

Blockchain Solution for IoT-based Critical Infrastructures: Byzantine Fault Tolerance

Omar Alfandi*, Safa Otoum*, Yaser Jararweh[†]

*College of Technological Innovation (CTI), Zayed University, UAE

[†]Duquesne University, PA, USA

Emails: {Safa.Otoum; Omar.Alfandi}@zu.ac.ae, Jararweh@duq.edu

Abstract—Providing an acceptable level of security for Internet of Things (IoT)-based critical infrastructures, such as the connected vehicles, considers as an open research issue. Nowadays, blockchain overcomes a wide range of network limitations. In the context of IoT and blockchain, Byzantine Fault Tolerance (BFT)-based consensus protocol, that elects a set of authenticated devices/nodes within the network, considers as a solution for achieving the desired energy efficiency over the other consensus protocols. In BFT, the elected devices are responsible for ensuring the data blocks' integrity and preventing the concurrently appended blocks that might contain some malicious data. In this paper, we evaluate the fault-tolerance with different network settings, i.e., the number of connected vehicles. We verify and validate the proposed model with MATLAB/Simulink package simulations. The results show that our proposed hybrid scenario performed over the non-hybrid scenario taking throughput and latency in the consideration as the evaluated metrics.

Index Terms—Internet of Things (IoT), Blockchain, Byzantine Fault Tolerance (BFT), Fault Tolerance (FT).

I. INTRODUCTION

Since the Internet of Things (IoT) devices are in an open environment, easily reachable areas, as well as they, are under the opponent's control. The blockchain-based solution can insure such devices transmitted information in a safe, secure and confident way [1] [2].

IoT network consists of smart devices (sensors, cars, actuators) that gather data about their surrounding environments and wirelessly connect within the network to be able to share the aggregated data. Some remote devices (phones, laptops) can control such IoT devices that enable users to take advantage of the aggregated data. Such aggregated data can be stored in various storage (cloud, database or on the IoT devices).

Fault tolerance considers as a mechanism that enables a system to continue working in case of failures in one or more of its components. Byzantine Fault Tolerance (BFT) mechanism considers as a requirement for critical infrastructures that depend on numerous sensors' behaviours.

Data centralization considered as one of the major issues for IoT infrastructures, where the aggregated data from different IoT devices is stored on centralized storage (such as the servers), which open the door for third parties to access and manipulate such aggregated data and cause a failure for the whole network [3].

Blockchain distributed characteristics (no central control point) could improve the IoT devices' fault tolerance criteria. In the blockchain, all devices (nodes) collectively are responsible for approving any new blocks [4]. IoTs centralized property cause some resources consumption, such as, in IoT networks, when all workloads are performed in a centralized point, the computing, storage and bandwidth will be consumed and lack the number of connected devices. Blockchain - IoT integration would be a solution to resource consumption issues. Such as, the decentralized networks will distribute the workloads over many devices and provide more computing resources than the centralized IoT networks. In Blockchain technology, the data stability is guaranteed with the cryptography characteristic. Adding such cryptography characteristics to IoT devices can guarantee secure data transmission and storage within IoT networks. Within such a mechanism, a blockchain can keep track of the devices connected to the IoT network and keep the history of each device [4].

All IoT Devices identification could also be guaranteed within blockchain-IoT integration, which enables the data authentication and authorization within all IoT devices data. The blockchain technology can also guarantee secure communication among individual IoT devices [2], this can be done by storing the transmitted data in various transactions and performing transactions validation with the individual devices [5].

To this end, blockchain allows IoT devices to communicate within each-others through encrypted identities (ids) using point to point (p2p) communication and without the interfering of central storage, a database or a server, where each transaction is added to a block and distributed equally to all devices within the network [4] [3]. In this manuscript, we aim at analyzing the network serviceability for a blockchain system that serves critical IoT infrastructures.

II. RELATED WORKS

IoT network consists of numerous devices that are interconnected to each other to aggregate various types of information. IoT targets various services and infrastructures with various requirements.

The work in [6] presented an architecture for a blockchain-based smart city in which they used various blockchain

systems such as NXT, Ethereum systems for different IoT scenarios.

The authors in [7] proposed a fault tolerance and energy-efficient protocol that scheduled active sensor nodes and backup sensor nodes in the backbone to tolerate the system failures.

The work in [8] presented a flexible, hierarchical clustering method for homogeneous Wireless Sensor Network (WSN) in which they aimed to solve the fault tolerance as one of their five issues. The Function Delegation protocol has been adopted to provide the required fault tolerance. The authors in [9] developed a fault tolerance method with cluster-based structure and introduced a fault model based on sensors behaviours in the heterogeneous WSNs.

The consensus aims to guarantee the system security, liveness, and consistency of data stored in the ledger [10]. Other works have presented pBFT as the consensus protocol as the works in [11], [12] and [13].

The authors in [13] extended the pBFT by using fast track and client-actively involving into the consensus process in which the client is the major player who is responsible for replicas integrity-checking purposes.

III. MODEL

Blockchain-IoT integration could be in various architectures, however, deciding where the interactions will be, consider as the first step of the architecture. Here, in our proposed solution, we introduced the hybrid approach for Blockchain-IoT interaction in which most of the interactions and the transmitted data will be directly shared amongst the IoT devices (vehicles), however, some data will be stored in the blockchain. This hybrid architecture can leverage the advantages of both the IoT and blockchain interactions. The hybrid architecture enables implementing cloud and fog computing which is beneficial for overcoming some blockchain and IoT devices limitations such as reducing the operating expenses. Hybrid architecture has been implemented, fog computing which adopts fewer computational devices, such as gateways has been introduced to our hybrid architecture.

In this work, BFT has been used for integrating blockchains into IoT networks, such protocol has overcome the byzantine issue and protect the network against faults by distributing the services on multiple nodes.

Practical BFT (pBFT), as a consensus method, has been used in our work. In pBFT, all IoT devices are responsible for the election process to add the next device where the consensus can be achieved if more than two-thirds ($2/3$) of devices agree on the new block. The reasons behind choosing pBFT as our consensus technique is because of its fast speed, low latency and low computational overhead.

pBFT considers as a consensus technique for distributed systems to reach a consensus even if some devices didn't respond or respond with fault information. In BFT, network failure can be overcome by considering a collective decision making (correct and incorrect devices), that goals to minimize the influence of the incorrect device.

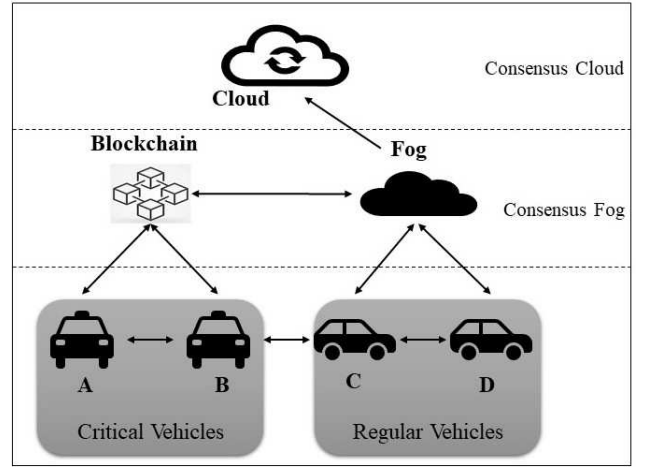


Fig. 1. Blockchain-based IoT critical infrastructure model

As shown in 1, our system is composed of two types of IoT devices, the critical vehicles (police cars, ambulance or fire trucks,...etc) and the regular personal vehicles. The aggregated data from critical vehicles (such as the vehicles A and B in Figure 1) is directed to the consensus to be verified and appended to the blockchain then the consensus data will be directed to the fog. On the other side, the aggregated data from regular vehicles will be directly forwarded to the fog. The aggregated data in the fog will be directed to the cloud which will be a consensus cloud. The need for the consensus procedure is to ensure that every block (vehicle) in the chain is true and authentic since we are dealing with critical infrastructure.

Since Byzantine fault is hard to detect and it can destroy the whole system [14] [9], in our blockchain-based IoT network, we adopt Practical BFT (pBFT) as our consensus protocol that can immediately confirm the appended blocks as well as it tolerates n malicious with $3n + 1$ replicas nodes. Once any critical vehicle sends its data to the consensus block and till the data is confirmed and added to the blockchain, various hops and consensus machines will be utilized. Adding more machines to the consensus process will improve its fault resiliency, which is our focus in this manuscript.

In the practical BFT, a new device will send a request to the primary device (proposer) which has been chosen following the Round Robin style. The primary device broadcasts this request to all the replicas at the same time where these replicas perform the request and send the reply to the new device. The new device will wait for $n + 1$ replies with the same reply, where n refers to the maximum number of faulty replicas.

The used algorithm is presented in details in Algorithm 1, where d_{id} is the new device id, d_{sig} refers to the new device signature, p represents the primary device and m refers to all devices except the primary p . In Algorithm 1, the primary device starts by broadcasting a PRE-PREPARE message, that contains the new proposed device, and the other devices will enter the PRE-PREPARED phase and broadcast this message

to all devices. After that, the devices will change their states from PRE-PREPARED to PREPARED where the prepared devices will send COMMIT messages to each other and move to the COMMIT state and add the new block to the chain and move to the FINAL COMMITTED phase. The pBFT has been implemented for critical vehicles only.

Algorithm 1 practical Byzantine Fault Tolerance (pBFT)

- 1: **procedure** PBFT
- 2: **input:** d_{id}, d_{sig}, t, m, p
- 3: **Request:** new device (d) sends a request r
- 4: **Received Request:** P received the request r .
- 5: broadcast $\ll PRE - PREPARE$
- 6: **for** $i = 1, 2, 3, \dots, m$ **do**
- 7: m change their status to $PRE - PREPARED$
- 8: m broadcast this message to each others
- 9: m change their phases to PREPARED
- 10: m send COMMIT messages to each other
- 11: **After** $2n + 1$ **commits:** devices will move to COMMIT phase.
- 12: **COMMIT:** devices will move to FINAL COMMITTED phase.
- 13: Devices elect new primary device (P')

The notations used in the description of the system model, adopted techniques and algorithms are presented in Table I.

TABLE I
NOTATIONS USED IN TESTED METHODS

Notation	Description
d	Newly connected device
r	request
d_{id}	the new device id
d_{sig}	the new device signature
t	Time
m	all devices except the primary
p	the primary device

IV. PERFORMANCE EVALUATION

In our simulation, we simulate our model using MATLAB/Simulink package along with the Contiki operating system [15]. We tested the pBFT protocol on 100 nodes. We focused on the network throughput along with the latency perspectives as presented in Figures 2 3 4 and 5.

Two scenarios has been tested in which:

- First Scenario: Non-Hybrid model
All vehicles are connected to blockchain for consensus then the collected data are directed to the cloud without the use of fog. Figure 2 represents the throughput for two data sizes (5K and 10K) for this scenario. The latency of this scenario is presented in Figure 3.
- Second Scenario: Hybrid model
Our proposed hybrid model in which only critical devices are considered for consensus while regular devices directly send their collected data to the fog. Figure 4 represents the throughput for two data sizes (5K and 10K)

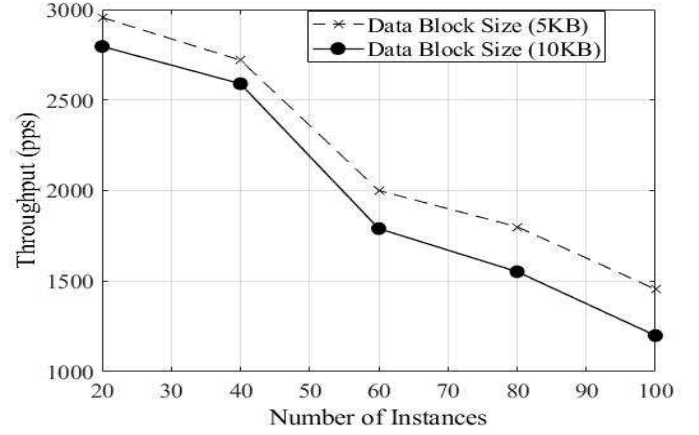


Fig. 2. Throughput comparison between 5K and 10K data size for the first scenario

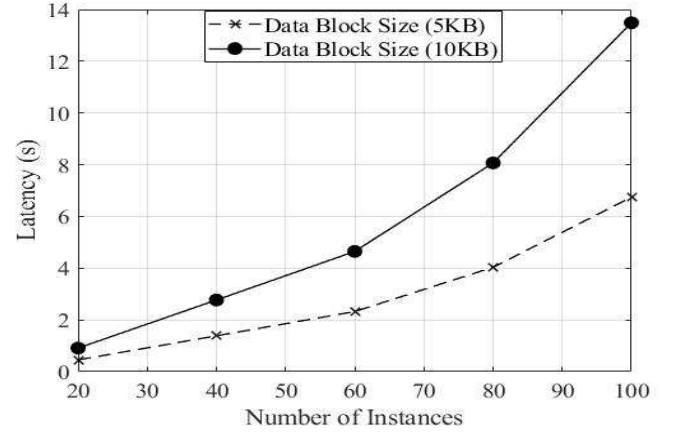


Fig. 3. Latency comparison between 5K and 10K data size for the first scenario

for this scenario. The latency of this scenario is presented in Figure 5. A clear comparison between the first and second scenarios in terms of latency and throughput is presented in Figures 6 and 7 below.

V. CONCLUSION

In critical infrastructures, network resiliency has a vital role in the credibility of any monitoring system. In this manuscript, we presented an integrated blockchain-based Internet of Things (IoT) architecture supporting fault tolerance capabilities as Byzantine based blockchain system. Fault resiliency has been achieved by introducing the blockchain capability with keeping a consensus fog. The presented results were supported using MATLAB simulations. The adopted practical Byzantine Fault Tolerance (pBFT) overcome many fault issues.

REFERENCES

- [1] M. Aloiail, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc*

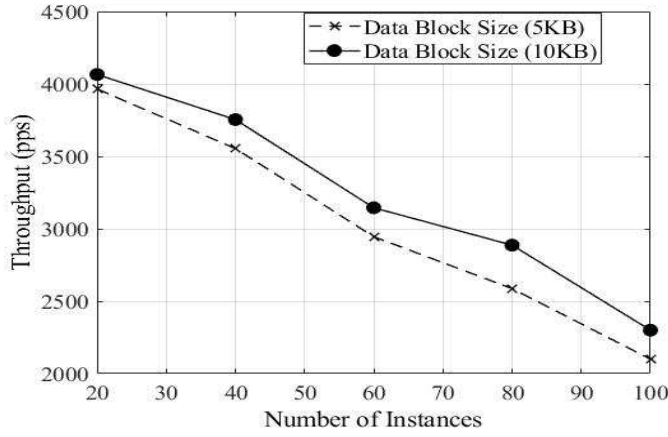


Fig. 4. Throughput comparison between 5K and 10K data size for the proposed hybrid model

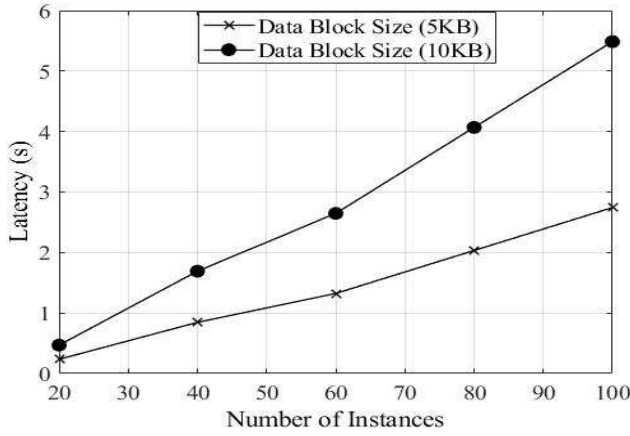


Fig. 5. Latency comparison between 5K and 10K data size for the proposed hybrid model

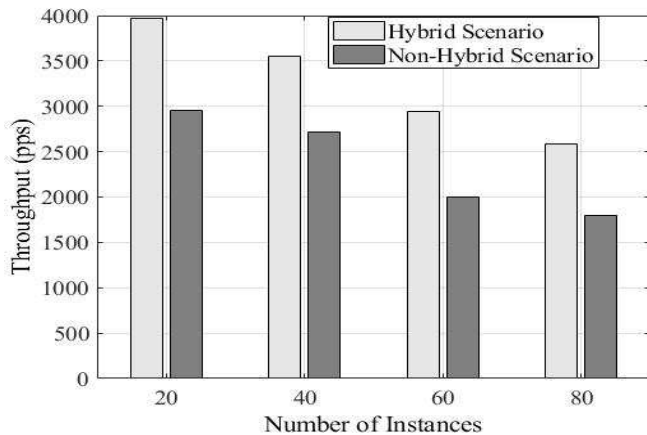


Fig. 6. Throughput comparison between the two scenarios

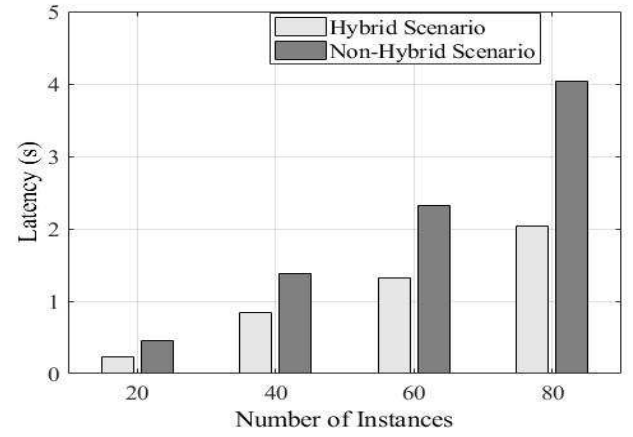


Fig. 7. Latency comparison between the two scenarios

Networks, vol. 90, p. 101842, 2019, recent advances on security and privacy in Intelligent Transportation Systems.

[2] I. R. A. M. J. N. K. R. Rathee G., Sharma A, "A blockchain framework

for securing connected and autonomous vehicles," *Sensors (Basel)*, 07 2019.

- [3] I. Al Ridhawi, S. Otoum, M. Aloqaily, Y. Jararweh, and T. Baker, "Providing secure and reliable communication for next generation networks in smart cities," *Sustainable Cities and Society*, p. 102080, 02 2020.
- [4] M. Aloqaily, S. Otoum, L. Tseng, and J. Othman, "Blockchain for managing heterogeneous internet of things: A perspective architecture," *IEEE Network*, 02 2020.
- [5] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for iot networks," 09 2018.
- [6] K. Biswas and V. Muthukumarasamy, "Securing smart cities using blockchain technology," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Dec 2016, pp. 1392–1393.
- [7] C. Chen, K. Ssu, and H. C. Jiau, "Fault-tolerant topology control with adjustable transmission ranges in wireless sensor networks," in *13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)*, Dec 2007, pp. 131–138.
- [8] R. Kumar and U. Kumar, "A hierarchal cluster framework for wireless sensor network," in *2012 International Conference on Advances in Computing and Communications*, Aug 2012, pp. 46–50.
- [9] S. Chang and T. Huang, "A fuzzy knowledge based fault tolerance algorithm in wireless sensor networks," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, March 2012, pp. 891–896.
- [10] L. Lamport, "Generalized consensus and paxos," Tech. Rep. MSR-TR-2005-33, March 2005.
- [11] A. Bessani, J. a. Sousa, and M. Vukoliundefined, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, ser. SERIAL '17. New York, NY, USA: Association for Computing Machinery, 2017.
- [12] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muramidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," *CoRR*, vol. abs/1801.10228, 2018.
- [13] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzyva: Speculative byzantine fault tolerance," vol. 51, 01 2007, pp. 45–58.
- [14] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI '99. USA: USENIX Association, 1999, p. 173–186.
- [15] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," *29th Annual IEEE International Conference on Local Computer Networks*, pp. 455–462, 2004.