Trustworthy IoT Computing Environment Based on Layered Blockchain Consensus Framework

Yueyu Dong, Fei Dai(✉), and Mingming Qin

Southwest Forestry University, Kunming, China
{dongyueyu,daifei,swfuqmm}@swfu.edu.cn

Abstract. The Internet of Things is widely used and has far-reaching significance. It is essential to ensure the trustworthiness of the IoT computing environment. Using blockchain technology to store and manage the data traces in the IoT is feasible to implement the trustworthy IoT. In a "Cloud-Edge-End" structure, the difference in the degree of energy constraint between various parts makes the use of the same consensus algorithm a compromise between overall performance degradation and energy constraints on terminal devices. We take advantage of the high modularity of the chained consensus algorithm to build a layered consensus mechanism framework, running a two-phase consensus algorithm in the local environment of the terminal devices, and a three-phase consensus algorithm with better overall performance. Preliminary evaluation shows that this scheme is feasible.

Keywords: Trustworthy IoT · Blockchain · Consensus framework

1 Introduction

The Internet of Things (IoT), which realizes the Internet of Everything, connects the information network and the real world. The IoT has unprecedentedly expanded the reach of the Internet, and has spawned many new applications, such as smart homes, smart cities, smart healthcare, and industrial Internet of Things. It is considered the key technology of the fourth industrial revolution. Under the data-driven paradigm, the IoT is not only an important data information source, but also an important infrastructure for data processing and application.

On the other hand, IoT devices are generally exposed and lack environmental safety guarantees. The wireless communication technology is widely used in the IoT, which is vulnerable to interference, data theft, and network attacks. The security and credibility of IoT devices and data has received great attention [3, 6]. The transactions in the IoT, especially whether the relevant data is trustworthy, are critical to the application of the IoT [2, 7]. For scenarios such as smart transportation and smart medical care, it is even directly related to human life safety.

To achieve a trustworthy IoT, it is necessary to ensure that the IoT obtains information and provides feedback based on credible data. Blockchain is a new computing model

M. R. Khosravi et al. (Eds.): CloudComp 2021, LNICST 430, pp. 167–180, 2022.
https://doi.org/10.1007/978-3-030-99191-3_13

that integrates distributed storage, consensus mechanism, and encryption algorithm. The decentralization and anti-tampering characteristics of the blockchain can ensure the security and trustworthiness of data in the IoT system. Use blockchain to store complete data traces, including who, when, where and how to obtain or generate, how processed, what kind of derivatives were generated, what kind of applications were used, and so on. It can ensure that the data is tamper-proof, complete, and traceable.

The IoT is a heterogeneous system. In a typical "Cloud-Edge-End" three-tier environment, there are huge differences in computing power, storage capacity, network bandwidth, and energy supply among cloud computing centers, edge servers at the edge of the network, and numerous terminal devices. Especially, energy constraints are most representative. It can be clearly divided into two parts: resources-unconstrained part and resources-constrained part. The consensus mechanism is the core foundation for the realization of functions of the blockchain. One single consensus mechanism can adapt to the huge differences in such a heterogeneous system is a major challenge of the integration blockchain with the IoT, which can run on the cloud with sufficient resources, and can also work on resource-constrained terminal devices.

The Byzantine Fault Tolerant (BFT) algorithm based on State Machine Replica (SMR) is an important type of blockchain consensus mechanism, such as PBFT [10]. These algorithms are derived from early distributed systems and required blockchain-oriented improvements. HotStuff [11] is representative work in this series of improvements. This algorithm implements the so-called chained SMR protocol. The process of consensus can be accomplished by a chain of identical structured consensus phases in a manner similar to the chain of blocks in blockchains. It archives the separation of safety rules and liveness rules. The liveness mechanism can be implemented independently. The high modularity of the identical structured consensus phases makes it possible to build a unified blockchain consensus framework in the IoT heterogeneous environment based on this algorithm. This framework is promising to meet the demands of the trustworthy IoT.

Specifically, the main contributions of this article include:

- Proposed a general consensus framework suitable for heterogeneous IoT environments based on a common algorithm foundation;
- Designed the voting phase and view changing algorithm to adapt to different layers in the framework;
- Proposed the deployment plan of the consensus framework based on microservices.

2 Related Works

Blockchain originated from encrypted electronic currency, and with further research, its unique value has become increasingly visible. In the field of IoT, using blockchain to ensure the trustworthiness of transactions is the most typical application. Huang et al. [4] proposed a blockchain system that can run in the resource-constrained environment of IoT terminal devices, which ensures that the transactions on these devices are tamper-proof and non-repudiation. Liu et al. [5] used blockchain in a heterogeneous IoT environment to ensure data security and credibility. It also emphasizes that the blockchain

consensus mechanism must not only be able to adapt to "brawny" nodes with abundant resources, but also be able to adapt to "wimpy" nodes with limited resources. The work of Bai et al. [2] focuses on the trustworthiness of computing service scheduling and edge data sharing in the heterogeneous environment of the industrial energy Internet of things.

Traditional IoT applications run on the infrastructure composed of cloud servers and terminal devices. Due to their own characteristics, IoT terminal devices lack sufficient capabilities and need to transmit data to the cloud for computing and storage. Therefore the industry proposes a new computing model that provides data processing capabilities at the edge of the network, that is, Edge Computing. The IoT system infrastructure that introduces edge computing has changed from the traditional two layers to three layers composed of "Cloud-Edge-End". The introduction of edge gateways, edge servers and other devices has enabled the computing and storage capabilities of the traditional cloud to be deployed closer to IoT terminal devices. Under the offloading of computing tasks, data analysis and processing can be performed at nearby locations, which greatly avoids the huge communication overhead with the cloud. To achieve blockchain consensus in the IoT environment, the role of edge servers is unique and critical. Terminal devices usually lack sufficient resources and cannot afford the calculation and communication costs of the consensus process. Terminal devices are energy-constrained, and cannot always stay online. Works such as DPoS [7] also shows that in the blockchain, it is not the best choice for all nodes to directly participate in the consensus process. It is a common solution to use the edge server as the agent instead of the terminal node to participate in the consensus. In Edgence [9], the blockchain is used to implement self-management and self-supervision of decentralized applications. The core function of the proposed platform is to be deployed on edge servers. The literature PoQF [1] applies blockchain in the vehicular network environment. Its work shows that without the support of edge servers, the blockchain consensus mechanism is difficult to implement in the vehicular network environment.

In the heterogeneous environment of the IoT, the blockchain consensus algorithm needs to adapt to the resource-constrained terminal devices. The literatures Liu, Huang, and PoQF all proposed new consensus algorithms for this demand. Among them, the literature Huang and PoQF independently designed new algorithms, and the literature Liu used their new strategies to improve the classic GHOST algorithm. Entirely new algorithms are not fully tested. Improved existing algorithms, especially those that have been verified by practical applications, is mature and reliable. It is usually without too many compromises. The HotStuff algorithm has Byzantine fault tolerance comparable to classic algorithms such as PBFT. And it has been applied in Facebook's electronic currency Libra [8].

3 Trustworthy IoT Model with Integrated Blockchain

The key to achieve a trustworthy IoT is to ensure that the data in the IoT system is credible. The blockchain has the characteristics of decentralization and tamper-proof. Integrating the blockchain into the IoT and recording data traces with the blockchain can ensure the data credibility (Fig. 1). In this model, all participating nodes should be verified and trustworthy.

3.1 System Model Overview

In the typical "Cloud-Edge-End" three-tier structure of the IoT, the whole system can be clearly divided into two parts according to the degree of energy constraint. Cloud and edge servers belong to the part with less resource constraints; the part mainly composed of terminal devices, which is the narrowly defined the Internet of Things, is the part with generally constrained energy supply. For a blockchain consensus algorithm to be able to run in a "Cloud-Edge-End" environment, it must adapt to this difference. The blockchain integrated into the IoT also needs to be composed of two various parts. The part running on the terminal devices ensures that the data in the local network is credible. The part that runs on the cloud and edge servers records the data traces on the cloud and edge servers. The edge server has a unique position. The two parts can be connected through the edge servers, so as to implement the global trustworthiness of data.

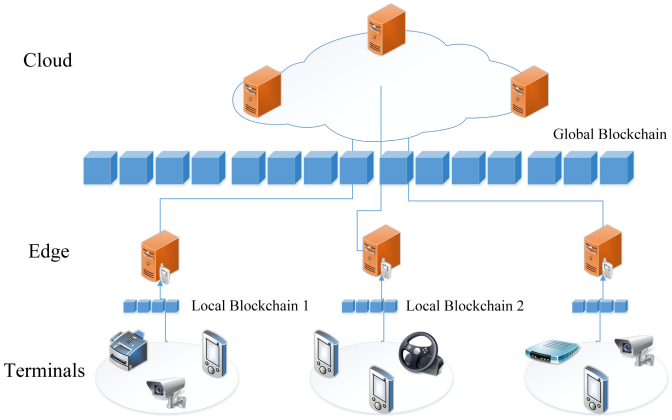

Fig. 1. A synergistic framework for the integration of IoT and blockchain

3.2 Build a Two-Layer Blockchain for the IoT

The blockchain integrated into the IoT needs to adapt to the two different components in the IoT. The blockchain on the energy-constrained part collects the data traces in this part, and packs a series of data transactions into blocks under consensus. The process of packaging and consensus can be performed on the edge server by task offloading. In fact, edge servers also act as participants in this layer of blockchain consensus. Therefore, whether when the edge server is selected as the leader to dominates the consensus and to pack data traces into blocks, or when other nodes offload packing tasks to the edge server, there will always be blocks copies of trusted data traces of the energy-constrained part on the edge server. These data traces blocks will be placed onto a local blockchain. The local blockchain does not need to record all the blocks, and just keep a sufficient amount of recent blocks to meet the needs of the consensus process. By participating in the consensus process of the energy-unconstrained part by the edge server, these data traces record blocks can be finally added to the global blockchain and stored in the energy-unconstrained part. Shown in Fig. 2.

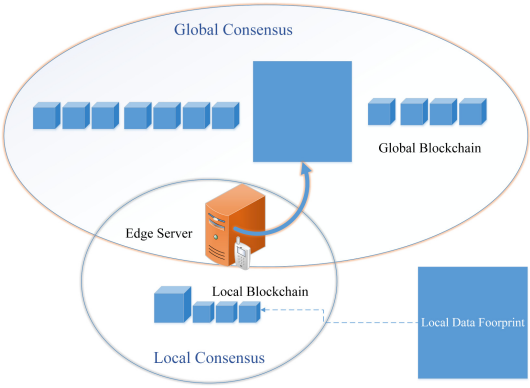

Fig. 2. Exchange between the global consensus and the local consensus

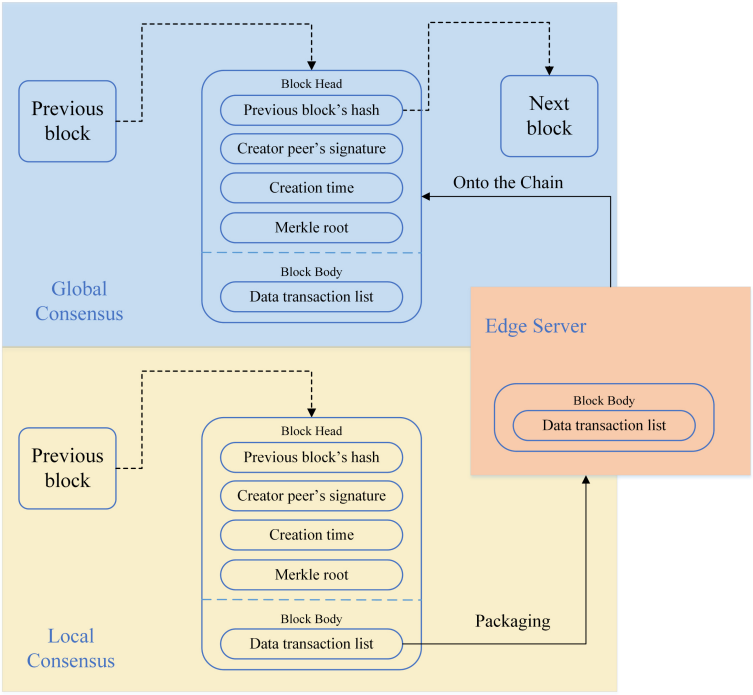

Fig. 3. The structure of the layered blockchain

The structure and operations of the global blockchain running on the energy-unconstrained part are more similar to the ordinary blockchain. The edge servers and the clouds record the data traces in the energy-unstrained part, pack the data traces into blocks, and join blocks into the global blockchain by consensus. The latest confirmed block in the local blockchain is added to the global blockchain under consensus

by the edge server serving as the consensus initiating node. In this way, as shown in Fig. 3, the data traces in the entire IoT system are recorded in the blockchain to achieve trustworthiness.

4 Layered Chained BFT (LCBFT) Consensus Mechanism

The most important contribution of HotStuff is a byzantine fault tolerant consensus framework for blockchain. Under this framework, the consensus process can be regarded as a series of successive phases and phases are identical structured. This denoted as chained consensus in its document. And it achieves the separation of safety rules and liveness rules. Consensus voting with different "chain lengths" company with different view change mechanisms. Such a consensus framework has high modularity and has the ability to adapt to different scenarios. Based on this idea, the LCBFT consensus mechanism is proposed to implement the aforementioned two-layer blockchain.

4.1 Overview of the HotStuff

The process of the basic HotStuff algorithm includes the leader node initiating a consensus; voting on the consensus proposal; the leader broadcasting voting results, the replica node voting on the results of the first round of consensus, and pre-submitting; the leader broadcasting the results of the second round of voting; the replicas vote on the results of the second round of voting, and finally reach a consensus. The whole process can be simplified and described into three structure-identical phases. Each phase consists of two main steps: the leader node broadcasts the proposal, and the replica node votes. As shown in Fig. 4.

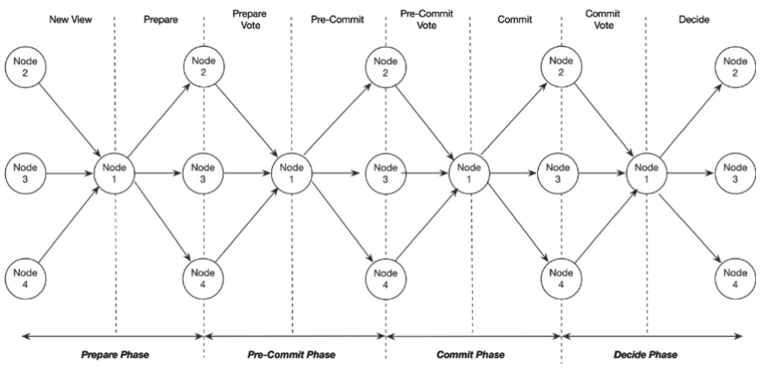

Fig. 4. HotStuff consensus process includes several identical-structured phases

After the completion of the three phases, this round of consensus was reached. To change the view to ensure the quality of the chain. Because the three phases of one round are identical-structured, and the content voted to be confirmed in each phase is successive, consensus process can form a pipeline. The second phase on the first block

can be used as the first phase on the second block. When the consensus on the fourth block is started, the three phases on the first block have been completed, and the first block has been confirmed and cannot be tampered with ever. At this time, the so-called "Three-Chain" is formed.

The above description is mainly about the security rules in the consensus mechanism. The liveness rules corresponding to safety rules are also required. HotStuff itself is a Three-Chain consensus protocol. The consensus on the current proposal also means the confirmation of the location of the unmodifiable block in the blockchain. The cost of view changing is linear, therefore leader replacement can be performed in each round. Under the HotStuff framework, the Two-Chain consensus protocol can also be constructed. Explicit corresponding liveness rules are needed, which is similar as the view changing algorithm in PBFT, or the mandatory delay in Casper, and so on.

4.2 Consensus and Block Generation in Energy-Constrained Part

Due to limited resources, the blockchain consensus mechanism in this part requires lower computational overhead. The use of two-phase consensus can effectively reduce computational overhead. In this part, the key issue is the credible record of data traces, not a complete blockchain for persistent storage. All need to do is to keep the "chain of necessary length" to meet the needs of the consensus process.

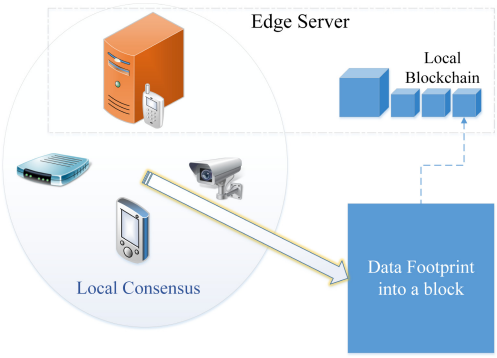

Fig. 5. Data footprints collected and recorded trustworthily in local consensus process

The edge server participates the consensus progress in this part. After the consensus is reached, the data transactions are packed into blocks, and the edge server records the data traces of this part and submits them to the global blockchain, then these blocks will be stored onto the chain under consensus. Shown in Fig. 5. The edge server is naturally suitable as the leader of the consensus in the energy-constrained part. When other nodes act as leaders, they also can offload computing tasks such as collecting votes to the edge server.

4.3 Global Blockchain Consensus and Joining Blocks onto the Chain

The consensus of global block chain is reached with the participation of clouds and edge servers, and data traces occurring on cloud and edge servers are stored on the global blocks chain (Fig. 6). Among them, the blocks that the edge server submitted includes data traces in the energy-constrained part. The global consensus of block chain adopts a three-phase process to improve data throughput, and changes view every round to ensure the quality of the blockchain.

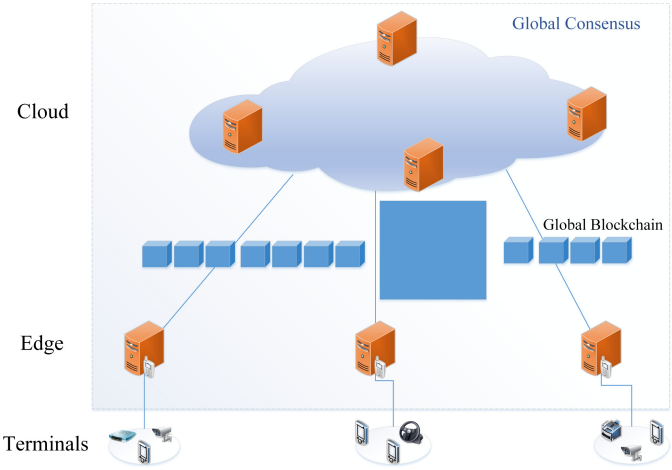

Fig. 6. Cloud and edge servers participate in global consensus

4.4 Liveness Mechanism of Consensus Process

Considering the constrained energy supply in the resource-constrained part, the Two-Chain consensus mechanism is used in this part. At this point, the consensus process is similar to classic algorithms such as PBFT, and multiple rounds of voting are required to determine a new leader according to the view changing protocol to complete the view change. This process is costly in calculation and network communication. But in this scenario, the view changing is not performed every round, and it is only performed when the leader node is faulty or not trusted. Therefore, the frequency of view change can be reduced by optimizing the leader selecting. thereby the overall cost of the consensus mechanism can be reduced. In the “Cloud-Edge-End” IoT environment, determining leader mainly by stable online time can ensure that stable nodes such as edge servers act as leader to dominate the consensus process, and can reduce the overhead caused by view change. Algorithm 1 and 2 describes the whole process systematically.

Algorithm 1 Local Consensus

1: **begin**
2: Two phases voting for safety
3: Choose new leader node
4: Several voting to confirm new leader and the check point for liveness
5: **end**

Algorithm 2 Choose new leader node

1: **begin**
2: **for** every terminal t_i (including the edge server) in the same local consensus domain **do**
3: Check accumulative available time ta_i of t_i
4: **if** ta_i is the max **then**
5: Choose t_i as new leader node
6: **end if**
7: **end for**
8: **end**

Three-Chain consensus can be carried out in the resource-unconstrained part composed of cloud servers and edge servers, and the view can be switched every round to improve the quality of the blockchain. The cost of each round is linear, but it is still necessary to select the leader node for a new round of consensus. In this part, because there are no constraints in capability and energy, a new leader node can be determined in a round-robin manner. As shown in Algorithm 3.

Algorithm 3 Get leader in global consensus

1: **begin**
2: Check the identifying number i of the current leader node p_i
3: **if** $(i + 1)$ % n (amount of the peers) == j **then**
4: Choose p_j as the leader node of next round
5: **end if**
6: **end**

5 Microservice-Based Consensus Protocol Deployment Plan

In the chained consensus protocol, each phase of the consensus process is identical-structured, that is, collecting votes and publishing voting results or launching the next round of voting. In the HotStuff protocol, voting is achieved with the help of Threshold Digital Signature. From the perspective of application deployment, one phase of the consensus process can be deployed as a reusable software module in the computing environment, and the consensus process can be implemented through multiple reuses. The three-phase mechanism and the two-phase mechanism used in different parts of the system can be achieved by reusing corresponding times. The view changing process to ensure liveness in the two-phase consensus mechanism is also a series of voting phases. It also can be implemented by running reusable software modules, which can be brought about by threshold signatures.

Algorithm 4 One basic phase of the consensus

```
1: begin
2:     as a leader:
3:         collect votes with Threshold Digital Signature
4:         broadcast the vote result
5:     as a replica:
6:         wait for message from leader, and vote after message received
7:     as the next leader:
8:         wait for messages until there are n-f votes, then start to act as the leader
           node
6: end
```

Microservices are a feasible way to deploy reusable software modules. Reusable modules can be deployed on cloud servers and edge servers in the form of microservices. In the global consensus process, the cloud server and the edge server reach consensus through three rounds of reuse the reusable module, and record the data traces on the chain. In the local consensus process, poor energy supply constrains replica nodes to run the consensus algorithm. At this time, the related computing tasks can be migrated to the edge server by offloading. A consensus can be achieved by reusing of the aforementioned reusable modules certain times. In the consensus process, the edge server may be a malicious node, but this does not affect its use as a destination for tasks offloading in the edge computing mode. If a view changing occurs, it can also execute the tasks offloaded from those energy-constrained nodes, which cannot afford the execution cost. So as to perform the view change and start the next round of consensus. The microservices Si provides a basic phase computing of the chained consensus algorithm, which is deployed in the part consisting of the cloud servers CSi and the edge servers ESi. The microservices also can be deployed on energy-unconstrained terminal devices (Fig. 7).

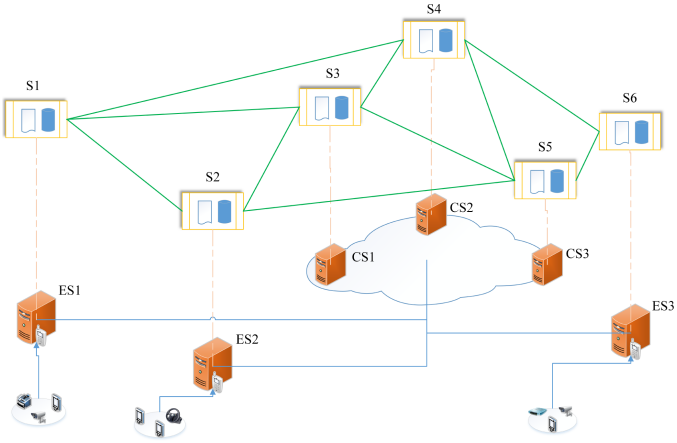

Fig. 7. Deployment of the consensus protocol based on microservice

6 Evaluation

The energy consumption of the consensus process is proportional to the calculation amount. By analyzing the calculation amount of the consensus process, the energy consumption of the consensus process can be judged.

6.1 Computational Structure of the Two Consensus Mechanisms

As mentioned earlier, each round of global consensus is a three-phase process, and thus achieves linear cost of the view change. Because no additional complex liveness rule is needed, the view is changed every round. The calculation of the global consensus is mainly composed of three basic consensus phases (Algorithm 4), as shown in Fig. 8(a).

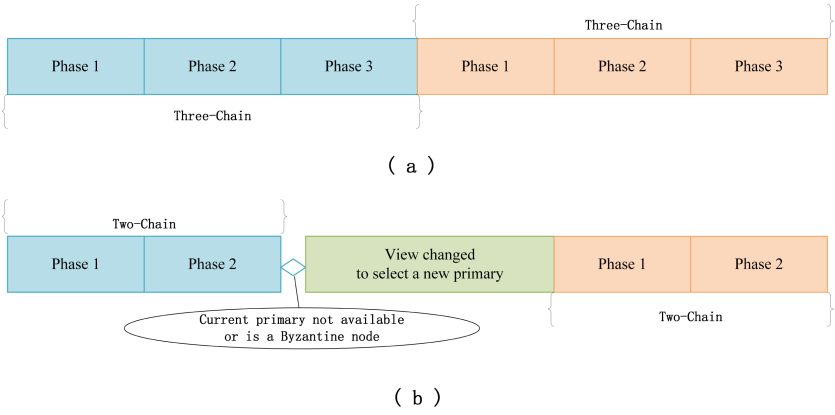

Fig. 8. The structure of one consensus round (a) is the structure of one round in 3-Phase consensus (b) is the structure of one round in 2-Phase consensus

Local consensus is carried out in a two-phase manner per round, as shown in Fig. 8(b). Different from the three-phase method of global consensus, an additional special view changing protocol needs to be run as main body of the liveness rules. For example, in the typical two-phase BFT consensus algorithm PBFT, after the view change is initiated, a consensus on the current stable checkpoint and the willingness to change to new view needs to be voted on, and then a reliable view change can be implemented to ensure the liveness of the consensus (Algorithm 5). Drawing on the thinking behind the chained consensus algorithm, this process can be described as a form composed of two basic phases, see Algorithm 6.

Algorithm 5 View change in PBFT

1: **begin**
2: Replica *i* collected *2f+1* votes for the current stable checkpoint, then broadcast *view-change* message with greater sequence number
3: Replica nodes vote to confirm new view *v+1* to new leader node selected according to *Algorithm 2*
4: New leader node collected *2f+1* votes for the view *v+1*, then broadcast *new-view* message with vote result
5: **end**

Algorithm 6 Description of view change based on the basic phases

1: **begin**
2: The basic phase to confirm the current stable checkpoint and propose changing to new view *v+1*
3: The basic phase to confirm the new view *v+1* and announce the begin of the new view
4: **end**

It should be noted that the view change in the local consensus is only performed when the current leader node is faulty or malicious. The view changing protocol will only be executed in these cases. On the whole, each round of the local consensus process is mainly composed of two basic phases and a view change appearing with a certain probability whose calculation amount equivalent to two basic phases.

6.2 Energy Consumption Analysis of Two Consensus Mechanism

Based on the above analysis of the calculation amount, a basic consensus phase can be used as the measurement unit of the calculation amount. Then, based on the measurements of calculation amount, the energy consumption of the consensus process can be analyzed.

The calculation amount of each round of global consensus is recorded as 3 basic phases. Let CM_g represent the calculation amount of the global consensus, and CM_s represent the calculation amount of a basic phase, then the calculation amount of each round of the global consensus can be recorded as:

$$CM_g = 3 * CM_s \tag{1}$$

In the calculation of each round of local consensus, it is certain that there will be two basic consensus phases. There is also a view change that appears with a certain probability p, and the calculation amount of the view change is equivalent to the two basic phases. The calculation amount of each round of local consensus CM_l can be recorded as:

$$CM_l = 2 * CM_s + p * 2 * CM_s \tag{2}$$

With reference to relevant literature and analysis based on the actual situation of the IoT, p is usually much less than 0.5. Therefore, CM_l is less than $3CM_s$ and less than CM_g, and our proposed two-layer consensus framework can meet the energy consumption constraints of the IoT, and can implement a trustworthy computing environment in the IoT.

7 Conclusion

After preliminary evaluation, the overall cost of the two-phase consensus is lower than that of the three-phase consensus, which is suitable for energy-constrained terminal devices. The consensus framework we propose can build a trustworthy IoT computing environment. The structure of the chained consensus facilitates deployment in a "Cloud-Edge-End" environment with microservices.

Acknowledgements. This work has been supported by the Project of National Natural Science Foundation of China under Grant No. 61702442 and 61862065, the Application Basic Research Project in Yunnan Province Grant No. 2018FB105, the Major Project of Science and Technology of Yunnan Province under Grant No. 202002AD080002 and No. 2019ZE005, the Project of Scientific Research Foundation of Yunnan Department of Education under Grant No.2017ZZX212.

References

1. Ayaz, F., Sheng, Z., Tian, D., Guan, Y.L.: A Proof-of-Quality-Factor (PoQF)-based blockchain and edge computing for vehicular message dissemination. IEEE Internet Things J. **8**, 2468–2482 (2021). https://doi.org/10.1109/JIOT.2020.3026731

2. Bai, F., Shen, T., Yu, Z., Zeng, K., Gong, B.: Trustworthy blockchain-empowered collaborative edge computing-as-a-service scheduling and data sharing in the IIoE. IEEE Internet Things J. X (2021). https://doi.org/10.1109/JIOT.2021.3058125
3. Guo, S., Hu, X., Guo, S., Qiu, X., Qi, F.: Blockchain meets edge computing: a distributed and trusted authentication system. IEEE Trans. Ind. Inf. **16**, 1972–1983 (2020). https://doi.org/10.1109/TII.2019.2938001
4. Huang, Z., Mi, Z., Hua, Z.: HCloud: a trusted JointCloud serverless platform for IoT systems with blockchain. China Commun. **17**, 1 (2020). https://doi.org/10.23919/JCC.2020.09.001
5. Liu, Y., Wang, K., Qian, K., Du, M., Guo, S.: Tornado: enabling blockchain in heterogeneous internet of things through a space-structured approach. IEEE Internet Things J. **7**, 1273–1286 (2020). https://doi.org/10.1109/JIOT.2019.2954128
6. Mendki, P.: Blockchain enabled IoT edge computing. ACM Int. Conf. Proc. Ser. Part **F1481**, 66–69 (2019). https://doi.org/10.1145/3320/15433/20166
7. Sun, W., Liu, J., Yue, Y., Wang, P.: Joint Resource allocation and incentive design for blockchain-based mobile edge computing. IEEE Trans. Wirel. Commun. **19**, 6050–6064 (2020). https://doi.org/10.1109/TWC.2020.2999721
8. Team, T.L.: State Machine Replication in the Libra Blockchain, pp. 1–21 (2020)
9. Xu, J., Wang, S., Zhou, A., Yang, F.: Edgence: a Blockchain-enabled edge-computing platform for intelligent IoT-based dApps. China Commun. **17**(4), 78–87 (2020)
10. Xu, X., Zhu, D., Yang, X., Wang, S., Qi, L., Dou, W.: Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain. ACM Trans Internet Technol. **21** (2021). https://doi.org/10.1145/3395331
11. Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., Abraham, I.: HotStuff: BFT consensus in the lens of blockchain, 1–23 (2018). (arXiv)