Transitive Network: Tokenless IOU Credit Network in Ethereum

Adithya Bhat[1], Pedro Moreno-Sanchez[2], and Aniket Kate[1]

[1] Purdue University
[2] TU Wien

Abstract. Transitive Network aims to build a completely decentralized scalable tokenless IOweYou (IOU) credit network as a smart contract in Ethereum. Transitive Network imposes no barrier to entry and provides a cost effective way to perform payments in a credit network. Ethereum platform provides additional benefit of building state channels that allow further reduction of costs for users. We compare performance of Transitive Network with Ripple and observe significant cost improvements in most transactions and comparable low costs for path-based payments.

1 Introduction

A credit network represents flexible and transitive trust among users in terms of credit allocations between them. In fact, a credit network can be represented as a directed graph where nodes denote individual users and weighted edges represent credit links between individuals. For example, assume that Alice owes Bob $50, this can be represented as a directed edge from Alice to Bob with a weight of $50. By extending this to several users and links, it is possible to build a more complex credit network that better represents today's financial relations.

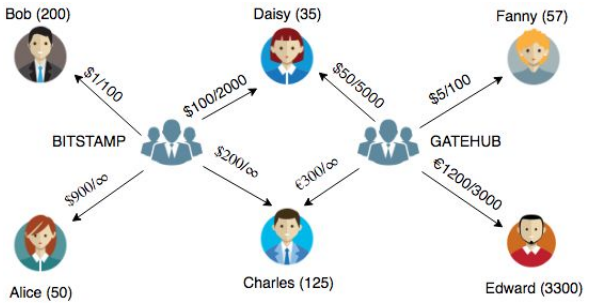


Fig. 1: An illustrated example of Ripple network. The number in brackets next to the user represent the amount of XRP owned by the user. Each link consists of a number a/b, where a is the amount currently owed, b is the upper bound.

A credit network created as aforementioned enables transitive payments between users. An illustrative example is shown in Figure 1. If Bob wants to

pay Fanny $50, he can use the path: Bob→Bitstamp→Daisy→Gatehub→Fanny. Moreover, credit on the links can be defined in different currencies so that cross-currency transactions are also supported. In fact, the core idea of credit network has been found useful in a plethora of applications [1, 2, 6].

The Ripple network is a widely deployed blockchain technology that uses the notion of credit network as its core to enable path-based, same- and cross-currency transactions among their users. At the time of writing, it contains at least 1033 nodes[4] and 13892 links[5] that serves a financial instrument mostly focused on the benefit for the financial institutes worldwide.

2 Motivation and Application

In this project, we question two of the key pillars of the Ripple network: the Ripple consensus protocol and the need for XRP, Ripple own's native currency introduced in the system. First, Ripple has considered its own consensus protocol and no complete, formal description is available, let alone proper security analysis. Moreover, its current implementation has lead to consensus protocol executed by a few validators, being thus far from a decentralized system.

Second, the Ripple native cryptocurrency XRP was introduced in principle as a denial of service mechanism (e.g., every transaction must pay a fee in XRP). However, other actions such as creating an exchange offer or a new link between two nodes also require a non-trivial amount of XRP (called reserve) and that is set to 5 XRP per link and 20 XRP per node at the time of writing. Given the current prices of cryptocurrencies, this puts a high cost for users.

These two main drawbacks of Ripple call for an alternative implementation and this is the main motivation in this project. In this project, we implement Transitive Network[3], a tokenless IOU credit network in Ethereum that overcomes these challenges as follows.

First, it relies on the well-studied, distributed consensus protocol of Ethereum. Second, it does not add any token (e.g., XRP). Instead, the focus of Transitive Network is to reduce the costs for the credit network operation.

By achieving these goals, Transitive Network will bring for the first time the credit network business over Ethereum in a seamless manner.

3 Transitive Network

Transitive Network implements the set of functions required for a fully functional credit network. In particular, we consider the following operations (The operations in parenthesis represent the equivalent operation in Ripple) Add Node (Create Wallet), Create Link (Trust Set), Update Link(Trust Set), Add Offer (Create Offer) and Credit Network Pay (Payment).

3.1 Challenges Faced

Computation and storage cost: Some of the key challenges faced involve minimizing the gas costs and storage. For example, transitive network uses the

SHA3 hash of the link data structure as the ID for the link. The contract provides a helper function to recompute the ID of the link if the currency ID and the end users are specified. This helps to reduce the cost of computation to search and find the link data structure if the ID can be directly provided. The assumption is that the end user does not directly interact with the smart contract and has helper programs and APIs that help users to interact with the smart contract. The smart contract also provides similar helper functions to find the offer ID.

Storage of boolean values in an Ethereum smart contract is inefficient as it naively takes a byte of storage per boolean. As storage is several orders of magnitude more expensive than computation, our optimization consisted in storing several flags in a single unsigned integer, thus saving important bytes in the smart contract.

Flexibility and extensibility of the smart contract: In order to make the transitive network smart contract flexible and extensible, the contract logs events whenever the state of the graph changes. This helps interested parties to build a copy of the graph online and keep it up to date easily. Interesting services such as path finding or cheapest currency transfer routing can be offered easily on top of the smart contract.

Another challenge faced in implementing the credit network on Ethereum is that it is not flexible in terms of cryptographic algorithms in the sense that one must implement the scheme with the provided cryptographic primitives and cannot change or swap one algorithm for another for efficiency and security reasons without increasing the cost to the users. We solve this by reducing dependence on the external algorithms which can be achieved by helper functions and log functions. For example, the smart contract logs the state of the graph and this can be used by any interested third party to rebuild the graph offline and use it to find paths in the graph for payments as a stand alone service or for personal reasons.

Lack of floating point support: An interesting challenge faced when developing exchange rates is the lack of floating point support for Solidity. This leads to rounding errors, for instance, in the computation of the final payment amount after applying the exchange rate for a certain currency. This can be solved by taking the exchange rate as input in the rational integer form. That is any rational number can be expressed as a number p/q where p and q are integers. The contract then applies the exchange rate by multiplying the amount with p and then finally rounding the number obtained after dividing by q.

4 Results

The smart contract is evaluated in terms of the gas it costs to run the functions. Helper functions are usually functions that do not change the state and just query the state to view contents. These do not cost any gas to the user.

The Table 1 shows the amount of gas consumed for the important contract functions. The fees are computed based on the gas price that is considered sufficient as of the date of snapshotting the fees[3].

Function	**Gas Used**	**Min Fee** [3][4]	**Avg Fee**[5]	**Max Fee**[6]
Create Wallet(AddNode)	43474	0.0089	0.0177	0.0354
Create Link	117990	0.0240	0.0481	0.0962
Update Link	37263	0.0076	0.0152	0.0304
Pay[7](H_0C_0)	45586	0.0093	0.0186	0.0372
Pay(H_1C_0)	59438	0.0121	0.0242	0.0484
Pay(H_2C_0)	73226	0.0149	0.0298	0.0597
Pay(H_3C_0)	87014	0.0177	0.0355	0.0709
Pay(H_1C_1)	64150	0.0131	0.0261	0.0523
Pay(H_2C_1)	77939	0.0159	0.0318	0.0635
Pay(H_3C_1)	91728	0.0187	0.0374	0.0748
Create Offer(AddOffer)	74015	0.0151	0.0302	0.0603
Cancel Offer	51571	0.0105	0.0210	0.0420

Table 1: Gas Usage for the contract functions

The costs of executing similar operations in Ripple are documented in the Table 2. The table provides an overview of the cost to create wallet which is the cost to create in the credit network graph, and also the costs to create links and offers. It can be clearly noted that the most expensive component is the reserve cost that is the amount of XRP that gets locked in an account. The locked XRP cannot be used for any other purposes.

Function	**Reserve Cost**[8]	**Tx Cost**[8]	**Total Cost**[8]	**USD Cost**[9]
Create Wallet	20	0.00385	20.00385	9.1274
Create Link	5	0.00385	5.00385	2.2829
Create Offer	5	0.00385	5.00358	2.2829
Path-based Tx	0	0.00385	0.00385	0.0018

Table 2: Costs associated with Ripple

The Table 3 compares the costs of performing operations in Ripple and the costs of performing the same operation using the smart contract. The choice of H_3C_0 to represent a path based transaction in Ripple is because most of the payments in Ripple do not exceed more than 3 links.

[3] USD/ETH High rate as on November 1, 2018 is $203.75/ETH taken from https://coinmarketcap.com/currencies/Ethereum/historical-data/

[4] Minimum Gas Price accepted on the network is 1GWei/gas

[5] Average Gas Price of the transactions accepted in the network is 2GWei/gas

[6] Gas Price for the fastest acceptance of the transaction is 4GWei/gas

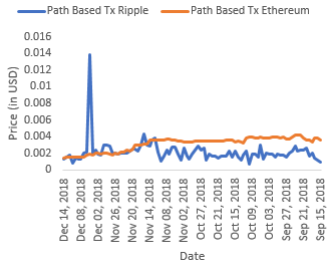
[7] H_{number} stands for the number of hops, C_{number} stands for the number of currency conversions

[8] In XRP

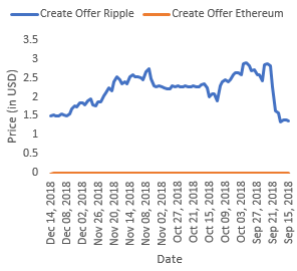
[9] USD Cost is Highest Price on Nov 1 times XRP. That is, 0.456303*Total XRP Cost

Ripple Function	Ripple Cost[10]	Eq TNet Function[11]	TNet Cost[12]
Create Wallet	9.1261	AddNode	0.0177
Create Link	2.2815	CreateLink	0.0481
Create Offer	2.2815	AddOffer	0.0302
Path-based Tx	0.0018	Pay(H_3C_0)	0.0355

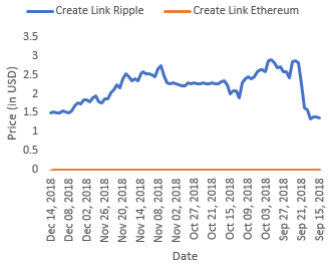
Table 3: Comparison of costs in Ripple vs Transitive Network



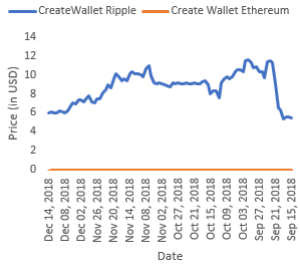
(a) Payment Cost Comparison



(b) New Offer Cost Comparison



(c) New Link Cost Comparison



(d) Wallet Creation Cost Comparison

Fig. 2: Comparison of operation costs between Ripple and Ethereum

For more fine grained comparison of costs, average prices of both Ripple and Ethereum are chosen and the costs for the different operations are computed and shown in Figure 2d, 2a, 2b and Figure 2c.

From Figure 2, it can be clearly seen that operations in Ripple are more expensive than the operations on Ethereum. This is mainly due to the reserve cost of 20 XRP for a new wallet and 5 XRP for the others. This reserve cost adds additional costs in Ripple. Figure 2a is essentially a comparison of trans-

[10] (in USD)
[11] Equivalent Transitive Network Function
[12] Transitive Network Cost (in USD)

action fees in Ripple and Ethereum and it is can be observed that they are very inexpensive and comparable.

Figure 2a compares the cost to perform a payment in Ripple with the cost to perform a payment in Ethereum. It can be seen that the costs are low in both the cases and are comparable to each other.

Figure 2b compares the cost to create currency exchange offer in Ripple with the cost to create a currency exchange offer in Ethereum. It can be clearly seen that the cost to create an offer is more expensive than in Ethereum.

Figure 2c illustrates the difference in costs of creating a link in Ripple with the costs of creating a link in Ethereum. It can be clearly seen that it is expensive to create offers in Ripple.

In Figure 2d, the cost to create a node in the credit network is compared between Ripple and Ethereum for a period of three months. It can be clearly seen that Ripple is very expensive in terms of cost to join the credit network.

5 Conclusion and Future Work

Transitive Network demonstrates that it is possible to build cost-effective credit networks on decentralized blockchains without the invention of a new token. We observe that the costs incurred are much lower or comparable to those of Ripple. We also extend and improve the functionalities of the credit network and offer better features such as atomic multi-path payments.

In the presence of state channels, we can further reduce the average case cost of transactions of the credit network. To the best of our knowledge this is the first ever implementation of IOU credit network in a decentralized smart contract setting.

References

1. Ansley Post, Vijit Shah, and Alan Mislove. 2011. Bazaar: strengthening user reputations in online marketplaces. In Proceedings of the 8th USENIX conference on Networked systems design and implementation (NSDI'11). USENIX Association, Berkeley, CA, USA, 183-196.
2. Alan Mislove, Ansley Post, Peter Druschel, and Krishna P. Gummadi. 2008. Ostra: leveraging trust to thwart unwanted communication. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI'08), Jon Crowcroft and Mike Dahlin (Eds.). USENIX Association, Berkeley, CA, USA, 15-30.
3. Transitive Network: A tokenless layer-2 network for Ethereum, https://github.com/pedrorechez/transitivenetwork
4. Rippled Network Topology | XRP Charts. https://xrpcharts.ripple.com/#/topology
5. Inc., R. Ripple Data API v2, 2018. https://ripple.com/build/data-api-v2/ .
6. Inc., R. Ripple Website, 2018. https://ripple.com .