Technische Universität Berlin

Extending Secure Data Storage in Decentralized Networks

by

Youngrak Ryu

Matriculation Number 380241

Technische Universität Berlin
School IV - Electrical Engineering and Computer Science
Department of Telecommunication Systems
Service-centric Networking

Master's Thesis Outline

April 4, 2020

supervisor:
Dirk Thatmann

SNET
SERVICE-CENTRIC NETWORKING

Extending Secure Data Storage in Decentralized Networks

1 State of the Art

Most legacy network systems are composed as centralized architecture, in which the main system afford services to connected users. When users request any information, the frontend application takes over requests of the user and brings data from the connected server or backend application. This manner takes advantage of development and management. However, as the growing scale of the system, the centralized network encounters difficulties, that are a weakness of outside attack, e.g., network connection instability, DDoS, storage cost with rising scale, and sensitive data management. In contrast to the centralized network system, the decentralized network system has become a big trend to avoid the disadvantages of the centralized network. Distributed Ledger Technology is one of the decentralized network technologies, and is composed of nodes, which verify with the cryptographical identifier, then interconnect using the peer-to-peer network, and transmit data between nodes. The system to avoid data modification uses consensus algorithms for reliable nodes. Therefore nodes can be synchronized. A distributed hash table (DHT) [1] is a lookup service using key and value or list paired. Using encrypted key DHT can get exact values. With distinguish DHT characters, more use cases are extended (e.g., e-Wallet, Digital Identification, Distributed File System, and Peer-to-Peer file sharing). The decentralized storage to store data uses the distributed file system, e.g., InterPlanetary File System (IPFS) [2][3], BitTorrent File System (BTFS)[4], Ethereum Swarm, and BigchainDB. The distributed file systems are characterized by sharing fragmented data as a chunk that is stored in nodes. IP based communication using TCP and UDP has protocol-level issues[5][6]. Ethernet MTU allows the limitation for a maximum of 1500 bytes per packet. Therefore transmission systems set a limit to about 1000 bytes without the header space for system reliability[7]. Thus, Ståx, which is based on a Kademlia DHT[8], that afford only UDP, consists of various interconnecting types of nodes, and the nodes transmit data through the HTTP-level P2P tunnel using UDP. When nodes make the transmission of mass data such as image and music, there still exist some protocol-level issues. To enable secure decentralized data storage requires comparison and analyzation between other systems. Furthermore, I would like to develop a framework with research findings, which will be applied to Ståx's Kademlia DHT.

2 Problem

Nodes should interconnect using DHT, and nodes make a connection via P2P tunnel to enable data storage in decentralized networks. As establish P2P, between nodes have to verify credentials. Self-Sovereign Identity (SSI)[9] will be applied in Ståx and uses the Decentralized identifier (DID)[10]. The framework will also implement SSI using DID. Transmission protocols such as TCP and UDP have the limitation of the size of a single packet, which is not able to send the over 1500 bytes packet. When the client transfers to another node data such as an image file, the transmission problem can occur. Because of this issue, the transmission uses Packet Fragmentation, which splits packets into smaller pieces, and then fragmented packets are sending to avoid MTU. Fragmented packets are reassembled at the received host. However, the packet fragmentation also appears technical problems, which are packet filtering at the Firewall or NAT router, fragmented packets can be lost in the transmission, and the implementation in UDP of packet fragmentation is difficult. Ståx is

the Kademlia DHT using UDP based system. Ståx's Kademlia DHT offers MTU size for UDP up to 1000 bytes. Thus, Ståx can appear such problems.

- As lost datagram has happened, the protocol requires a checksum process.
- As a node is corrupted, the asynchronous error can occur.

Thus, the latency can increase with the transmission of mass data. With these problems, the interconnected nodes have trouble synchronizing data. The nodes cannot trust each other, and the reliability problem may arise from unsynchronized data. Because of this reason, most DHTs are only used to store smaller amounts of data. The Ståx system is feasibly designed to interconnect between various platforms, e.g., PC, smartphone, and IoT. The framework will be developed into cross-compile based. IoTs are limited environment and offer minimized resource. Therefore the framework has to be compressed and not to be heavy.

3 Approach

The goal is to enable secured data storage in decentralized networks. The framework will be applied as proof of concept to the Ståx system. In order to improve the framework, other existing decentralized data storages need to be investigated. In the investigation, distinguish characters will be discovered and will compare to Ståx. Furthermore, the developed framework would like to make a trial of using IPFS and BTFS, and then with the findings, the prototype will be applied. The prototype using the Ståx system will be developed by ANSI C language for supporting extending various platforms such as IoT devices. For testbed, an environment will be configured using Docker-composed. Virtual nodes will be implemented using Docker-composed, and then the prototype will make tests on the environment.

4 Challenge

The most important challenge is supporting IoT devices. Nodes of different types should be feasible, connecting each other. The growth of IoT would be the main node platform in the decentralized network. Numerous IoT device nodes will deploy on the network, thus developing the cross-compiling framework is a significant assignment. For reaching the requirement framework will be Ansi C implementation and composed scale for using limited IoT resources as well. Another challenge is the efficient data transmission between nodes. For solving this problem, Kademlia DHT of Ståx should improve to transmit the bigger size of a packet. However, it still remains problems, e.g., latency and asynchronous error. When reducing existing overhead, the packet can be optimized. If as possible, the prototype can combine an extending Kademlia DHT using TCP and UDP-lite[11].This solution may solve the limitation problem of packet size. When the problems are solved, the decentralized network can extend transmittable feasible data types, e.g., image, the large amount of the dataset, and music, over the existing limitation.

5 Evaluation

The completely developed prototype will be evaluated by demanded procedures. The main challenge is supporting IoT. Therefore Interconnecting between different platforms is the most important valuation basis. Also, other platforms can be implemented by different languages such as Python,

Go, or Javascript. Thus the prototype will make an interoperability test. To evaluate the prototype progresses a simulation of transmission mass data distribution. In the simulation, the between nodes will read or write data through the findings that can be analyzed as pros and cons. The evaluation should analyze the comparison of the implemented prototype between other systems such as IPFS, BTFS, and standard Kademlia by measurement of transmission rate, error rate, latency, propagation speed, and reliability. Through the user test, the developed system can verify more issues. Discovered new challenges or developable problems can be further work.

6 Schedule

- 1/2 month Research of decentralized network systems and related work
- 1/2 month Design the prototype using Ståx
- 1/2 month Implement the Ståx and other systems
- 2 month Develop and implement the prototype, and if there is enough time, develop the extending prototype using other protocols
- 1/2 month Simulation and evaluation
- 2 month Write thesis

References

[1] S. Sivaraja, M. Thiyagarajah, T. Piranavam, C.-H. Lung, and S. Majumdar, "Efficient multiple-keyword search in dht-based decentralized systems," in *2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*. IEEE, 2008, pp. 406–412.

[2] (2020) Ipfs documentation - what is ipfs? [Online]. Available: https://docs.ipfs.io

[3] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[4] B. Inc. (2020) Btfs documentation - what is btfs? [Online]. Available: https://docs.btfs.io

[5] "Internet Protocol," RFC 791, Sep. 1981. [Online]. Available: https://rfc-editor.org/rfc/rfc791.txt

[6] "IP datagram reassembly algorithms," RFC 815, Jul. 1982. [Online]. Available: https://rfc-editor.org/rfc/rfc815.txt

[7] R. Bonica, F. Baker, G. Huston, B. Hinden, O. Trøan, and F. Gont, "IP Fragmentation Considered Fragile," Internet Engineering Task Force, Internet-Draft draft-ietf-intarea-frag-fragile-17, Sep. 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-intarea-frag-fragile-17

[8] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[9] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, vol. 29, no. 2016, 2016.

[10] M. Sabadello, K. Den Hartog, C. Lundkvist, C. Franz, A. Elias, A. Hughes, J. Jordan, and D. Zagidulin, "Introduction to did auth," *Rebooting the Web of Trust VI*, 2018.

[11] L.-E. Jonsson, L. Åke Larzon, G. Fairhurst, S. Pink, and M. Degermark, "The Lightweight User Datagram Protocol (UDP-Lite)," RFC 3828, Jul. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3828.txt