Blockgraph proof-of-concept

David Cordova Morales, Pedro Velloso, Alexandre Guerre, Thi-Mai-Trang Nguyen, Guy Pujolle, Khaldoun Alagha, Guillaume Dua

Blockgraph Proof-of-Concept

David A. Cordova M
Pedro B. Velloso
Alexandre Guerre
Thi-Mai-Trang Nguyen
LIP6 / Sorbonne Université
Paris, France
firstname.lastname@lip6.fr

Guy Pujolle
Khaldoun Alagha
Green Communications
Paris, France
firstname.lastname@green-communications.fr

Guillaume Dua
Squad
Paris, France
gdua@squad.fr

ABSTRACT

Blockgraph is a new structure of blockchain in which the blocks are interconnected in the form of a graph instead of a chain. Blockgraph has been designed to enable the use of blockchain in mobile adhoc networks and mesh networks by dealing with the problem of network partition due to node mobility. This paper presents a proof of the blockgraph concept using a testbed composed of five mesh routers. The demonstration shows that the blockgraph is created and maintained by the participant nodes in case of network split and merge.

CCS CONCEPTS

• **Networks** → **Network architectures**; *Mobile networks.*

KEYWORDS

Blockchain, Blockgraph, MANETs

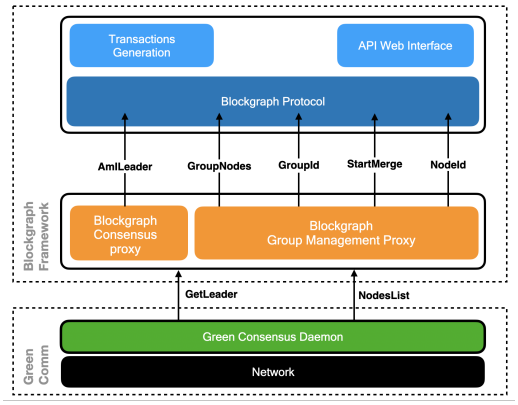

Figure 1: The overall architecture with interfaces of our Blockgraph solution using Green's consensus

1 INTRODUCTION

Blockchain is an emerging technology that provides a secure distributed ledger, which can store transactions in a tamper-resistant manner, allowing high availability and auditability. The last years have shown that not only cryptocurrencies and financial applications can profit from the benefits of blockchain. In fact, distributed ledgers might keep any kind of data not necessarily related to money and finance. Hence, several other applications, like electronic voting, health care and IoT, may profit from blockchain's advantages to improve their performance [4]. However, many of those new applications are characterized by high mobility and thus rely on Mobile Ad hoc Networks (MANETs) to work properly, as for instance, vehicular networks. Nevertheless, traditional blockchains were not designed to cope with mobility, since topology changes might cause the network to split into different independent partitions, as well as to merge into a single partition. Therefore, keeping the distributed ledger in a mobile scenario is a challenging issue.

In this context, we proposed a new solution, to adapt blockchain technology to MANETs called Blockgraph [2]. The main idea consists of representing the blockchain as a Direct Acyclic Graph (DAG), in which each independent partition generates its own blockchain corresponding to a single branch of the graph. Therefore, a partition split corresponds to new ramifications in the Blockgraph, while a partition merge correspond to the join of two or more branches into a single one. Thus, our Blockgraph protoocol is responsible for maintaining the distributed database in the presence of topology changes.

It is well-known that mobile wireless scenarios might impose severe network conditions that are not easily captured by simulations. Therefore, real implementations of MANET solutions are imprescindible not only to show their feasibility, but also to evaluate and improve their performance. In this demonstration, we present an implementation of our Blockgraph solution in a proprietary environment, which provides a platform with all the basic network capabilities that allow mobile ad hoc communications. Besides the implementation of our Blockgraph protocol, we also used a simple PBFT-like consensus algorithm. This demonstration serves as a proof-of-concept of the efficiency of our solution to maintain a distributed database in conditions of topology changes, as often can happen in MANETs.

2 BLOCKGRAPH FRAMEWORK

When we first introduce our Blockgraph solution in [2], we presented it as a framework composed of three different modules: (i) the consensus algorithm, which is the module in charge on agreeing on a single node capable of creating a new block; (ii) the group management system, which is the module responsible for discovering the network topology and for sending notifications to the Blockgraph protocol of network topology changes, and (iii) the Blockgraph protocol, which is in charge of managing the Blockgraph data structure and all the different functions leading to that goal; this includes, transactions and mempool management, the correctness of the Blockgraph data structure and data replication

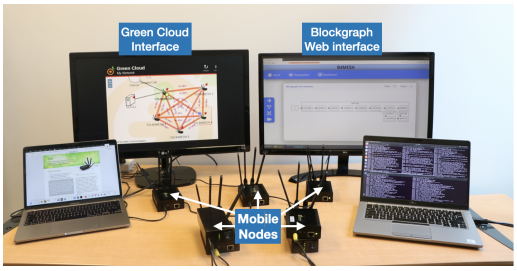

Figure 2: Testbed

across all nodes. For this implementation, we use a consensus algorithm developed by Green Communications as source of network topology information that can be used by our group management module but also as a leader provider to our consensus module. The architecture resulted from our implementation with Green's mesh routers is illustrated in Figure 1.

2.1 Consensus Daemon

The Consensus uses a PBFT-like leader election algorithm to elect a leader node in every network partition. To achieve this, the consensus daemon uses the routing table of QOLSR [1] routing protocol, which discovers the network topology in order to route packets in the network. It extracts information from the routing table to obtain an updated list of the reachable nodes in the network partition participating in the consensus process. For our Blockgraph framework, the consensus only needs to pass the updated list of reachable nodes to our group management module and the leader node to our consensus module. With these information our Blockgraph protocol has all the needed elements to function properly.

2.2 Merge Procedure

The Blockgraph merge procedure is the process of synchronizing divergent Blockgraph data structures into a single one. This process is triggered by the group management module when notifying the Blockgraph protocol of a merge. After a merge is detected, Blockgraph protocol must wait for a new leader to be elected for the new network partition. Once a leader is elected, it creates a merge block, which contains no transactions, but references to every last block created by each network partition, right before the merge. Upon the reception of a merge block from the current leader, the node uses the references included in the block header to identify the blocks that are not part of its Blockgraph data structure. Following, the node starts recovering blocks from divergent branches until finding a common block. For a detail explanation of the merge procedure, we refer the reader to our original work [2].

3 DEMONSTRATION

The testbed is composed of five low power mesh routers [3] deployed in a lab as illustrated in Figure 2. The Blockgraph daemon is installed in each mesh router.

At boot, each router connects to each other to join the Blockgraph network by creating a genesis block and start to emulate transaction generation. For the proof-of-concept of Blockgraph, we create a network partition by taking two mesh routers far away in the corridor until the network is completely split into two clusters. One network partition is composed of three mesh routers and the other with two routers. We let the two network partitions separated from each other long enough so both network partitions can generate new blocks. We then come back with these two routers to observe the behavior of Blockgraph in the case of merge.

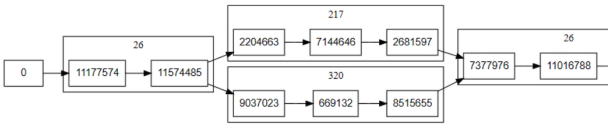

Figure 3: Blockgraph visualization

The visualization of the Blockgraph in router 1 is shown in Figure 3. We can see that the Blockgraph starts with block 0 (genesis block) then there are 2 blocks (in cluster 26) committed before a network partition happens. When the network is split into two clusters, the blocks continue to be created and committed to each of the two branches. Each branch corresponds to a part of the Blockgraph maintained by each cluster. We can see in this demonstration that after the split, 3 blocks were committed in cluster 217 and 3 others in cluster 320.

When we come back with two moving routers, there is a merge of two clusters (cluster IDs 217 and 320), into a single cluster of five nodes (cluster ID 26) as we have at the beginning. We can observe in Figure 3 the merge block (block-id = 7377976) is connected to two previous blocks (2681597 and 8515655) by the hashes of these blocks. As the cluster ID is the hash of all node IDs in the network partition, we can verify that the cluster ID after the merge is the same as the cluster ID before the split.

4 CONCLUSION

We implemented a prototype of Blockgraph using mesh routers. This demonstration serves as a proof-of-concept of our solution to maintain a blockchain-like structure in MANETs. In the next steps, the testbed will be used to measure main performance metrics such as block processing time and transaction processing time. These measurements will be then integrated into a simulation to evaluate the performances of Blockgraph in more complex mobility scenarios and larger scales.

ACKNOWLEDGEMENT

This research was carried out in the Bockchain for Mesh Networks (B4MESH) project funded by the French Ministry of Defense.

REFERENCES

[1] Hakim Badis and Khaldoun Al Agha. 2005. QOLSR, QoS routing for ad hoc wireless networks using OLSR. *European Transactions on Telecommunications* 16, 5 (2005), 427–442.
[2] David Cordova, Alexandre Laube, Thi-Mai-Trang Nguyen, et al. 2020. Blockgraph: A blockchain for mobile ad hoc networks. In *2020 4th Cyber Security in Networking Conference (CSNet)*. IEEE, 1–8.
[3] Your Own Internet device. [n.d.]. . https://www.green-communications.fr/wp-content/uploads/2021/01/YOI-Router.pdf
[4] Damiano Di Francesco Maesa and Paolo Mori. 2020. Blockchain 3.0 applications survey. *J. Parallel and Distrib. Comput.* 138 (2020), 99–114.