 10% and 20%, respectively.

Table 1. Comparing the FFs in Terms of the Number of Transistors, CQ Delay, and POWER

	Transistor count	**Total transistor width (um)**	**CQ (ps)**	**Set-up (ps)**	**Hold Time (ps)**	**Power (µw) 5% activity rate**	**Power (µw) 10% activity rate**	**Power (µw) 15% activity rate**
CSFF	24	5.16	46.3	92.2	-25.1	2.63	3.74	4.55
GEMFF	24	4.68	58.2	66.8	51.4	2.06	2.80	3.94

Figure 2 shows the power consumption comparison under different switching activities. The lower the switching activity, the more the power saving of GEMFF over CSFF. GEMFF consumes less power than CSFF by 21.8% and 25.1% at switching activities of 5% and 10%, respectively. Its power improvement over CSFF reduces to 13.5% at the switching activity of 15%.

Figure 3 shows the power consumption at different supply voltages from 0.4V to 1.0V. GEMFF has less power consumption over CSFF in all of the above voltages.

In terms of data-to-clock (DQ) delay, GEMFF’s delay, 125.0 ps, is about 10% less than CSFF’s delay, 138.5 ps.

In terms of PDP, CSFF’s PDP is 5.11 fJ while GEMFF’s PDP is 3.51 fJ at 10% switching activity. Hence, GEMFF has energy improvement over CSFF by about 30%.

Though power consumption has been improved in GEMFF, the hold time time is increased in GEMFF comparing with CSFF.

The above results demonstrate that GEMFF has lower power consumption than CSFF. Notice that GEMFF has larger hold time than CSFF, since GEMFF is a pulse triggered FF. This is a limitation of GEMFF.

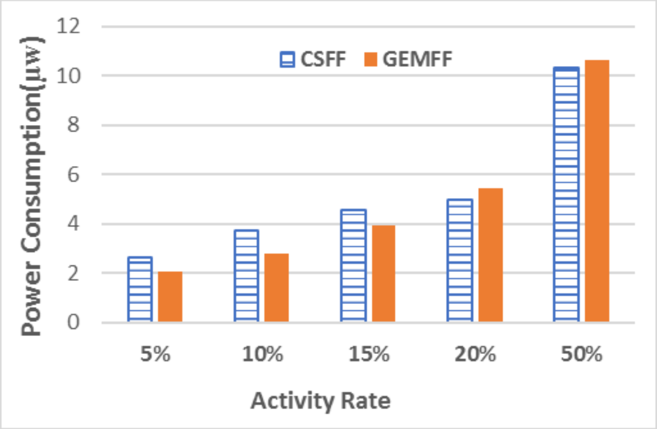


Figure 2. Power consumption under different switching activities: 5%- 50% @1.0V, 500 MHz

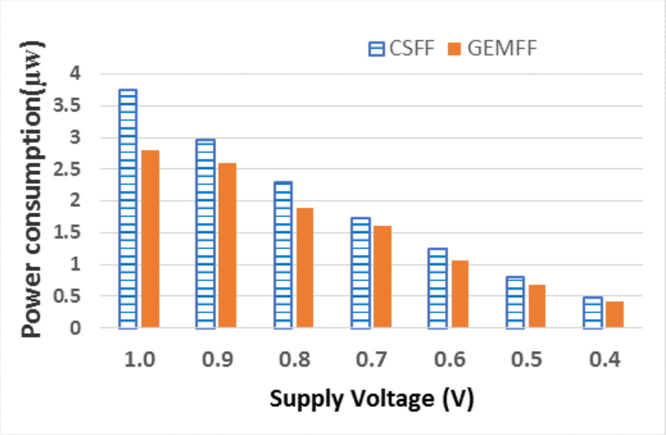


Figure 3. Power chart of different voltages 0.4V-1.0V at 10% switching activity, 500 MHz

4. Conclusions

In this paper, a novel flip-flop using clock gating circuitry with embedded XOR, GEMFF, is proposed. The new flip-flop uses clock gating to reduce redundant clock switching power. In view of power consumption, GEMFF outperforms CSFF by 25.1% at switching activity of 10%. Furthermore, at switching activities of 5% and 15%, GEMFF achieves less power consumption than CSFF as well. Hence GEMFF is applicable to achieve low power consumption.

Acknowledgements

This work is supported by grants from Microchip/Microsemi Inc, Dale and Sarah Ann Fowler, and Research fund from Office of Research, Chapman University.

REFERENCES

[1] J. L. Shin,et al.,(2013) “The next generation 64b SPARC core in a T4 SoC processor,” *IEEE Journal of Solid-State Circuits*, vol.48, no.1,82–90

[2] G. Gerosa (1994), “A 2.2W, 80 MHz superscalar RISC processor,” IEEE J. of Solid-State Circuits, vol. 29, no. 12, pp. 1440-1444.

[3] N. Kawai et al(2014)., “A fully static topologically-compressed 21-transistor flip-flop with 75% power saving,” IEEE Journal of Solid-State Circuits, vol. 49, no. 11, pp. 2526–2533.

[4] VAN LE, Loi, et al. (2018) “A 0.4-V, 0.138-fJ/Cycle Single-Phase-Clocking Redundant-Transition-Free 24T Flip-Flop With Change-Sensing Scheme in 40-nm CMOS,” *IEEE J. of Solid-State Circuits*, Vol. 99, pp. 1-12

[5] P. Zhao, Jason McNeely, Weidong Kuang, Zhongfeng Wang, (2011) “Design of sequential elements for low power clocking system” *IEEE Trans. VLSI Systems*, vol. 19, no. 5, 914 – 918

[6] M. Alioto, E.Consoli, G. Palumbo, (2011) “Analysis and comparison in the energy-delay-area domain of nanometer CMOS flip-flops: Part II–Results and figures of merit,” *IEEE Tran. VLSI*, 19(5), pp. 737-750

[7] P.Zhao, J.McNeely, (2007) “Low power clock branch sharing double-edge triggered flip-flop”, *IEEE Tran. VLSI Systems*, 15(3), pp. 338-345

[8] Y. Kim et al.(2014), “A static contention-flop in 45nm for low-power applications,” Papers, Feb. 2014, pp. 466–467.

HIERARCHICAL SCHEDULING FOR REAL-TIME PERIODIC TASKS IN SYMMETRIC MULTIPROCESSING

Tom Springer and Peiyi Zhao

Fowler School of Engineering, Chapman University, Orange, CA, USA

ABSTRACT

In this paper, we present a new hierarchical scheduling framework for periodic tasks in symmetric multiprocessor (SMP) platforms. Partitioned and global scheduling are the two main approaches used by SMP based systems where global scheduling is recommended for overall performance and partitioned scheduling is recommended for hard real-time performance. Our approach combines both the global and partitioned approaches of traditional SMP-based schedulers to provide hard real-time performance guarantees for critical tasks and improved response times for soft real-time tasks. Implemented as part of VxWorks, the results are confirmed using a real-time benchmark application, where response times were improved for soft real-time tasks while still providing hard real-time performance.

KEYWORDS

Real-time systems, hierarchical scheduling, symmetric multiprocessing, operating systems.

1. INTRODUCTION

The next generation embedded systems are working to consolidate large complex workloads onto multi-core platforms with mixed real-time applications. The existing architecture typically uses distributed uniprocessors connected over a common backplane where one processor may be assigned a soft real-time (SRT) task set and another processor a hard real-time (HRT) task set. The problem with this approach is it limits the computational throughput and increases costs as compared to multi-core platforms. It is for these reasons; designers are looking to re-host these new complex workloads onto multi-core platforms to reduce the size, weight and power (SWaP) requirements of traditional distributed systems.

Therefore, in this paper we look into symmetric multiprocessing (SMP) because most multi-core systems use SMP architecture. Briefly, SMP is a computing framework that manages the processing of tasks across multiple homogeneous processors or cores1 that share a common operating system, memory and I/O data path. One major challenge for SMP in mixed real-time scheduling is to effectively balance the competing needs of HRT and SRT tasks, such as temporal isolation, resource allocation or fault mitigation.

There are two main scheduling approaches for a SMP-based system: partitioned and global scheduling. Partitioned scheduling binds a task to a specific processor or core while global scheduling allows a task to migrate across multiple cores. Researchers have studied the

[1] Note that core and processor will be used interchangeably to indicate the basic computation unit of the CPU

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 91-111, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110810

schedulability of both approaches and have concluded that no single method dominates the other for all task sets [1]. Global scheduling provides better average case response times by performing load-balancing across multiple cores. However, the superior average case performance of global scheduling is not easily extended to hard real-time performance guarantees. For example, when performing load-balancing a global scheduler may migrate a task to another core and as a result invalidate the local cache. This invalidation process proves costly and can severely impact the determinism of the affected task.

On the other hand, partitioned scheduling statically assigns tasks to a specific core which can control task migration. Also known as CPU affinity, the idea is the designer can specify which tasks to run on a specific core then the scheduler obeys the order and only runs those tasks on the specified core. It also makes logical sense to bind all the tasks that access the same data to the same core(s) in this way they do not contend over data and ensure the task receives the full attention of the processor. However, when tasks are statically assigned to specific cores an unbalanced load distribution is likely to occur leading to a less than optimal utilization of the overall system.

Another concern involves the diversity and complexity of the various computational workloads in these next generation systems. Processing and criticality requirements may vary significantly where different operating modes could have vastly different workloads. In addition to the computational variations, mission critical type systems must perform continuously in harsh environments where they are expected to perform at least a subset of some critical functions under an overloaded or fault condition. The occurrence of an overload or fault must not hinder the overall survivability of the embedded system. Consequently, what is needed may be a more collective type of resource allocation where tasks are assigned resources according to their functionality requirements. In this way, applications can be grouped by service classes based upon their processing and criticality constraints.

Unfortunately, traditional SMP-based schedulers are not suitable to this type of collective resource allocation because they perform fine-grained scheduling at the task level. Since, these schedulers do not differentiate between tasks of different applications system-wide performance may not be the ideal metric for application specific requirements. Additionally, HRT and SRT tasks have competing objectives. HRT tasks require strict timing constraints where deadline misses are not tolerated. While SRT tasks can accept some deadlines misses but place a greater premium on task response time.

To solve these issues in this paper we present a new multi-core hierarchical scheduling framework (HSP) for periodic tasks in SMP-based systems. Our objective is to provide a hierarchical scheduling mechanism that can more effectively adapt to execution time variations in mixed real-time environments. Traditionally, the approach to scheduling mixed real-time applications has been to provide conservative WCET values to ensure the timing correctness of the HRT tasks. The problem with this approach is it usually leads to underutilized resources and poor response times because the actual WCET value of a task is rarely realized. As a result we look to exploit this underutilization by utilizing both the partitioned and non-partitioned scheduling mechanisms of a SMP-based system.

The benefits of this new scheduler are: (1) Better determinism for hard real-time tasks and improved response times for soft-real time tasks as compared to the global and partitioned scheduling methods of traditional SMP-based schedulers. (2) An application based resource allocation scheme which enhances scalability by reducing excessive interprocessor communication, bus contention and synchronization overhead. (3) A scheduling mechanism which provides for improved resource utilization and task acceptance rates. (4) Temporal isolation for hard real-time tasks where lower priority tasks cannot affect the timing behavior during overload or fault conditions.

The remainder of this paper is organized as follows. Section 2 provides an overview of the hierarchical scheduling framework used by our scheduling mechanism. Section 3 discusses previous work on hierarchical scheduling and SMP based scheduling mechanisms. Section 4 provides an overview of our hierarchical scheduler (HSP). Section 5 presents the schedulability analysis of our scheduler in a multicore environment. Section 6 utilizes task set simulations to provide comparisons between our hierarchical scheduling approach and the scheduling mechanisms for a traditional SMP-based scheduler. Section 7 describes the implementation of our hierarchical scheduling mechanism as an extension to Wind River's VxWorks RTOS and ported onto a commercially available multi-core processor. In Section 8 we conclude with future work and the research summary

2. Preliminaries

This section provides a discussion of the terminology used in the paper as well as an overview of hierarchical scheduling to provide as a reference for understanding the overall architecture of hierarchical scheduling in a symmetric multiprocessing environment.

2.1. Terminology

We consider a periodic task model defined as $\tau_i\left(T_i, C_i^{Lo}, C_i^{Hi}, D_i\right)$, where T_i is defined as the task period, C_i^{Lo} and C_i^{Hi} are defined as the average case execution time (ACET) and the worst case execution time (WCET) respectively and finally D_i is defined as the relative deadline. It is assumed that each task τ_i is a constrained task such that $C_i^{Hi} \leq D_i \leq T_i$. Each task τ_i must receive C_i^{Hi} within D_i or it is considered late. It is also assumed that C_i^{Hi} processor units are assigned to a task in a non-concurrent manner.

A subsystem (i.e. application) consists of a task set defined as a collection of periodic tasks $T_s = \{\tau_1, \tau_2, \dots \tau_n\}$. A system S consists of *n* homogenous processors while a subsystem consists of *m* processors such that $1 \leq m \leq n$. Each subsystem is characterized by a *multiprocessor resource* model [2] which specifies the resource supply provided to the subsystem (also known as a clustering). The *multiprocessor periodic resource* (MPR) model is defined as $\left(P_s, Q_{s,}, m_i\right)$, where Q_s provides the resource budget over P_s time units to a subsystem consisting of m_i processors. Therefore, a schedulable subsystem must meet the condition $Q_s \leq mP_s$.

In uniprocessor scheduling the supply bound function *(sbf)* is used to bound the supply required for schedulability of the subsystem. Authors in [2] extended this approach for hierarchical multiprocessor frameworks for deriving schedulability conditions of the subsystem. Therefore, the supply bound function for a multicore subsystem sbf_s is defined as:

$$sbf_s(t) = \begin{cases} kQ_s + max\{0, [I - kP_s]m + Q_s\}, \; t \geq P_s - \left\lceil \frac{Q_s}{m} \right\rceil \\ 0, Otherwise \end{cases} \tag{1}$$

where, $k = \left\lfloor \frac{t-(P_s - \left\lceil \frac{Q}{m} \right\rceil}{P_s} \right\rfloor$ and $I = t - 2P_s + \left\lceil \frac{Q_s}{m} \right\rceil$. Additionally, a lower bound of the sbf_s has been derived for improved schedulability. The lower bound supply lsb_s function is defined as:

$$lsb_s(t) = \frac{Q_s}{P_s}\left(t - 2\left(P_s - \frac{Q_s}{m}\right)\right) \tag{2}$$

The schedule for a subsystem that generates the resource supply in a time interval of $[0, t)$ is shown in Figure 1 along with the linear lower bound function. In Figure 2 we define $\alpha = \left\lfloor \frac{Q_s}{m} \right\rfloor$ and $\beta = Q_s - m\alpha$.

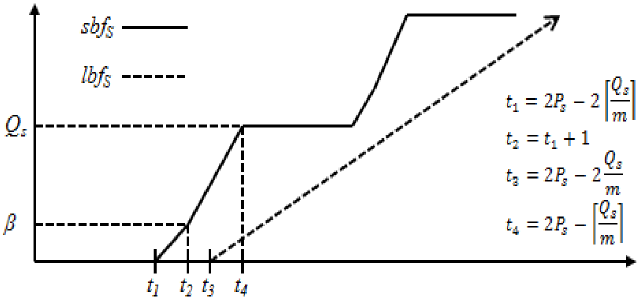


Figure 1: Supply bound and lower supply function for a subsystem

The MPR model presented by authors in [2] presents a framework that allows a subsystem exclusive access over a share of the multi-core platform. This share is then guaranteed by the sbf_s to provide a minimum resource supply to a subsystem. Therefore, HSP can utilize the MPR model to provide temporal isolation and schedulability guarantees between subsystems.

2.2. Hierarchical Scheduling

The basic framework of a hierarchically scheduled system [3] [4] for a uniprocessor platform is composed of multiple applications (subsystems) where each subsystem can be composed of a single or multiple tasks (see Figure 2). A *global* scheduler controls which subsystem is allocated the processor while the *local* scheduler determines which subsystem's task should actually execute

This two-level hierarchical scheduling approach is general enough in that it can be extended to a multiprocessor platform. In this case the scheduling of tasks within a subsystem, across m processors can be performed by the subsystem (local) scheduler while the scheduling of subsystems across the multiprocessor platform is performed by the system (global) scheduler. For example, consider a system where the overall utilization for each subsystem is $\sum_i^n \frac{C_i}{T_i}$ and $S_1 = 1.3, S_2 = 0.133 \; and \; S_3 = 1.122$ then the overall budget is 2.5 and m = 3, then the global scheduler will provide two units of resource from two processors and the remaining 0.5 units will be provided by the third processor.

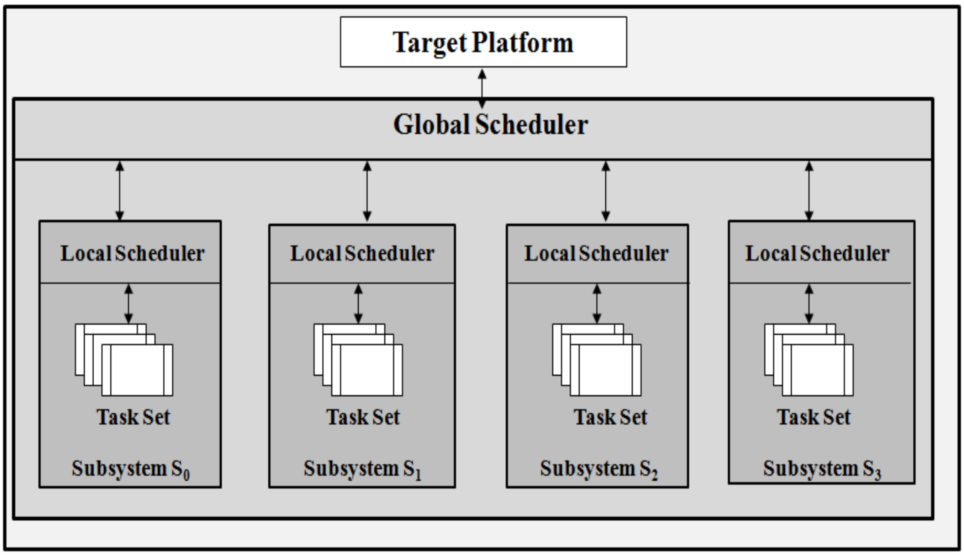


Figure 2: Hierarchical Scheduling Framework example

3. RELATED WORK

Initially an HSF was proposed by authors in [6] [8] as a means to perform composability analysis for open systems development. The motivation being that it can quickly become intractable to accurately verify the timing behavior of the embedded system as the complexity increases. The approach was to verify the timing behavior of each individual subsystem independently then compose each subsystem into the overall system.

A considerable amount of research has also been performed with hierarchical scheduling in a uniprocessor environment [4][7][9]. There has also been a fair amount of work in investigating how resources are shared across subsystems in an HSF [3][5][10]. However, there has not been a lot of work performed in actually applying a hierarchical scheduler to a multi-core environment. This lack of research is due in part to the fact that existing hierarchical scheduling algorithms are not easily extendable to multi-core environments. A couple reasons is that existing algorithms do not incorporate the inherent parallelism of a multi-core system and unfairness or task starvation can result if applied in a naïve manner.

Authors in [11] have presented a hierarchical multiprocessor algorithm known as H-SMP which was designed for a SMP-based platform. Their approach is to take a task set (i.e. an application) and assign it to the various cores in the subsystem based upon the application's level of parallelism and service requirements. Applications with higher service requirements would be allocated a higher bandwidth partition. For example, applications with soft real-time requirements would be receive a higher service level than applications with a best-effort type of service requirement. The primary limitation of this approach is that the CPU partitioning is done statically based upon a priori simulated workloads which may not represent real-world applications. In particular this static bandwidth partitioning may not achieve the best CPU partitioning for a dynamically changing workload. Another drawback is there is no explicit notion of criticality for adaptability to changing computational environments. In other words, tasks are assigned fixed budgets based upon their pre-determined WCET values where overly conservative WCET estimates could lead to system underutilization or higher task rejection rates.

-Additional work was done by authors [12][13] to provide a mixed-criticality scheduling framework for real-time operating systems (RTOS). Their approach was to use hierarchical scheduling to temporally isolate tasks of different criticality levels. A different scheduling algorithm was assigned to each criticality level. For example, tasks with the highest criticality were assigned a cyclic executive scheduler while less critical tasks were assigned other schedulers like earliest deadline first (EDF). Temporal isolation is enforced by a server with a specific budget which is statically assigned to each critically level.

There has also been some work done [14][15][16] in semi-partitioned scheduling in multiprocessors. The idea is that some tasks are assigned according to the partitioned scheduling approach while other tasks are assigned by global scheduling and therefore allowed to migrate. In order to determine how tasks are assigned the authors took a look at the task workload and then tried to assign that tasks to processors accordingly. For example, tasks with a high workload (i.e. high utilization factor $U_i = \frac{C_i}{T_i}$) would be partitioned while tasks with a low workload would be scheduled globally. Other approaches have looked at how to assign tasks to reduce cache misses [17] by using partitioned scheduling for the task most likely to generate a high number of cache invalidations. The main limitation with these approaches are that the processor assignments are done a priori with no real notion of criticality for HRT or SRT tasks to adapt to computational changes, such as task overloads.

In our work we take an adaptive approach where non-critical resources are assigned dynamically based upon environmental changes. Instead of static partitioning tasks are allocated based upon a feedback mechanism that the scheduler uses to adjust resource allocation to more effectively adapt to diverse computational workloads at run time. In order to support a service level requirement approach like H-SMP tasks are guaranteed a certain budget but are allowed to share any unused budget by employing capacity sharing mechanisms. A type of capacity sharing algorithm, known as slack stealing [24] is used which allows a lower-priority task to share the bandwidth of a higher priority task. In this way critical functions can be guaranteed a certain level of service but any unused resource can then be re-allocated to task with a lower service level thereby improving the performance, such as reduced response times, of the lower priority task.

4. HSP ALGORITHM DESCRIPTION

This section provides an overview of the HSP scheduling framework which is used to more effectively manage HRT and SRT tasks on a symmetric multiprocessing platform. Our approach employs a two-level hierarchical scheduled framework (see Figure 3) to provide resource partitioning and temporal isolation between subsystems. Additionally, HSP utilizes elements of both the partitioned and global scheduling approaches to maximize the benefits of both scheduling mechanisms.

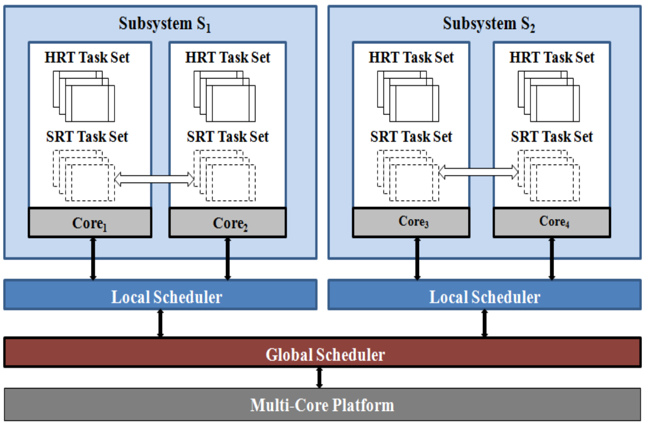


Figure 3: Hierarchical Scheduling for Multicore Processor

However, unlike uniprocessor based hierarchical scheduling SMP-based hierarchical scheduling needs to contend with tasks that can be stationary or migratory. In order to account for this added complication SMP-based hierarchical scheduling requires enhanced functionality which includes: processor assignment, task set schedulability analysis and run-time scheduling. Processor assignment is the algorithm that determines how an application is assigned to the various processors allocated by the subsystem. The tasks that comprise an application are assigned to processors based upon a combination of mixed-criticality scheduling and semi-partitioned scheduling. The schedulability analysis determines whether the HRT/SRT task set is schedulable on a specific processor. Run-time scheduling determines when tasks execute as well as manage when a task should migrate to another idle core in the subsystem.

4.1. Processor Assignment

HSP like other traditional partitioned scheduling approaches assigns each task to a particular processor based upon some type of bin-packing heuristics. HRT tasks with strict timing constraints are assigned to a specific core first according to the chosen heuristic and if the schedulability condition can be satisfied for that core. In this way HRT tasks can get the full attention of the processor and improve the deterministic behavior of the task. Consider Table 1 that defines a task set for the example Subsystem1 depicted in Figure 3. According to Table 1 tasks that are partitioned (p) are considered HRT tasks are statically assigned to a specific core and not allowed to migrate. Tasks that are global (g) are considered SRT tasks and allowed to migrate across cores in the subsystem. This is similar to mixed-criticality scheduling that assigns highly critical tasks to specific cores but allows less critical tasks to migrate.

Table 1: Example subsystem task set

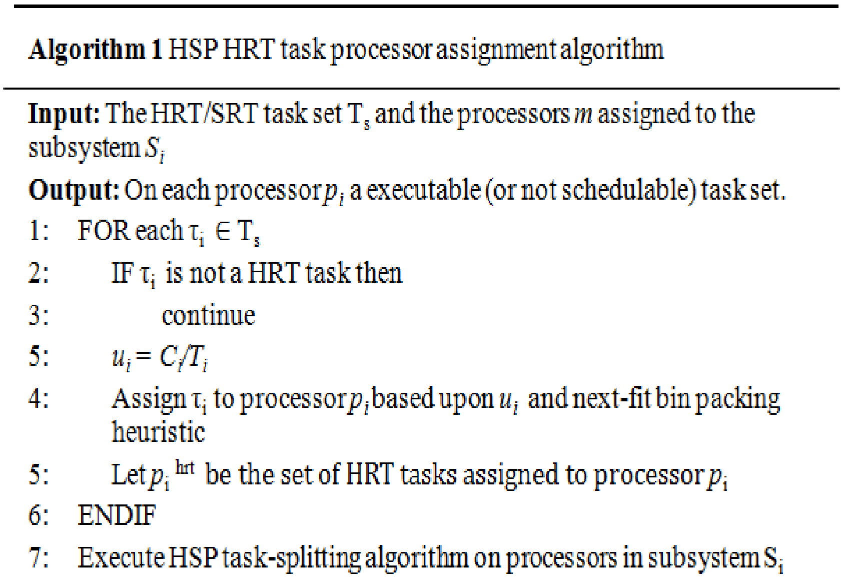
Task	Core	Ti	C_i^{Lo}	C_i^{Hi}	Di
τ1	p	5	1	2	5
τ2	p	10	2	4	10
τ3	p	15	1	3	15
τ4	p	20	2	5	20
τ5	g	15	1	3	15
τ6	g	20	2	4	20
τ7	g	25	2	5	25

For the purpose of schedulability guarantees the HRT tasks are allocated a budget, by the hierarchical scheduler, equal to the task's WCET value (C_i^{Hi}), in this way tasks are guaranteed a fixed processing time by the subsystem's local scheduler. The HRT tasks are assigned to a core based upon the next-fit bin-packing heuristic and since the rate monotonic (RM) algorithm is optimal for fixed priority scheduling it is used as the determination of schedulability for partitioned tasks (see Algorithm 1). Therefore, the maximum utilization U_{si} for a core in a subsystem as defined by RM is:

$$U_{si} = \sum_{i=1}^{n} \frac{C_i^{Hi}}{T_i} \leq n\left(2^{1/n} - 1\right) \qquad (3)$$

From the example task set shown in Table 1 and the multi-core system depicted in Figure 3 the HRT tasks would be assigned a particular core as illustrated in Figure 4.

Algorithm 1: HRT Task assignment algorithm



```
Algorithm 1 HSP HRT task processor assignment algorithm

Input: The HRT/SRT task set $T_s$ and the processors $m$ assigned to the subsystem $S_i$
Output: On each processor $p_i$ a executable (or not schedulable) task set.
FOR each $\tau_i \in T_s$
    IF $\tau_i$ is not a HRT task then
        continue
    $u_i = C_i/T_i$
    Assign $\tau_i$ to processor $p_i$ based upon $u_i$ and next-fit bin packing heuristic
    Let $p_i^{hrt}$ be the set of HRT tasks assigned to processor $p_i$
ENDIF
Execute HSP task-splitting algorithm on processors in subsystem $S_i$
```

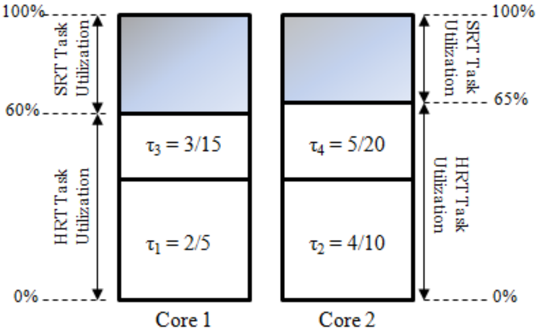
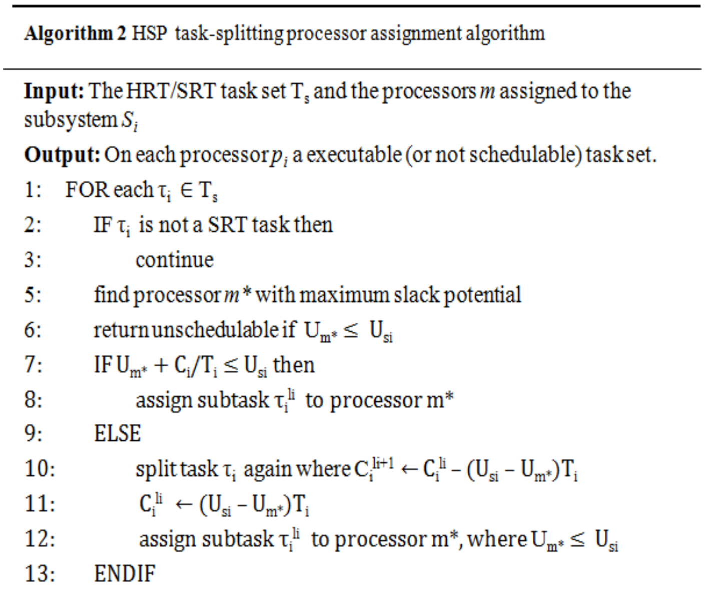


Figure 4: Partitioned task core assignments

After the HRT tasks are assigned to their respective cores the SRT tasks are assigned based upon the remaining resource capacity. If the SRT task does not fit onto a particular core to support the full execution capacity then the task is split across cores in the subsystem. Task splitting is based upon semi-partitioned scheduling which is defined as a task τ_i that is executed on l_i processors where$l_i \geq 2$. There are l_i subtasks denoted by$\tau_i^1, \tau_i^2, \ldots, \tau_i^{l_i}$, which are synchronized where no subtasks can run in parallel and each subtask τ_i^j has a computation time C_i^j such that$C_i = \sum_{j=1}^{j-1} C_i^j$. The algorithm for splitting a task τ_i is provided in Algorithm 2.

Algorithm 2: Task splitting assignment algorithm



Algorithm 2 HSP task-splitting processor assignment algorithm

```
Input: The HRT/SRT task set $T_s$ and the processors $m$ assigned to the subsystem $S_i$
Output: On each processor $p_i$ a executable (or not schedulable) task set.
FOR each $\tau_i \in T_s$
    IF $\tau_i$ is not a SRT task then
        continue
    find processor $m*$ with maximum slack potential
    return unschedulable if $U_{m*} \leq U_{si}$
    IF $U_{m*} + C_i/T_i \leq U_{si}$ then
        assign subtask $\tau_i^{li}$ to processor m*
    ELSE
        split task $\tau_i$ again where $C_i^{li+1} \leftarrow C_i^{li} - (U_{si} - U_{m*})T_i$
        $C_i^{li} \leftarrow (U_{si} - U_{m*})T_i$
        assign subtask $\tau_i^{li}$ to processor m*, where $U_{m*} \leq U_{si}$
    ENDIF
```

Consider the example provided below of how a task may be split across more than one processor. To help identify the core(s) with the maximum slack time potential for SRT task processor assignment.

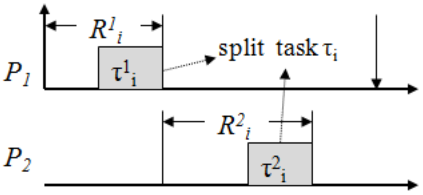


Figure 4: Split task across two processors

While Figure 5 illustrates how a split task could be split it does not describe the criteria used to assign the split tasks to the various processors in the subsystem. Traditional approaches have been to assign each share to processors with subsequent indexes so that τ_i^1 would be assigned to P_1 and τ_i^2 would be assigned to P_2. With semi-partitioned scheduling most tasks are assigned to a particular processor to reduce overhead while the remaining tasks are split to improve schedulability. The problem with this approach is there is no real notion of criticality and tasks are assigned to a processor based upon their respective WCET values which are typically conservative. Our approach with HSP is different in that task criticality is considered by assigning HRT tasks first ensuring that the tasks will be fixed to a particular processor thereby reducing runtime overhead. The schedulability is maintained for the SRT tasks by performing task-splitting and task response times are improved by taking advantage of the potential unused processing capacity, also known as slack. This slack potential is then used by HSP for processor assignment of SRT tasks. SRT tasks whether they requiring splitting or not are then assigned to available cores based upon the maximum slack potential for that core. Note that this slack potential is determined not by the WCET of a HRT task but rather by their average execution time denoted by C_i^{Lo}. In this way the maximum potential can be identified which represents a much less conservative calculation for improving task response times. The set of algorithms for identifying slack and taking advantage of it is known as slack stealing. A brief overview of slack stealing is provided in the subsection below; for more detail readers are encouraged to review the references.

4.1.1. Slack Stealing

According to Equation (3) a task set that meets the criteria will always make its deadlines. The problem is this criterion is based upon WCET values which are usually conservative calculations and there tends to be a large gap between the WCET value and the actual processing time of a HRT task. This gap, known as slack, presents an opportunity to minimize the response times of a SRT task. Authors in [24][25] describe how the slack is found by mapping out the processor schedule of the HRT tasks over their hyper-period in a task mapping table. The table is then examined to determine the slack present between the deadline and the next invocation of the task. In turn, this table is then examined by HSP to help identify the core(s) with the maximum slack time potential for SRT task processor assignment.

4.2. Task Scheduling

The local scheduler of a subsystem in HSP is responsible for scheduling of tasks on the various cores of the subsystem. Scheduling for the HRT tasks are straightforward in that traditional scheduling mechanisms, such as RM, where the priorities of each task are assigned so that:〖 τ〗_4< τ_3< τ_2<τ_1. Similar to HRT tasks priorities are assigned according to the RM except SRT tasks always have a lower priority than HRT tasks, such that SRT < HRT, except during slack stealing periods. During periods of slack stealing the SRT task is temporarily promoted to the same priority level as the HRT task that finished with some available slack time. In this way another HRT task of lower priority cannot preempt a SRT task while it is stealing the slack of another HRT task.

During run-time after a HRT task completes the local scheduler looks to exploit the slack time of an HRT tasks to improve a SRT task's response time. The run-time slack of a HRT task τ_i is based upon the budget (C_i^{Hi}) of task τ_i provided by the subsystem's S_i local scheduler. The task's budget for the subsystem's S_i local scheduler of a HRT task along with the feedback from the task provides the information needed to determine if there is any potential slack available to the SRT tasks. In order to calculate the slack at some arbitrary time t we look at the unused server budget of an HRT task in the interval$[t, t + D_i(t))$. Therefore, the slack is determined by the length of that interval less than the actual unused budget available from all of the HRT tasks that fall into that interval. The slack is defined as s_i(t) = $\sum_{\forall j \in hrt(i)} (Q_j - c_j)$ that is available to any SRT task at some arbitrary time t and c_j is the actual processing time of the HRT task. As an example consider the example task set in Table 1. Figure 6 represents the tasks scheduled on the first core while Figure 7 represents the tasks scheduled on the second core. The up arrow represents task start time and the down arrow represents the task completion time.

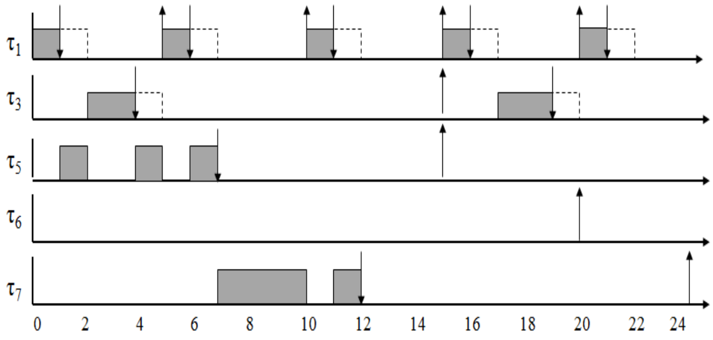


Figure 5: Core 1 task schedule

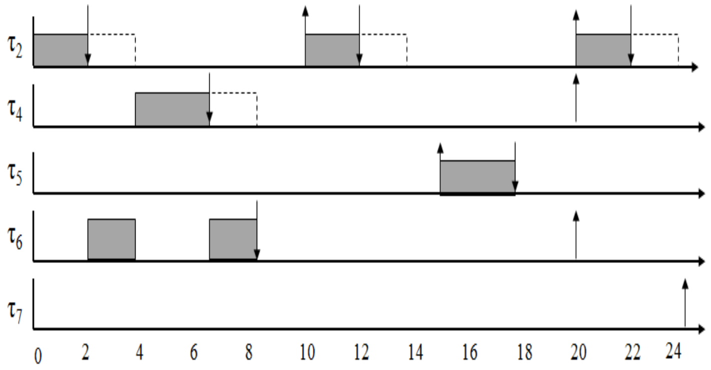


Figure 6: Core 2 task schedule

The HRT task set is statically assigned to a core and based upon the next-fit bin-packing heuristic tasks τ_1 and τ_3 are assigned core 1 while tasks τ_2 and τ_4 are assigned core 2. The highest priority SRT task τ_5 if the first task scheduled to run on either core when there is available processing or slack time. At time t1 task τ_5 is allowed to run by stealing the slack from task τ_1 but at time t2 is preempted by the HRT task τ_3. Task τ_5 is then allowed to steal slack from task τ_3 at time t4 and from task τ_1 then complete execution by time t7.

5. Schedulability Analysis

With the HSP all tasks execute up to their worst case execution time C_i^{Hi} but the local scheduler prevents the tasks from executing any further. If a task executes further than C_i^{Hi} it is considered in fault and aborted or considered overloaded and rescheduled until it is safe to be executed again. This section presents the response time analysis for HSP as it relates to partitioned and non-partitioned scheduling.

As mentioned in Section 4.1 the tasks are scheduled by a fixed priority preemptive scheduler and the task priorities are assigned according to the RM algorithm. Priority (p) is derived from the deadlines of the tasks, such that for any two tasks τ_i and τ_j their deadlines $D_i < D_j \Rightarrow p_i > p_j$. To test for schedulability, the standard Response Time Analysis (RTA) [19] [20] for uniprocessor scheduling can be extended to HSP. RTA first computes the worst-case completion time for each task (i.e. response time R_i) and then compares that value to the task deadline, such that $R_i \leq D_i$ for task τ_i. The response time value is calculated using recurrence relations:

$$R_i = C_i + \sum_{\tau_j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \tag{4}$$

where *hp(i)* defines the set of tasks with a higher priority than the task τ_i. The general response time Equation (4) can then be applied to mixed critically systems [12] where the LO-criticality and HI-criticality mode schedulability can be verified. HSP can then adapt this analysis and apply it to HRT tasks which are considered HI-criticality and SRT tasks which are considered LO-criticality. Standard RTA for a uniprocessor can be applied for SRT tasks as follows:

$$R_i^{Lo} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^{Lo}}{T_j} \right\rceil C_j^{Lo} \tag{5}$$

where *hp(i)* is the set of SRT tasks with a higher priority than task τ_i. The same analysis can also be applied to HRT tasks as follows:

$$R_i^{Hi} = C_i + \sum_{j \in hpH(i)} \left\lceil \frac{R_i^{Hi}}{T_j} \right\rceil C_j^{Hi} \tag{6}$$

where *hpH(i)* is the set of HRT tasks with a higher priority than task τ_i. For uniprocessor based systems the schedulability test is determined by calculating the response times of all tasks in an interval starting with a critical instant (case where all tasks experience their WCET) and comparing that to the task deadlines. However it has been shown [20] that it is a NP-hard problem when analyzing globally scheduled periodic tasks. The issue is that it is not easy to find a "representative" interval to represent the start of the critical instant. As a result, in a multicore system only sufficient results can be determined in a reasonable amount of time. Authors in [22] provide a sufficient RTA-based approach for schedulability tests for global scheduled multicore systems. The test is based upon the RTA test of Equation (4) and operates as follows:

$$R_i^{max} \leftarrow C_i + \frac{1}{m} \sum_{\tau_j \in hp(i)} \left(\left\lceil \frac{R_j^{max}}{T_j} \right\rceil C_j + C_j \right) \tag{7}$$

The schedulability analysis for semi-partitioned systems can then be derived by combing equation (4) and equation (7). To determine the schedulability for SRT and HRT tasks using average case execution time:

$$R_i^{Lo} \leftarrow C_i^{Lo} + \frac{1}{m} \left(SRT\left(\tau_i^{Lo}\right) + HRT\left(\tau_i^{Lo}\right) \right) \tag{8}$$

where $SRT\left(\tau_i^{Lo}\right)$ represents the SRT task set average execution times such that:

$$SRT\left(\tau_i^{Lo}\right) = \sum_{\tau_j \in hp(i)} \left\lceil \frac{R_i^{Lo}}{T_j} \right\rceil C_j^{Lo} + C_j^{Lo} \tag{9}$$

And $HRT\left(\tau_i^{Lo}\right)$ represents the HRT task set average execution times such that:

$$HRT\left(\tau_i^{Lo}\right) = \sum_{\tau_j \in hpH(i)} \left\lceil \frac{R_i^{Lo}}{T_j} \right\rceil C_j^{Lo} \tag{10}$$

where *hpH(i)* is the set of HRT tasks that are assigned to processor P_i. Additionally, to determine the schedulability for SRT and HRT tasks using worst case execution time:

$$R_i^{Hi} \leftarrow C_i^{Hi} + \frac{1}{m}\left(SRT\left(\tau_i^{Hi}\right) + HRT\left(\tau_i^{Hi}\right)\right) \qquad (11)$$

$$SRT\left(\tau_i^{Hi}\right) = \sum_{\tau_j \in hp(i)} \left\lceil \frac{R_i^{Hi}}{T_j} \right\rceil C_j^{Hi} + C_j^{Hi} \qquad (12)$$

$$HRT\left(\tau_i^{Hi}\right) = \sum_{\tau_j \in hpH(i)} \left\lceil \frac{R_i^{Hi}}{T_j} \right\rceil C_j^{Hi} \qquad (13)$$

Consider the task set represented by Table 1 in Section 4.1 the schedulability analysis for both SRT and HRT would be as follows.

Table 2: Example Task Set with Response Times

Task	Core	Ti	C_i^{Lo}	C_i^{Hi}	Di	R_i^{Lo}	R_i^{Hi}
τ1	p	5	1	2	5	1	2
τ2	p	10	2	4	10	2	4
τ3	p	15	1	3	15	2	5
τ4	p	20	2	5	20	4	9
τ5	g	15	1	3	15	4	15
τ6	g	20	2	4	20	7	29
τ7	g	20	2	5	25	8	58

6. PERFORMANCE ANALYSIS

For the purpose of comparisons, we used a combined SRT/HRT periodic task set that comprised a single subsystem (i.e. application) and spanned up to *m* cores, where *m* = 2, 4, 8. Task periods (p_i) were chosen using a uniform random distribution from the list {0.25Hz, 0.5Hz, 1Hz, 2 Hz, 4Hz, 5Hz, 8Hz, 10Hz, 20Hz, 25Hz, 50Hz, 100Hz, 200Hz}. The list was created to represent some typical rates of periodic tasks. Overall system utilization (u_{sys}) for each processor ranged from [0.50, 1.00] in increments of 0.05. Individual task utilization (u_i) was randomly generated with an expected value of 0.20 and a standard deviation of 0.15. The number of tasks in the set were determined by the summation of the individual tasks where $\sum_{i=0}^{n} u_i = u_{sys}$. The execution time (c_i) was calculated based upon the task period and task utilization such that $c_i = p_i * u_i$. The HRT/SRT tasks were randomly divided from the generated task set with an expected value of $n/2$ and a standard deviation of *n-2*.

HSP was compared against four other semi-partitioning algorithms used in mixed-criticality systems, DU-RM, DU-Audsley [26], DC-RM and DC-Audsley. Each algorithm, including HSP utilizes the next-fit bin packing heuristic but differ on processor and priority assignment. The DU-RM algorithm decreasingly assigns tasks based upon the task utilization and determines feasibility based upon the RM scheduler. In other words the task with the highest utilization factor is assigned to the first available processor. DU-Audsley is similar to DU-RM except Audsley's priority assignment is optimal for a given processor but the complexity is much higher than RM assignment. The DC-RM algorithm performs processor assignment based upon the decreasing criticality of a task so HRT tasks would be assigned to a processor before a SRT task. DC-Audsley also performs processor assignment based upon the task criticality but its priority assignment is different than DC-RM. Our approach with like DC-RM and DC-Audsley assigns a task based upon criticality but differs in that if there is not enough available utilization HSP will spilt tasks across any available processors. This has the potential to significantly improve schedulability.

For the simulations we generated 10,000 task sets from the parameters described in the previous paragraph. The task sets were determined to be schedulable if every task in the set was successfully assigned to the group of cores defined by the subsystem S_i. The performance criteria for the processor assignment algorithm was determined by the success ratio of the number of tasks scheduled by the number of submitted tasks accepted, defined as follows:

$$\frac{number\ of\ scheduable\ task\ sets}{number\ of\ attempted\ task\ sets}$$

The overall subsystem utilization was determined by $u_{sys} * m$, so that 1.0, 2.0 and 4.0 represents 50% utilization for $m = 2, 4, 8$ respectively. The data in Figures 8, 9 and 10 illustrates the results from u_{sys} = [0.5, 1.0] where HSP clearly provides better schedulability than the other processor assignment algorithms. Note that the other algorithms start to report failure around 0.5 to 0.7 while HSP does not start to report failure until close to 0.7 to 0.8. This coincides with other work [21][23] that states maximum schedulability for RM or DM is about 88% for uniprocessors. Also notice that HSP outperforms the other algorithms as the number of cores increase because this provides HSP the opportunity to share more of the computation across the various cores in the subsystem.

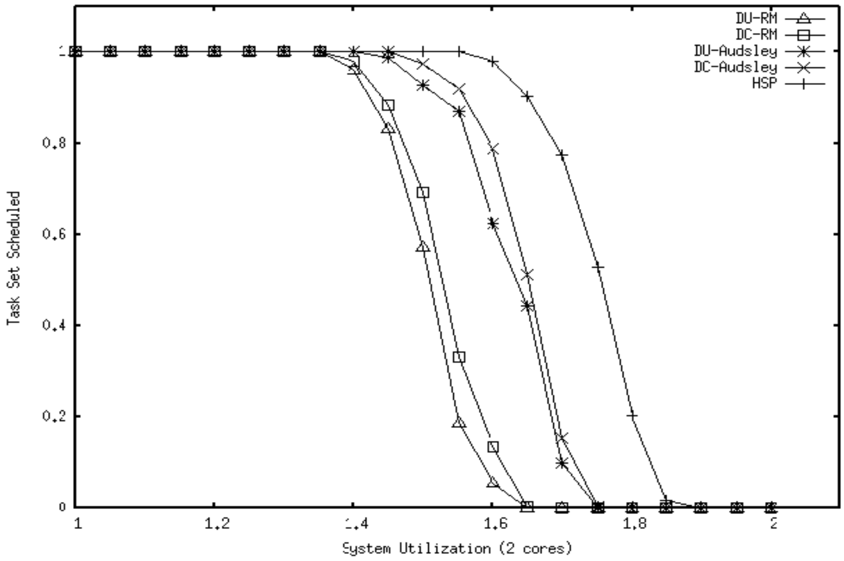


Figure 7: Task Set simulation 2 cores

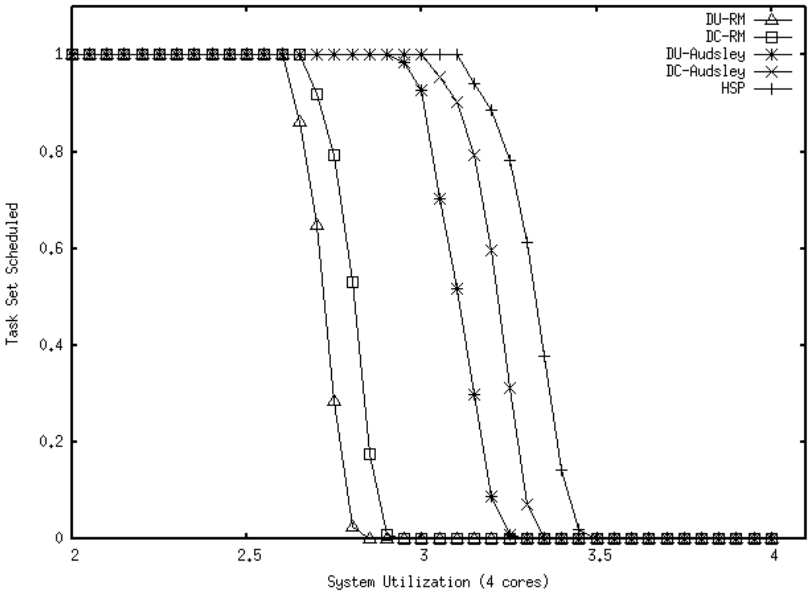


Figure 8: Task Set simulation 4 cores

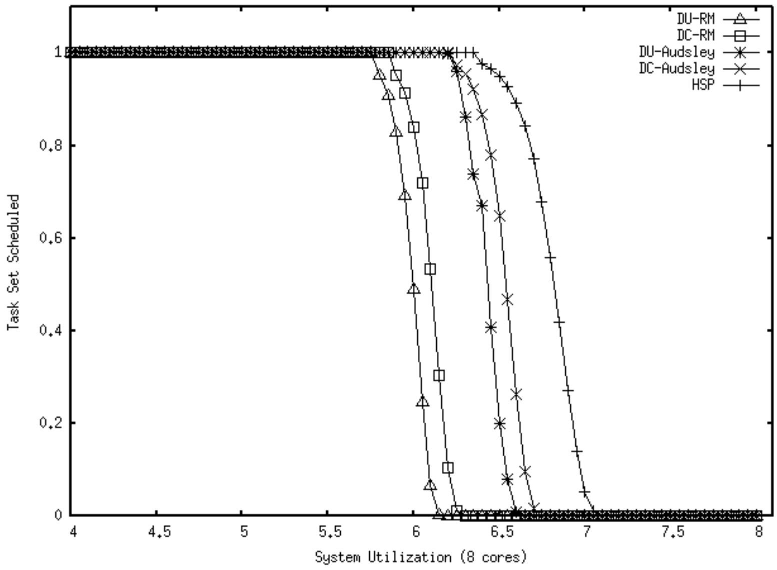


Figure 9: Task Set simulation 8 cores

7. IMPLEMENTATION

This section defines the design and implementation of HSP in the VxWorks real-time operating system (RTOS). The work is based upon the architecture presented in [27] and extended to work in a SMP-based platform.

7.1. Local Scheduler Implementation

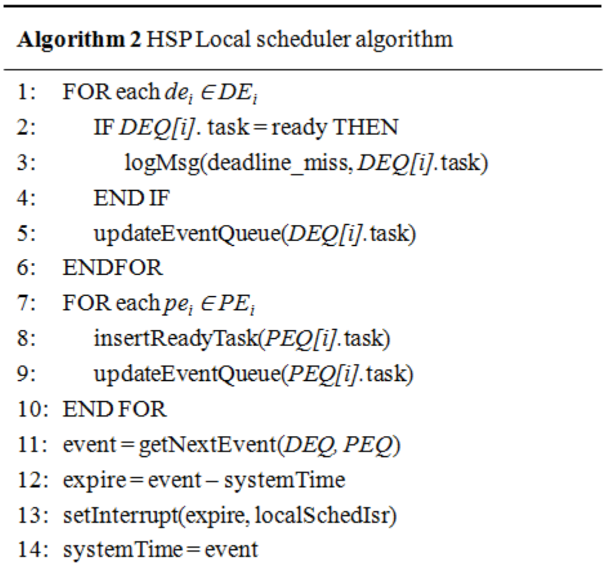
The native VxWorks scheduler can schedule tasks using either a preemptive priority based or a round-robin scheduling policy. In VxWorks 6.x and greater Wind River introduced the concept of real-time processes (RTP) which more closely resemble processes in general purpose operating systems like Linux. Tasks in kernel mode or processes in RTP mode are scheduled in the same way. Processes are created with memory protection so kernel memory space, ISRs and direct hardware access are prohibited. Tasks that operate in kernel mode have full access to kernel resources and are not subject to the same limitations as processes in RTP mode

We choose to implement HSP in kernel mode because the overhead in RTPs are prohibitive and HSP needs access to the kernel resources for task management. HSP was implemented on top of the native VxWorks scheduler as a type of extension or middleware that sits between the hierarchical scheduler and the VxWorks native scheduler. The VxWorks RTOS provides functions to extend the capability so various kernel mechanism can be customized to support HSP. For example, the scheduler can be extended with either a customized ready queue structure or to attach an interrupt handler that is executed at every clock tick.

The native VxWorks scheduler dispatches the highest priority task in the ready queue. Our approach utilizes the system call *tickAnnounceHookAdd( )* that is invoked at every tick interrupt and called before the native scheduler accesses the ready queue to dispatch the highest priority task. The ready queue is then manipulated by resuming a task *taskResume( )*, suspending a task *taskSuspend( )* or setting/changing priorities *taskPrioritySet( )*. The kernel's tick counter is also utilized to read *tickGet( )* and set *tickSet( )* as a means to manage the notion of time when the tick interrupt ISR is invoked.

The primary function of the local scheduler is to arrange tasks in the ready queue at every period start, in effect extend the VxWorks scheduler to support periodic tasks. The local scheduler is implemented as part of a custom ISR that is attached with the *tickAnnounceAdd( )* system call. The system call routines mentioned previously are then called to change the status of the task or to change task priorities. The native VxWorks scheduler is then invoked to perform the necessary functions (i.e. context switching) to dispatch the task on the appropriate processor. The pseudo code listed in Algorithm 3 below provides an overview of the local scheduler.

Algorithm 3: Local scheduler algorithm



Algorithm 2 HSP Local scheduler algorithm

```
1:  FOR each $de_i \in DE_i$
2:      IF $DEQ[i]$. task = ready THEN
3:          logMsg(deadline_miss, $DEQ[i]$.task)
4:      END IF
5:      updateEventQueue($DEQ[i]$.task)
6:  ENDFOR
7:  FOR each $pe_i \in PE_i$
8:      insertReadyTask($PEQ[i]$.task)
9:      updateEventQueue($PEQ[i]$.task)
10: END FOR
11: event = getNextEvent($DEQ$, $PEQ$)
12: expire = event – systemTime
13: setInterrupt(expire, localSchedIsr)
14: systemTime = event
```

The first step of the algorithm is to check if the task is still in the ready queue (lines 2-4) the then the deadline event queue (DEQ) is updated (line 5) to track the task deadlines. At each period start tasks are inserted into the ready queue (7-8). Tasks deadlines and periods are updated in the periodic event queue (PEQ). The next event is then updated by extracting the closet deadline/period from event queue (lines 11-12). The interrupt is set at the next event and the local system counter is updated (lines 12-14).

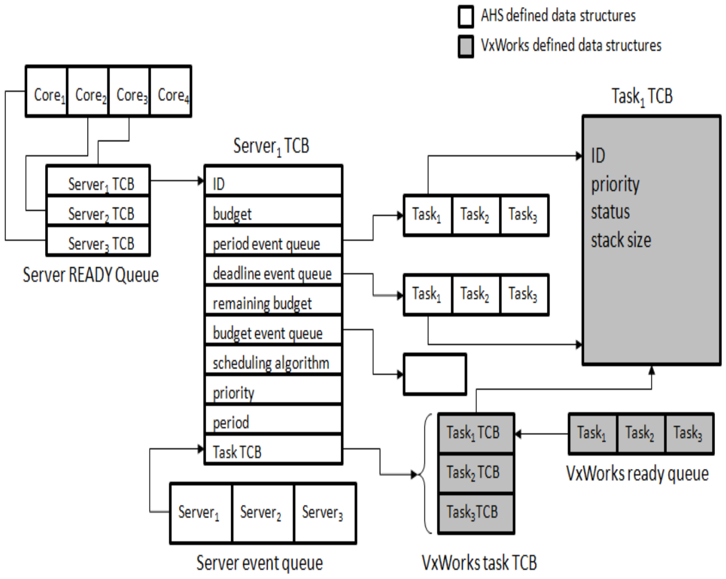


Figure 10: HSP Implementation in VxWorks

7.2. Global Scheduler Implementation

Global scheduling is used to implement the notion of servers in a hierarchical scheduled system. The global scheduler is responsible for managing all the events in the system which can include subsystem events, server events and server budget events. The global scheduler itself is a task in VxWorks with its own task control block (TCB) and task event queue. Figure 11 below illustrates the implementation of the required data structures to support global scheduling in HSP for VxWorks.

The TCBs needed to support global scheduling in VxWorks are described in the list below.
ID is a unique number associated with each server.

period_event_queue is a reference to the server's event queue which contains the task period.
period is the period of the server.

deadline_event_queue is a reference to the server's task queue which holds the task deadline.
budget is the server defined budget.

remaining_budget is the current remaining budget of the server.

priority is the server's priority.

scheduling algorithm is the server's local scheduling algorithm.

Task_TCB is a list to the VxWorks TCB task list. It references those task TCB's that are associated with the server.

7.3. Hardware Platform

HSP was implemented as described in the previous section with VxWorks 6.9 on a Freescale T4240: QorIQ 12 core (24 virtual-core) communications processor.

For evaluation purposes we ported the SNU Real-Time Benchmark Suite [18] and compared response times and overall system utilization using partitioned, non-partitioned (global) and hierarchical scheduling. The SNU real-time benchmark suite contains small C programs used for worst-case execution time analysis. This benchmark was chosen because it is completely structured (no unconditional jumps, no loop body exits,), no switch or do-while statements and no library calls or specific systems calls. The programs are mostly numeric and DSP algorithms.
In order to represent the periodic task model of an embedded system a subset of the programs in the benchmark suite were chosen and assigned arbitrary task rates (see Table 3).

Table 3: Simulated Periodic Task Set

C Program	Task	Rate	C_i^{Lo}	C_i^{Hi}
matmul	τ_1	50Hz	1.7ms	5.1ms
fft1	τ_2	40Hz	2.7ms	5.4ms
fir	τ_3	20Hz	10.4ms	20.8ms
lms	τ_4	10Hz	12.6ms	25.2ms
ludcmp	τ_5	40Hz	6.8ms	13.6ms
minver	τ_6	10Hz	3.5ms	10.5ms
qsort-exam	τ_7	5Hz	2.2ms	11.0ms

The tasks sets were assigned as HRT = $\{\tau_1, \tau_2, \tau_3, \tau_4\}$ and SRT = $\{\tau_5, \tau_6, \tau_7\}$. The HRT/SRT task sets comprised a single subsystem S_1 which was allocated two cores in the hierarchical system. The HRT/SRT task sets were conceived so that if the C_i^{Hi} value for each SRT task was realized then the task set is not schedulable and an overload condition would result. In order to evaluate the effectiveness of HSP the execution times of the overall task sets were increased from [0.00, 1.00], where 0.0 indicates all tasks are executed at their respective C_i^{Lo} levels and 1.0 indicates all tasks are executed at their respective C_i^{Hi} levels. The task response times were measured by the high resolution counter/timer used as part of the timestamp mechanism by WindRiver’s System Viewer application. Table 3 was used to represent their respective average case and worst case execution times for each task in the set.

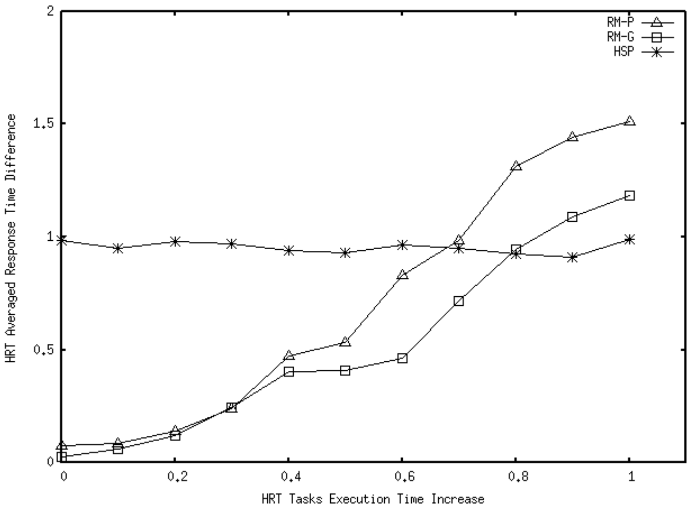


Figure 11: HRT Task Set Response Time Average

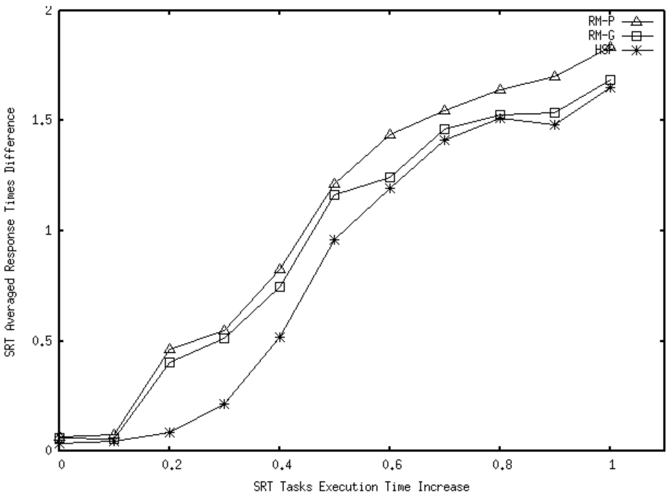


Figure 12: SRT Task Set Response Time Average

Figure 12 represents the measured response times of the HRT task set. To represent each individual task would create an overly crowded graph so the individual task response times were normalized and then averaged over the whole task set. Specifically each task response time was recorded then compared to the respective task's estimated response time. Let the actual task response time be defined as R_i^{Act}, the estimated lower bound response time is R_i^{Lo}, the upper bound response times is R_i^{Hi} so that the averaged response time difference is defined as:

$$\Delta i = \frac{\left(R_i^{Hi} - R_{Hi}^{Lo}\right) - R_i^{Act}}{\left(R_i^{Hi} - R_i^{Lo}\right)}$$

then the total task set response time average is defined as the average of all Δ_i for the HRT task set. What this means is a value of 0.0 indicates the measured task response times were at or near their respective R_i^{Lo} values and a value of 1.0 indicates R_i^{Hi} values. A value greater than 1.0 signifies that one or more tasks exceeded their deadline. Notice that for HSP the response time difference hover around 1.0 this is because the local scheduler does not allow other HRT tasks to execute before a higher priority task C_i^{Hi} execution time. Therefore, before the system starts to become overloaded around 0.6 the response times for both the partitioned method (RM-P) and the non-partitioned method (RM-G) outperform those of HSP. Recall, this is an acceptable situation because with HRT tasks we are less concerned about response times as we are with HRT timing constraints. Note, that at times 0.6 to 0.7 both RM-P and RM-G methods start to exceed 1.0 which indicates that tasks in the HRT set are beginning to experience deadline misses while with HSP no HRT tasks experience deadline misses.

The SRT task set performance is illustrated in Figure 13. Notice that early on before the system becomes overloaded from 0.0 to 0.4 HSP clearly outperforms both the RM-P and RM-G methods. This is because the HSP is able to take advantage of the slack generated by the HRT task set. Once the system starts to become overloaded at 0.5 HSP starts to converge to RM-G because there is no longer any available slack time. Both the RM-G and the HSP methods outperform RM-P because they are allowed to migrate across the cores in the subsystem.

8. CONCLUSIONS/FUTURE WORK

In this paper we considered the problem of how to assign and schedule HRT and SRT tasks in a symmetric multiprocessor environment to more effectively adapt to environmental changes. Those changes such as unexpected computational workload deviation were managed by hierarchical scheduling to provide the temporal isolation between tasks. The efficient assigning and scheduling of processors was accomplished by combining mixed-criticality and semi-partitioned scheduling. The result was demonstrated improvement of response times for SRT tasks and schedulability guarantees for HRT tasks where no deadlines were missed during periods of overload. As further confirmation for the validity of this approach we also implemented HSP as part of the VxWorks RTOS.

Future work includes evaluating the additional overhead HSP incurs in VxWorks as compared to traditional scheduling. Additionally, tasks as well as task sets are considered to be completely independent with no shared resources. A more practical implementation would include HSP scheduled tasks or subsystems that would have to share a mutual resource such as a semaphore.

REFERENCES

[1] J. Carpenter, S. Frank, P. Holman, A. Srinivasan, J. Anderson and S. Baruah. A categorization of real-time multiprocessor scheduling problems and algorithms. Handbook of Scheduling: Algorithms, Models and Performance Analysis. CRC Press LLC, 2003.

[2] I. Shin; A. Easwaran, I. Lee. Hierarchical Scheduling Framework for Virtual Clustering of Multiprocessors. Real-Time Systems, 2008. ECRTS '08. Euromicro Conference on, vol., no., pp.181,190, 2-4 July 2008.

[3] R.I. Davis and A. Burns. Resource Sharing in Hierarchical Fixed Priority Pre-emptive Systems. In RTSS'06.

[4] P. Goyal, X. Guo and H.M. Vin. A hierarchical CPU scheduler for multimedia operating systems. In OSDI, pp. 107-121, 1996

[5] N. Fisher, M. Bertogna and S. Baraugh. The Design of an EDF-Scheduled Resource-Sharing Open Environment. In RTSS '07.

[6] T-W. Kuo, C-H. Li. A Fixed Priority Driven Open Environment for Real-Time Applications. In Proc. of IEEE Real-Time Systems Symposium, 1999, pp. 256-267.

[7] R.I. Davis and A. Burns. Hierarchical Fixed Priority Pre-emptive Scheduling. Dept. Comp. Sci. Univ of York, 05.

[8] Z. Deng and J. W.-S. Liu. Scheduling real-time applications in an open environment. In Proc. of IEEE Real-Time Systems Symp, 1997, pp. 308–319.

[9] G. Lipari and S.K. Baraugh. Efficient scheduling of real-time multi-task applications in dynamic systems. In Proc. 6th IEEE Real-Time Technol. Appl. Symp. (RTAS'00), pp166-175.

[10] M. Behnam, T. Nolte, M Sjodin and I Shin. SIRAP: A synchronization protocol for hierarchical resource sharing real-time open systems. In Proc. 7th ACM and IEEE Int. Conf. Embedded Software (EM-SOFT 07).

[11] A. Chandra, P. Shenoy. Hierarchical Scheduling for Symmetric Multiprocessor. In IEEE Trans. on Parallel and Distributed Systems. 2013.

[12] M. Mollison, J. Erickson, J. Anderson, S. Baruah, J. Scoredos. Mixed-Criticality Real-Time Scheduling for Multicore System. IEEE (CIT 2010).

[13] J. Herman, C. Kenna, M. Mollison, J. Anderson. D. Johnson, RTOS Support for Multicore Mixed-Criticality Systems. (RTAS 2012)

[14] O. Kelly, H. Aydin, B. Zhao, On Partitioned Scheduling of Fixed-Priority Mixed Criticality Task Set. (TrustCom 2011)

[15] S. Kato, N. Yamasaki. Semi-Partitioned Fixed-Priority Scheduling on Multiprocessor. (RTAS 2009).

[16] S. Kato, N. Yamasaki, Y. Ishikawa. Semi-Partitioned Scheduling of Sporadic Task Systems on Mulitprocessor. (ECRTS 2009)

[17] B. Andersson, J. Jonsson. Fixed-priority preemptive multiprocessor scheduling: to partition or not to partition. (RTSA 2000)

[18] SNU Real-Time Benchmark, http://www.cprover.org

[19] M. Joseph and P. Pandya. Finding response times in a real-time systems. BCS Computer Journal pp390-396, 2009

[20] N. Audsley, A. Burns, M. Richardson, K. Tindell, A. Wellings. Applying new scheduling theory to static priority preemptive scheduling. Software Engineering Journal, pp284-292, 1993

[21] L. Papalau, P. Samalik. Design of an Efficient Resource Kernel for Consumer Devices, Stan Ackermans Institute, Eindhoven University if Technology, Eindhoven, Holland 2000.

[22] M. Bertogna, M. Cirinei. Response-Time Analysis for globally scheduled Symmetric Multiprocessor Platforms, RTSS 2007.

[23] J. Lehoczky, L. Sha, Y. Ding, The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior, IEEE Real-Time Systems Symp. 1989.

[24] R. Davis, K. Tindell, A. Burns, Scheduling Slack Tine in Fixed-Priority Pre-emptive Systems. In Proc. Real-Time Systems Symp. 1993.

[25] U., José M., J. Orozco, and R. Cayssials. Fast Slack Stealing methods for Embedded Real Time Systems. 26th IEEE International Real-Time Systems Symposium (RTSS 2005)-Work In Progress Session. 2005.

[26] N. Audsley. Optimal priority assignment and feasibility of static priority tasks with arbitrary start times. Technical Report, The University of York, 1991.

[27] M. Behnam, T. Nolte, I. Shin, M. Asberg. Towards Hierarchical Schedling in VxWorks. OSPERT 2008, Proc. of the Fourth International Workshop on Operating Systems Platforms for Embedded Real-Time Applications. 2008

ENSEMBLE MODEL FOR CHUNKING

Nilamadhaba Mohapatra, Namrata Sarraf and Swapna sarit Sahu

Department of Data Science, Zeotap, Bangalore, India

ABSTRACT

Transformer Models have taken over most of the Natural language Inference tasks. In recent times they have proved to beat several benchmarks. Chunking means splitting the sentences into tokens and then grouping them in a meaningful way. Chunking is a task that has gradually moved from POS tag-based statistical models to neural nets using Language models such as LSTM, Bidirectional LSTMs, attention models, etc. Deep neural net Models are deployed indirectly for classifying tokens as different tags defined under Named Recognition Tasks. Later these tags are used in conjunction with pointer frameworks for the final chunking task. In our paper, we propose an Ensemble Model using a fine-tuned Transformer Model and a recurrent neural network model together to predict tags and chunk substructures of a sentence. We analyzed the shortcomings of the transformer models in predicting different tags and then trained the BILSTM+CNN accordingly to compensate for the same.

KEYWORDS

Natural Language Processing- Named Entity Recognition, Chunking, Recurrent Neural networks, Transformer Model.

1. INTRODUCTION

Chunking is the process of splitting the words of a sentence into tokens and then grouping the tokens in a meaningful way. These chunks are our point of interest which are used to solve our relevant NLP tasks[3]. It labels every word of the sentence suitably and thus lays out a basic framework for bigger tasks such as question answering, information extraction, topic modeling, etc[16]. Named Entity Recognition as mentioned in the paper[2][10][11]is the process of extracting and tagging words that signify the names of certain places, people, organizations, time, etc. Several Natural Language Understanding Tasks like POS tagging, Tokenization, and Noun Phrase Identification are all chunking tasks.

In recent times Chunking has also been used in various domain-level projects with an end goal of retrieving custom substructures of a sentence[18]. We have employed the chunking model proposed in a similar use case. Since its wide use and application chunking requires a constant push to the state of the art models. We have hence proposed a model where we use RoBERTa[9], a Transformer Model ensemble with an RNN based BILSTM+CNN [8]The Transformer Model helps in attention maximization and hence expanding the learning abilities of the Model. It provides embeddings for words in a sentence highly correlated with other words in the sentence owing to its multi-head attention mechanism[19], and thus reflects a more relative semantic of a word instead of using just the POS tag for embeddings.[9] We have used pre-trained models that facilitate the understanding of words and their place from a much larger dataset and fine-tuning it further customizes and improves the model understanding of words related to specific tags associated with it.

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA, NLCAI - 2021
pp. 113-119, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110811

In our Recurrent Neural Network section of the Model, we use Bidirectional LSTM in addition to Convolutional Neural Networks CNN[8]. We introduced this segment of the model to compensate for the shortcomings of the Transformer Model and give a more general approach to Tagging and segmentation for chunking. Since LSTMs use the sequential word by word approach of processing, they can be slower in processing, and hence we use simpler shorter custom embeddings targeting exactly the common error points of Transformer Models[9]. The word embedding essentially utilizes a combination of positional embedding, POS Tag, and the word vector after whitespace tokenization. Our analysis of the RoBERTa Model showed an error in differentiating similarly tagged tokens and hence a boost to the Part of speech tagging is given here. The BiLSTM network is complemented with a stack of CNN layers to aid feature extraction from sentences and the feedback loop helps in accurately labelling these features and further segmenting them into chunks.

2. PRIOR WORK

Named Entity Recognition was initially performed using extensive knowledge base systems, its orthographic features, ontological and lexicon rules[4][2][14][15]. However, the new trend has shifted towards neural network-based structures to define entity relations [5][13]. Hence the mentioned top 20 state-of-the-art NER mechanisms are neural network-based including LSTM, GRUs, BERT, CNN, and a combination of them as suited. [2] [6][10][11][12].

Chunking has been done using machine learning-based models such as HMM(Hidden Markov Model) [7][17] and Maximum Entropy model and has gradually seen a shift to Statistical models such as Support Vector Machines and Boosting [8], [3], [7]. In more recent times, Neural Models have been on a rise as a tool for chunking. Neural network models are deployed as a classification system to classify Beginning, Inside, and Outside of the chunks required or also known as BIO tagging which is quite a popular Named Entity Recognition mechanism for segmentation. The latest paper submitted by IBM Watson uses a combination of Bi-LSTM and CNN to label the tokens of the sentence and then chunk them together accordingly [8]. They follow an encoder-decoder-pointer framework while segmenting and labelling chunks sequentially using a pointer. We use POS tag reinforced word vectors as input to this segment of the model. Since the model excelles majorly in sequential labelling and feature extraction, it helps widen the gap between similar tags occurring together frequently.

In this paper, we have deployed an ensemble model using the Transformer-based Model RoBERTa [9] and recurrent neural network[8] for labeling and segmentation, after which we group the hence labeled chunks and map the contexts and phrases together. Our experiment gives an F1 score 97.3 which beats the F1 score of the chunking methods employed in the paper by IBM 94.72. [8] both tested on the common CoNLL 2000 dataset. F1 score here signifies the harmonic mean of precision and recall derived from the confusion matrix built on test dataset to evaluate the classifier trained for entity tag labels which are then further used for chunking.

3. DATA

We have evaluated our Model on two main datasets, English Penn treebank[20] corpus and ConLL 2000 Dataset. The English Penn Treebank corpus, is an extensively used corpus for the evaluation of language models. The task includes annotating each word with its Part-of-Speech tag. There are 38,219 sentences, used for training, 5,527 sentences used for validation and 5,462 sentences are used for Testing purposes. ConLL 2000[21] is a widely used dataset for noun phrase chunking with 211727 training tokens and 47377 testing tokens. The annotation of the data is procured from the WSJ corpus by an automated program written by Sabine Buchholz from

Tilburg University. The objective of this task is to introduce machine learning methods which after training recognizes the chunk segmentation of the test data as accurately as possible.

4. EXPERIMENTS

The Named entity task in our experiment is to classify tokens into IOB annotations of either custom tags, POS tags or Entity tags for example organisation, person etc. These tags are then used for segmentation using the Inside and Outside tags and give a final output of chunked phrases of a sentence. The chunks are made as accurately as the tags are predicted.

Table 1. Comparative Analysis of Algorithms. Dataset (ConLL 2000 english).

Model	CONLL 2000
ROBERTA +(BI-LSTM+CNN)	97.3
ROBERTA	96.76
BERT	95.64
BI-LSTM+CNN	94.72

4.1. MODEL I

We deployed two Models to train and test for the Named Entity Recognition task.

The neural model we chose to train our NER downstream task was RoBERTa. This model is a robustly optimized form of BERT(Bidirectional Encoder Representations from Transformers) and is state of the art on 4 out of 9 GLUE (General Language Understanding Evaluation) tasks. The model is a modified version of BERT wherein it targets the flaws of BERT. BERT is pre-trained over 16GB of uncompressed text, which is a combination of Book Corpus[22] and English Wikipedia[23]. However, RoBERTa is trained on five English language corpora summing up to 160GB of uncompressed text. The huge upscale of data for pretraining gives RoBERTa the edge over BERT. In addition, RoBERTa omits the Next Sentence Prediction task from the pretraining, trains on longer sequences, and dynamically revises the masking patterns on training data [12].

We chose a Transformer Model as our neural architecture for generalization to bring in the concept of attention to the structure.[9] In the downstream task of Named Entity recognition, we need the complete semantics of the sentence. For this purpose, the attention weight of every token or time step for every other token present in the sentence is essential.

RoBERTa is consistent with the mathematics of BERT as well as the number of training layers, attention heads, and parameters. For our task, we used the RoBERTa base model which has 12 Layers of deep neural architecture, 16 Attention heads, and 110 training parameters. The model was fine-tuned to train for 6 epochs with a training batch size of 64 and a learning rate of $2*e^{-5}$. The model pre-trained for the Named Entity recognition task is imported.

4.2. MODEL II

The second model is an ensemble training of RoBERTa and Bi-LSTM + CNN.

The purpose of ensemble training is to boost the learning process of neural architecture. We fed different feature vectors into Model A and Model B to capture the entire meaning of a token. We deploy this model to further increase our F1 score by using two complementary training models.

RoBERTa uses its encoder to tokenize and embed the tokens into feature vectors. In the Bi-LSTM+CNN model, we use a custom tokenizer after analysing the misclassifications of Model I. In our analysis, we found that similar types of Parts of speech e.g.: Symbols, and Punctuations, and Nouns and Proper nouns were getting miss classified. To rectify this error, we built a custom tokenizer and embedding explained in the later sections.

4.2.1. MODEL A

RoBERTa, Refer Figure 1, model takes care of attention and co-dependency of tokens frequently coming together in a sentence. It uses RoBERTa Tokeniser and RoBERTa Embedding to encode the entire sentence and is then trained over for 6 epochs. We have explained the purpose of using RoBERTa in the earlier sections.

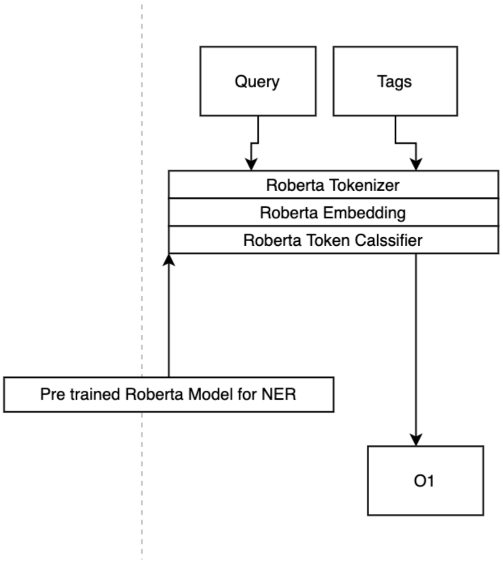


Figure 1. Transfor Model: RoBERTa: This model produces Output 1 or O1

4.2.2. MODEL B

The second leg of the ensemble training, Refer Figure 2, uses a custom tokenizer wherein every query is whitespace tokenizer, every token is passed into a word vectorizer and horizontally stacked with it's one hot encoded part of speech tag. This array is then multiplied with positional encoding of the token which finally outputs the token's embedding. We have used three features of a token namely its word semantics, position at which it shows up in the sentence and it's part of Speech in this Tokenizer which makes up for any misclassification that is brought about by Model A.

$$\textit{word embedding} = \textit{positional encoding}*([\textit{word vector, ohe pos tag}]) \quad \text{...Eqn.1}$$

where,

positional encoding = position of token in query / length of query

ohe pos tag = one hot encoded part of speech tag.

These word embeddings are sent to the BiLSTM layer followed by 1 dimensional convolution, max pooling and the vector is then flattened out before introducing non linearity and classification on top of it.

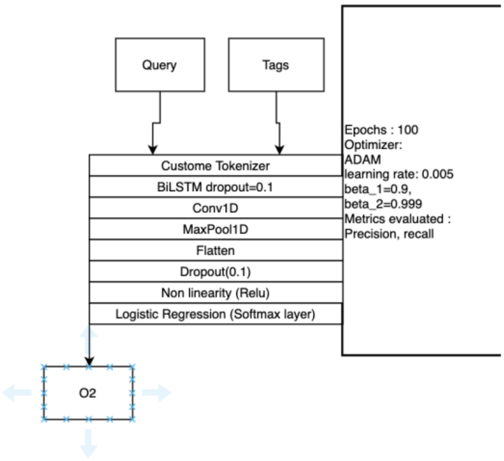


Figure 2. BiLSTM+CNN: This model produces Output 2 or O2 after the query goes through a custom tokenizer and embedder to further pass through the RNN and CNN layers

4.2.3. Ensemble

Weighted Average of Classification scores of Model A and Model B are then used for final Multi class classification.

We trained Model II for 150 epochs using the Adam Optimiser and a learning rate of 0.01. The model converged to an F1 score of 97.3 which is higher than either of the models described individually. We can thus conclude with this experiment that ensemble training using custom tokenizer encapsulates more information required for a Named Entity Recognition Task. The same segmentation and labelling approach is used here as Model I.

$$\textit{final classification scores} = w1*O1 + w2*O2 \quad \text{.... Eqn. 2}$$

where,
O1, O2 are outputs from Model A and Model B respectively.
w1, w2 are the weights associated with it.

Table 2. Comparative Analysis of Algorithms. Dataset (PENN Treebank).

Model	**CoNLL 2000**
RoBERTa +(Bi-LSTM+CNN)	98.4
IntNet + BiLSTM-CRF	95.29
NCRF++	95.06

5. Observations, Results, Benchmarks

The training sentences are preprocessed into a [token]-[tag]-[sentence-id] format for training. Every query is whitespace tokenized, every token hence obtained is mapped to its annotated tag and a sentence id corresponding to the query. A GPU Nvidia Tesla K80 is used for training which takes about 160 minutes per epoch. Naturally, for this kind of task the evaluation metric F1 score is taken into account because the data set has imbalanced tags and accuracy might therefore be biased towards the majority tag.

After training the model, a layer of segmentation is deployed to chunk the phrases and contexts together to further pass onto the last layer of the model to map the relevant context and phrases together. The result of the model hence is a dictionary where key, value pairs are the context phrase(s) pairs.

Comparing our results with the present state of the art for chunking[10], on conll 2000 data set Refer Table 1, the model exceeds the state of the art neural net model which uses Bi-LSTM in conjunction with CNN for encoder-decoder labeling and pointer framework for segmentation as mentioned before. Our model obtains an F1 score of 97.3 exceeding the preceding State of the Art 94.72 for conll 2000 dataset. The comparison is also done on 23 labels for 9000 training sentences and 900 testing sentences. On the Penn treebank dataset, Refer Table 2, our model achieves an F1 score of 98.4 for classifying the chunking tags which exceeds the present state of the art[8] with a score 95.29 which uses IntNet + BiLSTM-CRF..

6. CONCLUSIONS

In this paper, we aim to highlight the significance of the ensemble models. In the first base model, we used the current state-of-the-art model i.e. the transformer-based model. It uses an attention framework to understand the semantics of long-length sentences better. We found that the transformer-based model does the misclassifications at various Symbols, and Punctuations, and Nouns and Proper nouns. To compensate for this in the second base model, we explicitly provided POS tags of the token explicitly to the model. We used a BILSTM+CNN framework with explicit pos tags for each token. The CNN part optimizes the local context well and The BILSTM with pos tag tries to capture the context dependency with the surrounding word better. We experimented with this ensemble model on various open datasets for chunking tasks and found that this ensemble architecture broke the previous state-of-the-art. In future work, we would apply this architecture to different GLUE tasks to achieve similar kinds of results.

REFERENCES

[1] Nothman And James R. Curran And Tara Murphy, “Transforming Wikipedia Into Named Entity Training Data,” 2008.

[2] V. Yadav and S. Bethard, “A survey on recent advances in named Entity Recognition from deep learning models,” arXiv [cs.CL], 2019.

[3] E. Muszyńska, “Graph- and surface-level sentence chunking,” in Proceedings of the ACL 2016 Student Research Workshop, 2016.

[4] A. Siddharthan, "Complex Lexico-syntactic Reformulation of Sentences Using Typed Dependency Representations", ACL Anthology, 2021. [Online]. Available: https://www.aclweb.org/anthology/W10-4213.

[5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” arXiv [cs.CL], 2016

[6] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” in Benjamins Current Topics, Amsterdam: John Benjamins Publishing Company, 2009, pp. 3–28.

[7] T. Kudo and Y. Matsumoto, “Chunking with Support Vector Machines,” J. Nat. Lang. Process., vol. 9, no. 5, pp. 3–21, 2002.

[8] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, “Neural Models for Sequence Chunking,” arXiv [cs.CL], 2017.

[9] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv [cs.CL], 2019.

[10] R. Chalapathy, E. Zare Borzeshi, and M. Piccardi, “An investigation of recurrent neural architectures for drug name recognition,” in Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis, 2016.

[11] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in Proceedings of the 25th international conference on Machine learning - ICML ’08, 2008.

[12] F. Dernoncourt, J. Y. Lee, and P. Szolovits, “NeuroNER: an easy-to-use program for named-entity recognition based on neural networks,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2017..

[13] R. Panchendrarajan and A. Amaresan, "Bidirectional LSTM-CRF for Named Entity Recognition", ACL Anthology, 2021. [Online]. Available: https://www.aclweb.org/anthology/Y18-1061.

[14] Y. Li, K. Bontcheva, and H. Cunningham, “SVM based learning system for information extraction,” in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 319–339.

[15] W. Etaiwi, A. Awajan, and D. Suleiman, “Statistical Arabic name entity recognition approaches: A survey,” Procedia Comput. Sci., vol. 113, pp. 57–64, 2017.

[16] E. J. Otoo, D. Rotem, and S. Seshadri, “Optimal chunking of large multidimensional arrays for data warehousing,” in Proceedings of the ACM tenth international workshop on Data warehousing and OLAP - DOLAP ’07, 2007.

[17] H. Sharma, "Survey of Research on Chunking Techniques", Semanticscholar.org, 2021. [Online]. Available: https://www.semanticscholar.org/paper/Survey-of-Research-on-Chunking-Techniques-Sharma/ced929651b222d3b489df1c25ed9c801c7c3e749.

[18] P. S. Rosenbloom and J. Aasman, “Knowledge level and inductive uses of chunking (EBL),” in Soar: A Cognitive Architecture in Perspective, Dordrecht: Springer Netherlands, 1992, pp. 219–234.

[19] A.Vaswani et al., “Attention is all you need,” arXiv [cs.CL], 2017.

[20] M. Marcus, B. Santorini and M. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank", ACL Anthology, 2021. [Online]. Available: https://www.aclweb.org/anthology/J93-2004.

[21] E. F. Tjong Kim Sang and S. Buchholz, “Introduction to the CoNLL-2000 shared task: Chunking,” in Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning -, 2000.

[22] Y. Zhu et al., “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[23] Wikipedia: Database download

AUTHOR INDEX