Ripple Protocol performance improvement: Small world theory applied to cross border payments

Sebastián Facundo D'Agostino[1], Juan Pablo Timpanaro[1]

[1] Universidad Argentina de la Empresa, Buenos Aires, Argentina
{sebastiandagostino, jptimpanaro}@gmail.com

Abstract. In the context of faster and cheaper international money transfers, this paper makes an analysis of the Ripple Protocol Consensus Algorithm (RPCA) and provides an improved algorithm that takes advantage of the network topology when it holds several sub-networks with poor connections between them. RPCA reaches consensus based on a trusted list of nodes, which this paper theorizes to fit the small world concept from social networks. Network nodes are analyzed statically in order to add new nodes to interconnect clusters. The method used to measure the improvement is through simulation and calculated the algorithm response time with and without the added nodes. This paper achieves a performance improvement between 23 ms and 457 ms for networks of between 50 and 2000 nodes. In the context of high frequency trading, a small milliseconds improvement makes the difference for a transaction to be accepted or not.

Keywords: Ripple Protocol Consensus Algorithm, International Money Transfer, Distributed Systems, Peer-To-Peer Networks, Blockchain, Simulation, Bron-Kerbosch Algorithms.

1 Introduction

Cross-border payments are a general term that refers to payments that are made between two countries and that have some aspect of currency exchange inherent in the process. The settlement between countries can be done using the banking system itself or by correspondent banking.

Cross-border payments are slow, inefficient and costly for banks and companies because there is no single and ubiquitous global payment system. The increase in

global trade and improvements in the efficiency of the supply chain create demand for process improvement [10].

A Blockchain is defined as a distributed and replicated database that allows secure transactions without a central authority. The key element of Blockchain is the absence of a central authority for transaction validation, which is performed by a peer-to-peer network of designated nodes to validate transactions and participate in the consensus process [2].

The widespread adoption of Blockchain technology has the potential to redefine the current technical infrastructure of financial services. The change is expected to bring benefits to existing business processes through the elimination of intermediaries [2]. In addition, the technical aspects underlying Blockchain will provide immutability of data and transactions, resistance against computer attacks and fault tolerance.

One of the Blockchain solutions to cross-border payments problems and challenges is Ripple. Ripple is a peer-to-peer network for digital payments that allow users to use cryptographically signed transactions to store and transfer almost anything, particularly money. In order to send an asset (or a part of it) to someone who is not trusted, then the Ripple network finds a path between the sender and the receiver in such a way that each link is between two trusting parties.

Ripple designed the Ripple Protocol Consensus Algorithm (RPCA) that allows fast convergence and flexibility in network membership, allowing it to operate quickly and inexpensively in a global payment network with well-defined security and reliability properties [12]. The RPCA relies on a network of validating partners to accept or decline a transaction. Each partner takes a decision based on a smaller subgroup of the network that are considered to be trusted. This makes the Ripple network somewhat similar to social networks. The Ripple network has recently overcome the amount of 50 validating partners [4] and is still growing, but the scalability of robustness and performance of RPCA is yet to be seen.

Based on the hypothesis that the topology of the network will be the one of a social network, this article will analyze the small world phenomenon that appears in such networks and provide a static algorithm to improve consensus performance by adding connectivity between validating nodes instead of modifying the protocol itself.

This article will first make an introduction to the cross-border payments problem area, the Ripple network and the small world phenomenon. Then it will propose an algorithm to improve the RPCA performance and use simulation tools to provide a comparison of the effects of the proposed modifications.

2 Ripple network and protocol

Ripple is a federated payment system that uses its own consensus ledger and supports payments almost in real time. It is an open system, but payments are made between groups of private nodes (typically banks) in its Blockchain. Ripple is designed

to integrate with existing banking systems, working with them as an alternative to banking correspondent for cross-border payments. It can also be used between local banks for domestic payments. A system of distributed payments, such as Ripple, must be robust in the face of common failures in systems such as Byzantine (arbitrary or malicious) failures, which can be coordinated and originated from multiple sources in the network [12].

Ripple accounts use public key cryptography to digitally sign transactions. Ripple creates a new currency, called ripples (XRP). Ripple accounts may contain balances in ripples, currencies and other digital assets. Most of the instruments in Ripple are debt-based, except for XRPs and virtual currencies. XRP plays the role of a bridge currency among other digital assets in the Ripple network [6]. The Ripple protocol offers a built-in distributed exchange. Any user can place orders on the ledger and the transactions are processed automatically. The distributed exchange of Ripple can be used to route payments where different currencies are used at both ends [6].

The main components of the Ripple protocol are the following:

- Server: Is any entity that runs the server software that participates in the consensus process [12].
- Ledger: It is a record of the amount of currency in each user account and represents a floor of truth for the network [12].
- Last closed ledger: It is the most recent ledger that has been ratified by the consensus process and represents the current state of the network [12].
- Unique list of nodes (UNL): Each server maintains a UNL, which is a set of other servers that are consulted when determining consensus. Only the votes of the other members of the UNL are considered when determining the consensus (and not the entire network nodes). Thus, the UNL represents a subset of the network that when taken together, is taken as trusted by the server so that they do not conspire with the intention of defrauding the network. [12].

The Ripple Protocol Consensus Algorithm (or RPCA) is applied every few seconds by all the nodes, in order to maintain the accuracy and agreement of the network. Once consensus is reached, the current ledger is considered closed and becomes the last closed ledger. Assuming that the consensus algorithm is successful, and that there are no forks in the network, the last closed ledger maintained by all the nodes in the network will be identical.

The RPCA proceeds in the rounds. In each round:

1. Initially, each server takes all the valid transactions received prior to the start of the consensus round that have not yet been applied (these can include new operations, pending transactions of a previous consensus process, etc.), and makes them public in a list known as the set of candidates.
2. Each server then amalgamates the candidate sets of all the servers in its UNL and votes on the veracity of all transactions.
3. Transactions that receive a percentage greater than the minimum number of positive votes go to the next round, if it exists, while transactions that do not

receive enough votes are either discarded, or are included in the set of candidates for the beginning of the consensus process in the next ledger.

4. The final round requires a minimum percentage of 80% of the UNL of a server agreeing on a transaction. All transactions that meet this requirement are applied in the ledger, which is then closed, becoming the last closed ledger.

One of the greatest challenges of the algorithm is to prevent forks, which occur when two disjoint sets of nodes reach consensus independently.

The agreement is defined as the requirement to maintain a single global truth in a decentralized system. Since the UNLs for each server can be different, the agreement is not inherently guaranteed by the accuracy condition.

For example, if there are no restrictions on the number of members of the UNL, and the size of the UNL is not greater than 0.2 * n_{total} where n_{total} is the number of nodes in the entire network, then a fork is possible. This is illustrated in Figure 1.

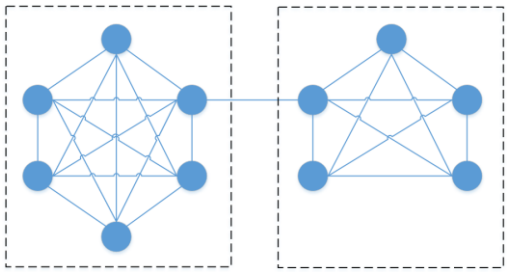

Fig. 1. Graph with two different cliques that reach different agreements

A clique is a set of nodes, where each node of the UNL is the same set of nodes. In Figure 1 it is possible to identify two cliques enclosed in dashed lines. These two cliques do not share any members, so it is possible for each to achieve a correct consensus independently, without having a network agreement.

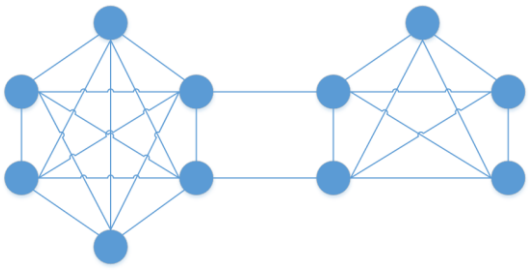

Fig. 2. Graph in which the possibility of cliques decreases due to greater connectivity

If the connectivity of the two cliques exceeds 0.2 * n_{total}, as in Figure 2, a fork is no longer possible because the clique disagreement would prevent the consensus from being reached at the required threshold of 80%.

In graph theory, a clique is defined as a subset of vertices of an undirected graph such that its induced subgraph is complete; that is, that each pair of different vertices is adjacent to the clique. The task of finding a clique of a certain size in a graph is called the clique problem, which consists in finding the maximum clique (the largest). This problem is of NP-complete computational complexity and is solved in exponential times [9].

The clique problem arises in the environment of social networks, where the nodes of a graph represent people and the edges represent the mutual knowledge between them. A clique represents a subset of people in which everyone knows each other.

3 The small world phenomenon

In social or friendship networks a small world phenomenon manifests itself. The idea of the small world comes from thinking that no matter how big the world (modelled as network or a graph), it is perceived as small because of the small distance between friends and friends of friends through whom anyone can meet any other person in the world. Studies on this subject are generally empirical.

The way in which RPCA defines the UNL shows similarities with social networks and this phenomenon because it is based on trust relationships between validating nodes. The original calculations and simulations for the RPCA used a random graph to justify the convergence [12].

In order to model real-world networks, graphs must have both grouping and small-world properties. The random graphs show the effect of the small world but do not show grouping (cliques). The opposite of a random graph is a completely ordered grid or mesh. If each node is connected to z nodes closest to it, it is easy to see that most of the immediate neighbors are also neighbors to each other, which shows grouping properties [11].

Watts and Strogatz have proposed a model for small-world networks, which fits well with general intuitions about the nature of social networks. This model is essentially a regular mesh with a certain degree of randomness in it to produce the small world effect. The Watts-Strogatz model defines two main ideas:

- Homophily: It is a principle that connects a node with others that are similar.
- Weak ties: Are links or relationships that connect a node with parts of the network that would otherwise be too distant. This type of ties are those that connect cliques and are generally random.

4 Related work

Ripple consensus protocol is quite recent, its adoption is yet limited and there has not been a lot of study on the matter but there are some overviews and reviews focusing on some aspects of the protocol and the network.

Frederick Armknecht and others do an analysis of forking in Ripple providing the necessary conditions for forking and a minimum intersection size for the UNL of different validator nodes [1]. They also analyze the transactions trend over time.

Peter Todd reviews the protocol implementation, analyzes the choice of the starter UNL and the possible attack scenarios in case the default validating nodes are modified [13]. Among attack scenarios, the more remarkable are: consensus split (which exploits different software versions on the serves) and coercion of validators (which explores the possibility of having a bigger number of faulty server nodes).

This paper takes from both papers the idea that the UNL is the key part of the protocol and in order to make RPCA scalable over time, it is required to improve the connectivity of the network with the UNL. The topology could be an issue in case of an attack, so this paper looks to innovate by theorizing that the Ripple network will grow according to a social network pattern.

5 Proposal

This paper proposes to use an alternative model of the Watts-Strogatz model as shown in Figure 3 to prevent the formation of cliques in the Ripple network in order to improve the overall grouping of the nodes in the network. To use such a model, it is required to add new validating nodes, which will be referenced as shortcuts, to connect highly clustered subgraphs and thus avoid attacks to the consensus.

The effect of including new nodes connecting cliques in the Ripple network will be analyzed. For such cases, the convergence time is expected to be shorter than not having these new nodes. Adding such nodes does not modify the protocol definition at all.

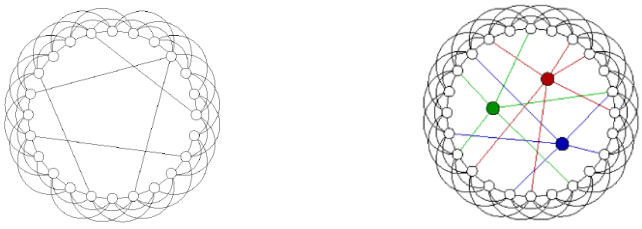

Fig. 3. The graph from the left shows the Watts-Strogatz model and the graph from the right shows an alternative model, in which there is a small number of individuals connected with many others widely distributed [11]

The algorithm proposed consists of the following steps:

1. Search the cliques of an existing network
2. Find the nodes that are not in the intersection of the cliques found in step 1
3. Create N nodes whose UNL (and transmission links list) are the nodes found on step 2

In order to generate the necessary characteristics for a new validator node, it is necessary to load the data of the topology of the network, by means of some computational method for its processing.

The algorithm is supposed to run offline and the data from the network configuration is processed in a batch process. The algorithm starts, for example from the network shown in Figure 4, where two large groups of nodes communicate with each other through bridge links.

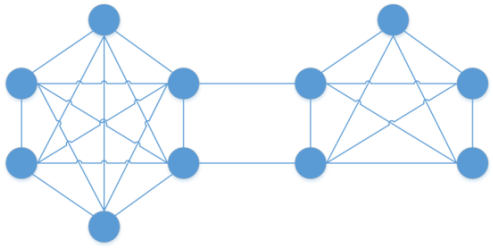

Fig. 4. Network with two large cliques (one with 6 nodes on the left side and 5 nodes on the right side). These cliques are connected by two bridge links.

The first step of the algorithm is the search of the existing cliques in the network. For this purpose, the Bron-Kerbosch algorithm [3] is used. Starting from Figure 4, the result of applying Bron-Kerbosch can be seen in Figure 5.

The cliques found are:

A: [1, 2, 3, 4, 5, 6] B: [7, 8, 9, 10, 11] C: [5, 8] D: [6, 7]

The second step of the algorithm is to find the nodes that do not intersect the sets of nodes obtained from the first step. From observing Figure 5, the intersections of the sets are noted:

$A \cap C = 5$ $A \cap D = 6$ $B \cap C = 8$ $B \cap D = 7$ $B \cap A = \emptyset$

To find the nodes that do not intersect, which will be identified in this case as E, the intersections to the union of all the sets are subtracted:

$E = (A \cup B \cup C \cup D) - (A \cap C) - (A \cap D) - (B \cap C) - (B \cap D) - (B \cap A)$

$E = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] - [5] - [6] - [8] - [7]$

$E = [1, 2, 3, 4, 9, 10, 11]$

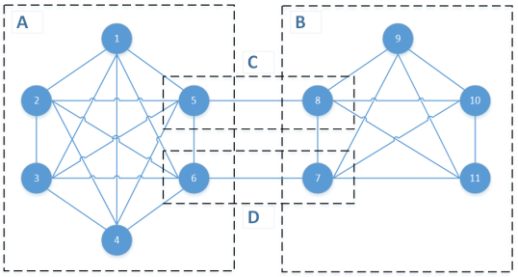

Fig. 5. Network with four cliques identified with the letters A, B, C, D according to the Bron-Kerbosh algorithm [3]

The third and last step of the algorithm is the creation of a new node whose UNL is the list of nodes obtained in the previous step.

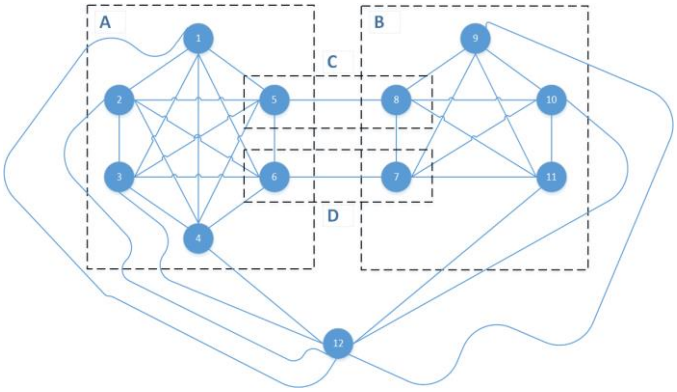

Fig. 6. Result of adding a new validator node with connections to the nodes of the cliques that do not intersect cliques.

Continuing with the example network used for the two previous steps, in the third step a new node is built, which we identify with the number 12, whose UNL is the set E = [1, 2, 3, 4, 9, 10, 11] obtained from the previous step. This new node is added to the network, or an N number of nodes with this configuration to provide greater redundancy in the network and better convergence performance based on the results thrown by the simulations. Figure 6 shows how the network would be by adding only a new node 12. In addition to adding the nodes from E to the UNL, the random set of the same size of E is added as links to which to transmit the information on the network (transactions to vote).

5 Experiments

In order to determine if the proposal presented in this paper effectively generates improvements in the protocol, it is necessary to use a simulation tool. Such a tool is provided by the developers of Ripple publicly, but for cases of networks with nodes

distributed randomly. To perform tests on networks with different topologies, it is necessary to adapt this simulation tool to handle these cases. Thus, a fork from that project was carried out in [8]. Besides that, network distributions with cliques are created with another software tool [7] and put into the simulator.

In order to compare different convergence times of the Ripple protocol, the network data will be analyzed statically (not real time) in each case. Under these conditions, the configuration of a new node (or more) that improves connectivity will be built. This means that any new node will add redundant links in the network.

There are different test cases that explore different situations:

- Network sizes (number of nodes N - 50, 500, 1000, 2000)
- Number of cliques
- Number of validator nodes to add

There are many variables to consider, but only the network size, the distribution according to the cliques and the number of validating nodes to be added will be analyzed. The rest of the variables will be considered parameters under the ceteris paribus principle.

The node size choices are not arbitrary. The size of 50 is approximately the current size of the Ripple network [4], the 1000 size is the same one used in the original Ripple paper [12] and 2000 is the simulation tool limitation due to memory and CPU usage. For bigger network sizes, it will be required to continue the re-engineering of the simulation tool.

The objective in all cases is to measure the convergence of the algorithm in time (or number of cycles). It will be possible to make a comparison of the convergence between the cases of networks with cliques and without cliques (with improvements).

The simulation starts in disagreement for a transaction, with one half of the nodes voting positive and the other half voting in negative. This condition was imitated from the original Ripple simulation process. Then the consensus process is executed. For each round, the result is printed both by the number of positive votes (loyal) and the number of negative votes (failed or conspiratorial). Consensus is reached by reaching the parametric percentage of consensus (80% of supermajority).

It is also important to emphasize that for each node and each link it is necessary to model a latency. In this work it is considered random within a given range, but always generated with the same seed to avoid using different latencies for the same examples.

In the different runs of simulations, the size of the network (identified with N) is increased. For each network size, the simulation is performed with the same network topology with many clusters and the simulation is run again by adding the nodes of the proposed algorithm. In all cases, a number of nodes equivalent to 8% of the original network is added. The first simulations that are run are those of 11 nodes. It is a case that serves to notice that 8% (1 node) for such a small network has a great incidence.

The graph on the left side of Figure 7 shows the evolution of the consensus percentage on the y axis and the convergence time in milliseconds on the x axis. An improvement of approximately 200 milliseconds per transaction is achieved. The curve with circle highlighted points corresponds to the simulation without nodes addition and the curve with triangle highlighted points corresponds to the simulation with the added node.

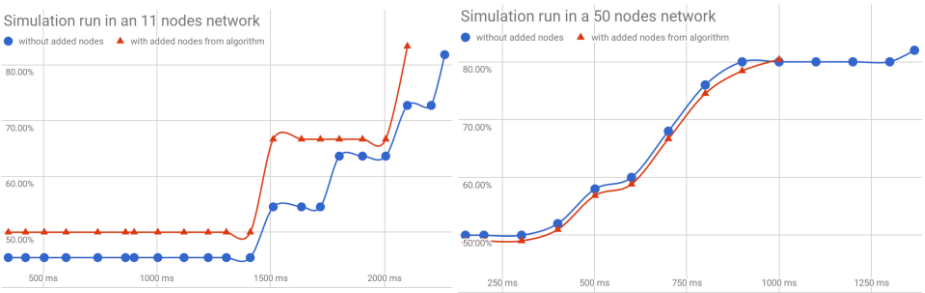

Fig. 7. Graph of simulation runs output of 11 nodes network (left) and 50 nodes network (right)

Consecutively, a network of 50 nodes is simulated. On the graph on the right side of Figure 8 a difference of 400 milliseconds can be appreciated. The simulation without the added nodes cannot overcome the 80% threshold until 400 milliseconds later. The incidence of a conflictive node affects convergence here. The graph shows how simulation without added nodes is limited to 80% of the consensus limit and does not exceed the 20% of the required tolerance.

In Figures 8 it can be observed that the networks with 500 and 1000 nodes respectively have a more marked convergence from 60%. The difference in performance is in favor of the proposed algorithm, but with different results. In the case of the network of 500 nodes, the improvement is only 50 milliseconds per transaction.

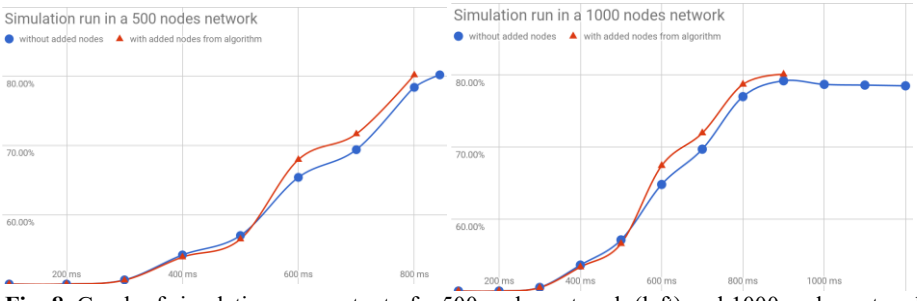

Fig. 8. Graph of simulation runs output of a 500 nodes network (left) and 1000 nodes network (right)

In the case of 1000 nodes it would seem that the difference would have been similar, but convergence is obtained with the improvement algorithm while there is divergence in the case without improvement. This divergence is probably an error in experimentation parameters because it is hard to choose such parameters to produce networks with fixed clique size. For keeping parameters fixed, it is a time that could not be obtained for this set of tests.

Figure 9 shows the graph with 2000 nodes, where the curve of the improvement algorithm accelerates a little before reaching 60%, but nevertheless an improvement of just 20 milliseconds is achieved.

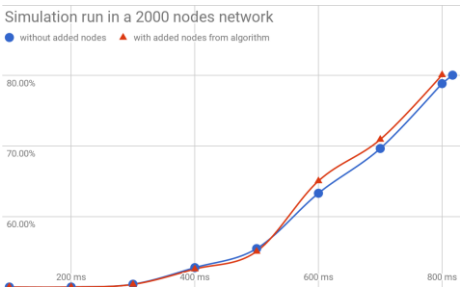

Fig. 9. Graph of simulation runs output of a 2000 nodes network

Finally, Table 1 shows the times in which consensus is reached in each of the simulations with and without adding nodes from the algorithm proposed in this paper.

Table 1. Convergence times of simulation runs with and without aggregate nodes

Node count	Convergence time without added nodes (ms)	Convergence time with added nodes (ms)	Improvement time with added nodes (ms)	Time reduction (%)
11	2264 ms	2026 ms	238 ms	11 %
50	1367 ms	910 ms	457 ms	33 %
500	910 ms	789 ms	66 ms	7 %
1000	(diverges)	820 ms	-	-
2000	817 ms	794 ms	23 ms	3 %

7 Conclusions

The results show that the convergence and performance of the algorithm vary according to the topology of the network. Considering a topology model, optimizations can be generated by adding connectivity without making changes in the protocol. In the case of smaller networks, the improvement is inversely proportional to those of larger size.

Taking advantage of the topology to earn between 23 and 457 milliseconds is a notable improvement of the convergence metric for the highly competitive environment in which transfers occur. Ripple's debt model may involve financial assets transactions or cross-border transfers. Particularly in the case of assets, there are currently cases in the New York Stock Exchange of high frequency negotiations in which a delay of 30 milliseconds can give advantage to an investor with lower latency without

losing money for transaction commissions [5]. Reducing the transaction time can prevent many transactions from being frustrated in favor of unfair competitors.

The need to explore the possibilities of the RPCA algorithm with different network topologies motivated the modification of the original simulation tool to make it more flexible for further practical analysis of the protocol using datasets.

This paper leaves open questions to be resolved in future lines of work, such as:

- Analyze the impact of the use of a different strategy in the addition of nodes
- Analyze the impact of node and link latencies variation in the simulations
- Improve the simulation tool even further and analyze the behavior of networks with more than 2000 validating nodes
- Analyze the introduction of the algorithm proposed in this article as part of the protocol itself, using for example node ranking functions, to improve the performance dynamically.

References

2. Armknecht F, Karame G, Mandal A, Youssef F, Zenner E (2015) Ripple Overview and Outlook https://pdfs.semanticscholar.org/cc9b/027bd1b8c0ee854992bb56e64d72ebf3355e.pdf Accessed 7 March 2017
3. Biella M, Zinetti, V (2016) Blockchain Technology and Applications from a Financial Perspective https://www.weusecoins.com/assets/pdf/library/UNICREDIT%20-%20Blockchain-Technology-and-Applications-from-a-Financial-Perspective.pdf Accessed 11 Aug 2016
5. Conte A (2013) Review of the Bron-Kerbosh algorithm and variations. http://www.dcs.gla.ac.uk/~pat/jchoco/clique/enumeration/report.pdf Accessed 27 Jul 2017
6. Del Castillo M (2017) Ripple's Distributed Ledger Network Passes 50-Validator Milestone https://www.coindesk.com/ripples-distributed-ledger-network-passes-50-validator-milestone/ Accessed 17 Sep 2017
7. Duhigg C (2009) Stock Traders Find Speed Pays, in Milliseconds http://www.nytimes.com/2009/07/24/business/24trading.html Accessed 29 Oct 2017
8. Franco P (2014) Understanding Bitcoin: Cryptography, Engineering and Economics. Wiley.
9. Github repository (networks generator): https://github.com/sebastiandagostino/graph-builder
10. Github repository (simulator): https://github.com/sebastiandagostino/ripple-simulator
11. Pardalos P. Xue J (1992) The Maximum Clique Problem http://www.dcs.gla.ac.uk/~pat/jchoco/clique/indSetMachrahanish/papers/The%20Maximum%20Clique%20Problem.pdf Accessed 24 Apr 2017
12. Park Y (2006) The Inefficiencies of Cross-Border Payments: How Current Forces Are Shaping the Future http://euro.ecom.cmu.edu/resources/elibrary/epay/crossborder.pdf Accessed 25 Aug 2016
13. Prizmic J (2003) Models of the Small World http://www-f1.ijs.si/~rudi/sola/prizmic.pdf Accessed 30 Apr 2017
14. Schwartz D, Youngs N, Britto A (2014) The Ripple Protocol Consensus Algorithm https://ripple.com/files/ripple_consensus_whitepaper.pdf Accessed 18 Aug 2016
15. Todd P (2015) Ripple Protocol Consensus Algorithm Review https://raw.githubusercontent.com/petertodd/ripple-consensus-analysis-paper/master/paper.pdf Accessed 15 Nov 2016