

# Measurements, Analyses, and Insights on the Entire Ethereum Blockchain Network

Xi Tong Lee  
Nanyang Technological University  
b150049@e.ntu.edu.sg

Arijit Khan  
Nanyang Technological University  
arijit.khan@ntu.edu.sg

Sourav Sen Gupta  
Nanyang Technological University  
sg.sourav@ntu.edu.sg

Yu Hann Ong  
Nanyang Technological University  
ongy0121@e.ntu.edu.sg

Xuan Liu  
Nanyang Technological University  
xuan002@e.ntu.edu.sg

## ABSTRACT

Blockchains are increasingly becoming popular due to the prevalence of cryptocurrencies and decentralized applications. Ethereum is a distributed public blockchain network that focuses on running code (smart contracts) for decentralized applications. More simply, it is a platform for sharing information in a global state that cannot be manipulated or changed. Ethereum blockchain introduces a novel ecosystem of human users and autonomous agents (smart contracts). In this network, we are interested in all possible interactions: user-to-user, user-to-contract, contract-to-user, and contract-to-contract. This requires us to construct interaction networks from the entire Ethereum blockchain data, where vertices are accounts (users, contracts) and arcs denote interactions. Our analyses on the networks reveal new insights by combining information from the four networks. We perform an in-depth study of these networks based on several graph properties consisting of both local and global properties, discuss their similarities and differences with social networks and the Web, draw interesting conclusions, and highlight important, future research directions.

## CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms; Exploratory data analysis**; • **Applied computing** → **Digital cash**.

## KEYWORDS

Blockchain, Ethereum, Smart Contracts, Tokens, Network Analysis

### ACM Reference Format:

Xi Tong Lee, Arijit Khan, Sourav Sen Gupta, Yu Hann Ong, and Xuan Liu. 2020. Measurements, Analyses, and Insights on the Entire Ethereum Blockchain Network. In *Proceedings of The Web Conference 2020 (WWW '20)*, April 20–24, 2020, Taipei, Taiwan. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3366423.3380103>

## 1 INTRODUCTION

Blockchain has taken the world by storm with its proliferation in an astoundingly diverse array of applications over the last few years.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WWW '20, April 20–24, 2020, Taipei, Taiwan

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7023-3/20/04.

<https://doi.org/10.1145/3366423.3380103>

Numerous cryptocurrencies and a plethora of business use-cases paved the way for inclusive decentralization and consensus-driven automation through innovative incentives and smart contracts. It has been more than ten years since Bitcoin [54] introduced the era of decentralized community-controlled currency. Since then, several variants have been introduced on the premise of a specific design philosophy or targeted at a specific application. Ethereum [72] paved a new way by introducing an automation layer on top of a permissionless blockchain fabric, through the use of complex smart contracts executed by a decentralized network. This led the development of blockchain into its current avatar, where decentralized applications are written on the framework of blockchain networks like Hyperledger, Corda, Ripple, Stellar, and many more.

While Bitcoin-like cryptocurrency networks concern themselves only with users (wallets) transacting over blockchain, Ethereum-like blockchains present a decentralized computing environment. Ethereum is a transaction-based state machine, where the state is made up of *accounts*. Transfer of value and information between accounts cause transitions in the global state of Ethereum, which are recorded in the blockchain [75]. There are primarily two types of accounts: (a) User accounts, controlled by external users with their private keys, and (b) Contract accounts, controlled by contract codes that behave like ‘autonomous agents’. Transactions in Ethereum are data packets sent by the user accounts, signed with their private keys, while Messages in Ethereum are virtual objects produced by contract accounts, generally sent to other contracts.

In addition to the native unit of value *ether*, Ethereum blockchain allows creation of Tokens, an abstraction of “digital assets”, with the help of suitable data structures and methods implemented through smart contracts. Similar to base transactions using ether, accounts in Ethereum may transact in tokens of various kinds, fungible or otherwise, through the appropriate smart contracts. This allows for a rich ecosystem of tokens, including various ERC20 (fungible) and ERC721 (non-fungible) tokens, to thrive on Ethereum blockchain.

**Motivation:** The genre of blockchain introduced by Ethereum brings forth a fascinating ecosystem of humans and autonomous agents (smart contracts), cohabiting the underlying blockchain fabric. It is neither like conventional social networks, where the players are human users, nor like the cryptocurrencies, where all interactions are transfer of value or asset. In essence, a blockchain network is closer to the Internet or Web, where users are allowed to interact with one another, as well as with programs. However, different from Web, there is also an interaction framework for smart contracts, where they can call (or kill) each other to maintain and advance the

global state of the blockchain. This motivates us to study a public permissionless blockchain network as a complex system. We choose Ethereum, the most prominent public permissionless blockchain, to measure and draw insights from the network interactions.

**Contributions:** Our main contributions are as follows.

- To the best of our knowledge, we are the first to conduct a comprehensive study of the large-scale Ethereum blockchain network, cohabited by both human users and autonomous agents (smart contracts). We investigate their complex interactions by constructing four networks from the entire Ethereum blockchain data, namely TraceNet, ContractNet, TransactionNet, and TokenNet (§3).

- We study the four blockchain networks based on local and global graph properties, e.g., network size, density, degree distribution, in-to-outdegree correlation, vertex centrality, reciprocity, assortativity, connected components, core decomposition, transitivity, clustering coefficient, higher-order motifs, articulation points, adhesion, cohesion, and small-world characterization. Such structural information is useful to characterize interactions and evaluate Ethereum-blockchain at scale, an effort that has not been attempted before. We also identify their similarities and differences with social networks and the Web, and draw interesting conclusions (§4.5).

- We further consider three prominent token subnetworks, Bancor, Binance Coin, and Zilliqa, and investigate the amount of activity in the token network over time, as well as the size of the core community driving the token economy over time. We identify interesting correlation between the temporal evolution of the number of cores in the token subgraphs against the corresponding evolution of price of the token in the cryptocurrency market (§6).

- We open source our dataset (<https://github.com/sgsourav/blockchain-network-analysis>) and highlight important research directions such as analysis of mining pools, identifying motifs to detect fraudulent activities, and temporal analysis of token subnetworks to forecast the price of Ethereum backed tokens (§7).

## 2 RELATED WORK

In the last decade, several works explored Bitcoin, cryptocurrencies, and other blockchain networks based on graph theory and network analysis. This line of research gained momentum due to the transparency offered by public permissionless blockchain, which allows anyone to access transactional information on the networks.

**Analysis of blockchain networks:** Graph measurement and analysis of blockchain started with the motivation of de-anonymizing the “pseudonymous” Bitcoin accounts, using heuristics for clustering addresses based on transaction behavior [23, 48]. Similar analysis have since been performed on seemingly “anonymous” cryptocurrencies like Monero [53] and Zcash [39], as well as across a diverse array of cryptocurrency ledgers [77]. While initiatives like Bitfodine [69] and Elliptic [1] perform chain analysis on the Bitcoin transaction network to extract intelligence, researchers have also measured the network characteristics to predict the market-price of Bitcoin [34, 41, 76], or to search for influential patterns [27].

Beyond specific applications, the large-scale network properties of Bitcoin transaction graph has been studied in [37, 62], and the abstraction of any blockchain as a transaction network for analysis has been considered in [6]. Unlike our four interaction networks in the Ethereum blockchain, these works are only about the bitcoin

transaction graph, where all interactions are transfer of value. Ferretti and D’Angelo also studied only transactions in the Ethereum blockchain [29] – not all types of network interactions. Last year, Somin, Gordon and Altshuler investigated the entire address graph spanned by ERC20 token trade in Ethereum blockchain [67], and also studied the social signals in the Ethereum ERC20 token trading network [68]. In their footsteps, Victor and Lüders have recently measured Ethereum-based ERC20 token networks [71].

We follow the direction of [27, 29, 62, 67, 68, 71] to measure and analyse the entire Ethereum blockchain network – above and beyond the financially relevant token transfer layer. Our approach closely follows the norms of measuring and analyzing social networks, Internet, and the Web, as the entirety of the Ethereum blockchain network presents itself as an equally complex system.

**Measurement of social networks, Internet, and the Web:** Sociologists studied various properties of social networks [73]. Milgram showed that people in the United States are connected by short path-lengths, often associated with the phrase “six degrees of separation” [50]. The degree distribution was studied in the context of movie actors network [11], followed by network of scientific collaborations [55] and Web of human sexual contacts [46]. Granovetter argued that social networks can be partitioned into ‘strong’ and ‘weak’ ties, and strong ties are tightly clustered [33]. Adamic et al. studied an early online social network at Stanford, and found that the network exhibits small-world behavior and local clustering [4]. Traces from CyWorld, MySpace, and Orkut were profiled in [5], Facebook in [74], instant-messaging networks in [45], as well as Youtube, Flickr, and Yahoo! 360 were analyzed in [52]. Other studies focused on the evolution and growth of online social networks, including Flickr, LiveJournal, and Yahoo! 360 [10, 42, 51]. Most of these studies reveal that social networks have higher fraction of symmetric links, and also exhibit higher level of local clustering due to community structure. Moreover, past studies confirm that social networks obey power-law scaling [11], and exhibit high clustering coefficients, establishing them as small-world networks [8].

In the areas of Web and Internet, Faloutsos et al. showed that the degree distribution of Internet follows power-law [28]. Siganos et al. demonstrated that the high-level structure of the Internet resembles a “jellyfish” [66]. A prominent study [18] found that the Web consists of a single, large strongly connected component (SCC), and other groups of vertices can either reach the SCC or can be reached from the SCC. Kleinberg [40] demonstrated that high-degree pages in the Web can be classified based on their functions as either hubs or authorities. This phenomenon of the Web is different from social networks, where one observes a high degree of reciprocity in directed user links, leading to a strong correlation between user indegrees and outdegrees [52].

**Measurement of blockchain networks:** Ours is the *first in-depth measurement of the blockchain network across four different layers, consisting of both human users (e.g., regular users, miners) and autonomous agents (e.g., smart contracts), as well as their complex interaction patterns*. We systematically identify and discuss its similarities and differences with various other networks, including social networks and the Web, and draw insights on the unique characteristics of the blockchain networks. We open source our dataset [44] to facilitate further research in this direction.

### 3 DATASETS AND EXPERIMENTAL SETUP

We are interested in all interactions between Ethereum accounts, both in terms of standard ether transactions and token transfers. This requires us to construct *interaction network* from the Ethereum blockchain data, where vertices are accounts (users or contracts) and arcs denote their interactions. There are four major types of interaction between Ethereum addresses – (i) User-to-User (transaction or token transfer), (ii) User-to-Contract (call or kill), (iii) Contract-to-User (transaction or token transfer), and (iv) Contract-to-Contract (create, call, kill or hard fork), as illustrated in Figure 1. In addition, there are some interactions to and from the Null address, denoting creation of smart contracts and generation of ether (mining rewards), respectively. We create four interaction networks.

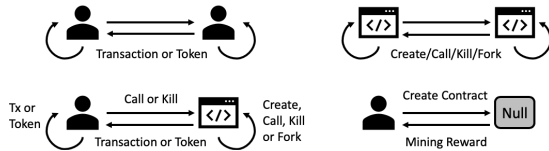


Figure 1: Interactions in the Ethereum Blockchain Network

Table 1: Ethereum Blockchain Data : Block #0 to #7185509

	Approximate Size of Dataset	Row Count
<i>blocks</i>	8 GB	7 185 509
<i>contracts</i>	15.7 GB	12 950 995
<i>transactions</i>	190 GB	388 018 489
<i>traces</i>	500 GB	974 766 498
<i>logs</i>	160 GB	289 552 838
<i>tokens</i>	11.4 MB	126 181
<i>token transfers</i>	58 GB	173 421 940

#### 3.1 Data Extraction and Creation of Networks

Google Cloud BigQuery curates the entire Ethereum blockchain data in terms of blocks, contracts, transactions, traces, logs, tokens and token transfers [2]. We extract all relevant data for Ethereum from the `ethereum_blockchain` dataset under the Google Cloud `bigquery-public-data` repository, till 2019-02-07 00:00:27 UTC, which amounts to all blocks from genesis (#0) up to #7185508. The extracted data consists of seven tables, as summarized in Table 1.

We note that the *blocks* and *transactions* tables are not complete or granular enough for our purpose, as the first stores only block headers, and the second does not consider interaction between smart contracts. Instead, we choose to focus on the *traces* table to extract the comprehensive interaction network for Ethereum addresses (each account has a unique 160-bit address). The *traces* table stores executions of all recorded messages and transactions (successful or not) in the Ethereum blockchain, which are directly mapped to the transaction and block where they belong. The `from_address` and `to_address` recorded in each trace helps us create individual arcs in the interaction network of Ethereum accounts, and it is also possible to group all traces triggered by a particular transaction.

*TraceNet*. We first create TraceNet, with all possible user and smart contract addresses found in the entire blockchain dataset as vertices, and all successful traces with non-null from/to addresses as arcs. We characterize the vertices by their type – regular users, miners / mining pools, regular contracts, and miner / mining pool contracts. This is the most comprehensive interaction network for Ethereum, with various well-classified vertices and arcs, as in Table 2.

Table 2: TraceNet: Complete Interaction Network

	Type	Count
<i>Vertex</i>	User Accounts (Regular Users)	62 851 964
	User Accounts (Miners / Mining Pools)	4 220
	Smart Contracts (Regular Contracts)	12 950 986
	Smart Contracts (Miners / Mining Pools)	9
	Total number of Vertices	75 807 179
<i>Arc</i>	Users to Users	198 479 972
	Users to Smart Contracts	178 170 016
	Smart Contracts to Users	74 196 065
	Smart Contracts to Smart Contracts	317 967 546
	Total number of Arcs	768 813 599

Table 3: ContractNet: Interaction Network of Contracts

	Type	Count
<i>Vertex</i>	Smart Contracts (ERC20)	27 548
	Smart Contracts (ERC721)	891
	Smart Contracts (others)	11 304 311
	Total number of Vertices	11 332 750
<i>Arc</i>	Create a Smart Contract	10 690 445
	Suicide of a Smart Contract	3 677 233
	Call a Smart Contract	303 599 752
	Daofork for a Hard Fork	116
	Total number of Arcs	317 967 546

Table 4: TransactionNet: Network of Transactions

	Type	Count
<i>Vertex</i>	User Accounts (Regular Users)	43 582,566
	User Accounts (Miners / Mining Pools)	4 222
	Smart Contracts (Regular Contracts)	1 940 731
	Smart Contracts (Miners / Mining Pools)	9
	Null Address	1
	Total number of Vertices	45 527 529
<i>Arc</i>	Users to Users	198 480 017
	Users to Smart Contracts	187 245 157
	Users to Null	2 293 315
	Total number of Arcs	388 018 489

Table 5: TokenNet: Network of Token Transactions

	Type	Count
<i>Vertex</i>	User Accounts (Regular Users)	26 930 958
	User Accounts (Miners / Mining Pools)	2 125
	Smart Contracts	3 496 009
	Smart Contracts (Miners / Mining Pools)	7
	Total number of Vertices	30 429 099
<i>Arc</i>	Users to Users	113 127 126
	Users to Smart Contracts	11 598 759
	Smart Contracts to Users	40 265 766
	Smart Contracts to Smart Contracts	8 430 289
	Total number of Arcs	173 421 940
<i>Arc</i>	ERC20 Token Transfer	161 656 811
	ERC721 Token Transfer	5 929 971
	ERC20 and ERC721 Token Transfer	1 540
	Neither ERC20 nor ERC721	5 833 618
	Total number of Arcs	173 421 940

*ContractNet*. The second network, ContractNet, is a subgraph of TraceNet, where we retain the arcs with both `from_address` and `to_address` belonging to smart contracts (verified using the *contracts* table). This provides us with a pure contract-to-contract interaction network on Ethereum, where arcs are direct messages and/or transactions between smart contracts. We observe four arc types in this category – (i) Create arcs that involves creation of new smart contracts, (ii) Suicide arcs where the owner of the smart contract decides to kill the smart contract, (iii) Call arcs that transfer ether from one account to another or call another smart contract, and (iv) Daofork arcs where a hard fork has occurred in the blockchain. These arcs connect three major vertex types – (i) ERC20 token contracts, (ii) ERC721 token contracts, and (iii) other contracts for intermediary functions and token-related services, as in Table 3.

**Table 6: Network characteristics of Ethereum blockchain: vertices and arcs, self-loops, density**

	# Vertices	MultiDigraph			Simple, undirected graph		
		# Arcs	# Self-loops (% of Arcs)	Density	# Arcs	# Self-loops (% of Arcs)	Density
TraceNet	75 807 179	768 813 599	3 036 915 (0.40%)	$1.34 \times 10^{-7}$	191 901 321	178 241 (0.09%)	$0.67 \times 10^{-7}$
ContractNet	11 332 750	317 967 546	<b>2 521 670 (0.79%)</b>	$24.8 \times 10^{-7}$	19 608 452	<b>63 234 (0.32%)</b>	$3.05 \times 10^{-7}$
TransactionNet	45 527 529	388 018 489	515 245 (0.13%)	$1.87 \times 10^{-7}$	128 368 878	115 007 (0.09%)	$1.24 \times 10^{-7}$
TokenNet	30 429 099	173 421 940	326 557 (0.19%)	<b><math>1.87 \times 10^{-7}</math></b>	93 844 445	36 950 (0.04%)	<b><math>2.03 \times 10^{-7}</math></b>

*TransactionNet*. The third network, TransactionNet, is the network of all Ethereum transactions recorded in the *transactions* table. Transactions are made by users, either to other users or smart contracts, or to a Null address in case of smart contract creation. The vertices and arcs of this network is thus similar to that of the TraceNet, with the exception of the Null address as an extra vertex in the network, and the ‘User to Null’ arcs, as shown in Table 4.

*TokenNet*. Finally, we create TokenNet, pertaining only to the transfer of tokens between Ethereum accounts. We use the *token transfers* table to extract only token related transactions in the blockchain (validated using the *tokens* table). The basic types of users and arcs are somewhat similar to that in the TransactionNet, with an additional level of arc characterization based on the token in use, as shown in Table 5. The arcs marked ‘Neither ERC20 nor ERC721’ denote intermediary processes that involve other smart contracts to help facilitate the actual token transfers. Arcs with the same `transaction_hash` belong to the same transaction, leaving us with 108 967 077 distinct transaction hashes in the TokenNet.

**Importance of four blockchain networks:** *Each interaction network provides us with a different perspective on Ethereum blockchain, and our analyses on the networks reveal new insights by combining information from the four networks.* While TraceNet presents a global view of interactions between Ethereum accounts, ContractNet focusses only on the automated multi-agent network of contracts, providing us with a functional view of the Ethereum state machine. While TransactionNet helps us analyze the base ether transactions in the blockchain, TokenNet focusses on the rich and diverse token ecosystem built on top of the Ethereum blockchain.

**Environment setup:** The code is implemented in Python 3.7, and NetworkX [36] is used for analyzing the graph. We perform experiments on a single core of a 100GB, 2.40GHz Xeon server.

### 3.2 Network Characteristics: Vertices and Arcs, Self-Loops and Density

The basic characteristics of our four networks (TraceNet, ContractNet, TransactionNet, and TokenNet) are given in Table 6. We consider two variations of each network. In *MultiDigraph*, we retain all directed arcs between a pair of vertices. Clearly, multiple arcs between a pair of vertices indicate that there exist repeated interactions or transfers between those addresses. In contrast, *Simple, undirected graph* denotes a simplified version where we consider at most one, undirected arc between every pair of vertices. Multiple arcs between vertices are counted only once in the simple graph.

In Table 6, the counts for arcs include self-loops, that is, transactions having the exact same `from_address` and `to_address`. An interpretation would be that an address transacted with itself, which happens in scenarios where users verify if it is possible to send Ether to themselves, or due to a mistake in input of `to_address`.

In Table 6 we provide the self-loop counts and its percentage over all arcs. With more arcs in MultiDiGraphs, it is expected that their self-loop counts would also be higher compared to that in the simple, undirected graphs. We observe that the self-loop percentage in ContractNet’s MultiDiGraph is significantly higher than that in the three other networks. Moreover, the number of self-loops in ContractNet’s MultiDiGraph is almost 40 times than that in its own simple, undirected graph, indicating that a lot of smart contracts make multiple calls to itself. Smart contracts are self-executing programs which may contain multiple functions, and contracts can call functions of other contracts or itself if processing a transaction requires some other functionality within the same contract.

Let the total number of vertices and arcs in a network be  $|V|$  and  $|E|$ , respectively. There are several definitions of network density [32]: *arc density* (average degree), *arc ratio*, *triangle density*, *triangle ratio*, etc. In Table 6, we report *arc ratio*, that is, the number of arcs / the number of possible arcs. For an undirected network, this is  $2|E|/\binom{|V|}{2}$ ; whereas for a directed network, it is  $|E|/\binom{|V|}{2}$ . Since  $|V|$  remains the same over MultiDigraph and simple, undirected graph, and the number of arcs in MultiDigraph is generally more than twice the number of arcs in the simple, undirected graph, the former should be denser than the later one. This is more prominent in ContractNet — its MultiDigraph is about 8 times denser than its simple, undirected graph, which indicates that *there are a lot of multiple arcs between vertices in the MultiDigraph of ContractNet*. It is explainable, as all arcs in ContractNet are between smart contracts. Different smart contracts often make use of some common contracts. Calls to a smart contract may also, in turn, evoke calls to other smart contracts. As smart contracts can realize similar functions, it is possible to have large volume of multiple calls in both directions between smart contracts, resulting in a lot of multiple arcs between vertices in MultiDigraph of ContractNet.

In TokenNet, on the other hand, the simple, undirected graph is denser than its MultiDigraph. The number of arcs in this MultiDigraph is almost twice the number of arcs in the corresponding simple, undirected graph. This means that *there are small volumes of bidirectional arcs between pairs of vertices in TokenNet*. This is because in token transfers, the token contract will update the balance of the two accounts internally, and there will not be an outgoing arc from the token contract to the receiver of the tokens [3].

## 4 LOCAL NETWORK PROPERTIES: MEASUREMENTS AND ANALYSES

Once the blockchain networks are constructed, we extract several features from these networks. Some of these features represent global properties of a network — also known as “summary features”. Others represent local properties of individual vertices; we refer to them as “local features”. The local properties are analyzed below.

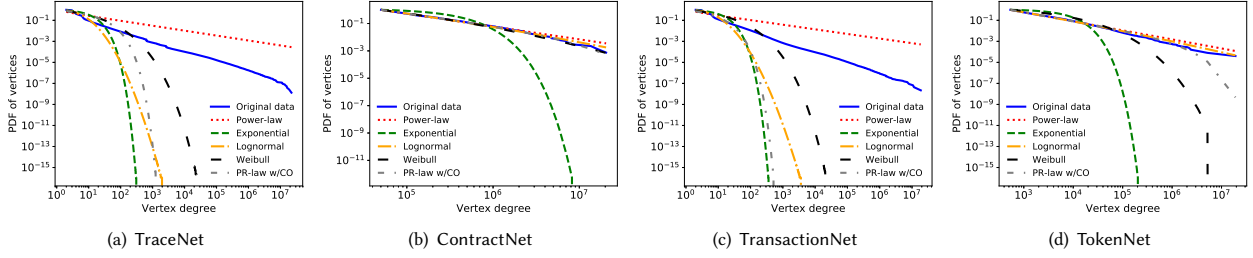


Figure 2: Degree distribution fit with power-law and four alternative distributions, MultiDiGraph versions

Table 7: Likelihood-ratio test results from comparing the best fit for four alternative distributions with the best fit power-law distribution for vertex degrees of the MultiDiGraph versions of our networks. Refer to § 4.1 for technical details of the tests.

	Power-law		Exponential			Log-normal				Weibull				Power-law w/ cut-off			
	$\alpha$	$K_{min}$	$\lambda$	$\mathcal{R}$	$p$ -val	$\mu$	$\alpha$	$\mathcal{R}$	$p$ -val	$a$	$b$	$\mathcal{R}$	$p$ -val	$\alpha$	$\lambda$	$\mathcal{R}$	$p$ -val
TraceNet	1.5	1	$1 \times 10^{-1}$	$7 \times 10^6$	$\approx 0$	1.4	0.7	$-4 \times 10^7$	$\approx 0$	0.7	0.4	$-1 \times 10^7$	$\approx 0$	1.0	$2 \times 10^{-2}$	$-7 \times 10^6$	$\approx 0$
ContractNet	1.9	50 196	$3 \times 10^{-6}$	$8 \times 10^2$	$\approx 0$	-2.9	4.1	$-2 \times 10^0$	0.2	1580.9	0.1	$-8 \times 10^{-1}$	0.8	1.9	$6 \times 10^{-8}$	$-4 \times 10^0$	0.003
TokenNet	1.4	1	$1 \times 10^{-1}$	$5 \times 10^5$	$\approx 0$	1.6	0.8	$-3 \times 10^7$	$\approx 0$	0.5	0.4	$-9 \times 10^6$	$\approx 0$	1.0	$6 \times 10^{-2}$	$-4 \times 10^5$	$\approx 0$
TransactionNet	1.9	535	$2 \times 10^{-4}$	$3 \times 10^4$	$\approx 0$	-30.4	6.7	$-9 \times 10^0$	0.01	7046.6	0.1	$1 \times 10^2$	$\approx 0$	1.9	$4 \times 10^{-7}$	$-2 \times 10^1$	$\approx 0$

#### 4.1 Vertex Degree Distribution

We begin by considering vertex degree distribution. Past researches have shown that many real-world networks are *power-law networks*, including Internet topologies [28], the Web [11, 43], social networks [4], neural networks [16], and power grids [60], that is, the fraction of vertices with degree  $k$  follows a power law, decaying like  $k^{-\alpha}$ , for large  $k$  (i.e.,  $k \geq \hat{K}_{min}$ ) and  $\alpha > 1$ . To test how well the degree distributions are modeled by a power-law, we calculate the best power-law fit using the maximum likelihood method [24]. Moreover, to determine whether a power-law distribution is better than alternative heavy-tailed distributions [19], we compare the fitted power-law to the (i) exponential, (ii) log-normal, (iii) power-law with exponential cutoff, and (iv) stretched exponential or Weibull distributions. All of these have been previously employed to model vertex degree distributions [8, 20, 38]. Each alternative model is also fitted via the maximum likelihood to the empirical degrees  $k \geq \hat{K}_{min}$ , with  $\hat{K}_{min}$  is given by the power-law fit [24].

Following [19], we use difference in log-likelihoods between the power-law and alternative models as test statistic:  $\mathcal{R} = \mathcal{L}_{PL} - \mathcal{L}_{Alt}$ , where  $\mathcal{L}_{PL}$  is the log-likelihood of the power-law model, and  $\mathcal{L}_{Alt}$  is the log-likelihood of an alternative model. The sign of  $\mathcal{R}$  implies which of the two models is favored, power law ( $\mathcal{R} > 0$ ) or alternative ( $\mathcal{R} < 0$ ). Since  $\mathcal{R}$  is derived from data, it is itself a random variable, and thus subject to statistical variations. As a result, the sign of  $\mathcal{R}$  is informative if we can determine that  $|\mathcal{R}|$  is far enough from 0. The standard solution is another hypothesis test, in which we calculate a  $p$ -value against a null model of  $\mathcal{R} = 0$ . If  $p \geq 0.1$ , the sign of  $\mathcal{R}$  is not informative, and the data cannot tell us which model, power-law or alternative, is a better fit. If  $p < 0.1$ , then the data provide a clear conclusion in favor of one model or the other.

We show the degree distribution fitting of MultiDiGraphs in Figure 2, and report the corresponding best parameters in Table 7. For power-law, we find that  $\alpha = 1.4 \sim 1.9$  in all our networks. Moreover, for our larger networks, TraceNet and TransactionNet, three alternative heavy-tailed distributions (i.e., Log-normal, Weibull,

and Power-law with cut-off) are better fit than power-law (due to large, negative  $\mathcal{R}$ , with  $p$ -value  $< 0.1$  in Table 7). Notice that in Figure 2 (a)(c), the number of vertices having degree less than  $10^3$  is quite large in number (higher PDF), for those vertices we find that Log-normal, Weibull, and Power-law with cut-off are better fit.

#### 4.2 Correlation of Indegree and Outdegree

We compare the indegree and outdegree of individual vertices in the four networks, considering the MultiDiGraph versions. Vertices with high indegrees might not have high outdegrees in blockchain graphs, and vice versa. In Figure 3, we show the cumulative distributions (CDF) of the outdegree-to-indegree ratio for vertices.

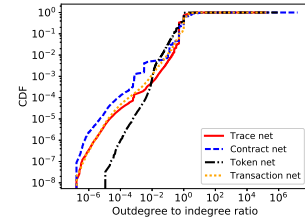


Figure 3: CDF of vertices w/ various outdegree/indegree ratio

We find that most vertices ( $\approx 50\%$ ) have similar outdegrees and indegrees, while a smaller fraction ( $\approx 30\%$ ) of vertices have significantly higher indegrees than outdegrees. Moreover, we also observe that about ( $\approx 20\%$ ) of vertices have significantly higher outdegrees than indegrees. This characteristic is similar to the Web [40], consisting of both hub (having higher outdegrees) and authority (with higher indegrees) vertices, and is very unlikely in social networks. Social networks usually have high correlation between indegrees and outdegrees [52]. In contrast, in blockchain networks, mining pools and mixers generally appear a lot in the from\_address of transactions and traces, as they provide various services and actions to other users. On the other hand, ICO smart contracts appear a lot in the to\_address, as an ICO contract is called frequently to facilitate disbursement of cryptocurrency tokens to investors.

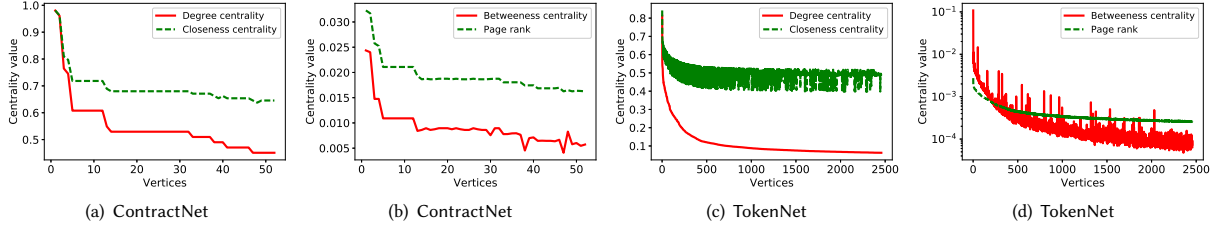


Figure 4: Centrality of vertices from the innermost core of the largest strongly connected component

### 4.3 Centrality Measures

Vertex centrality aims at scoring, ranking, and identification of important vertices according to their power, influence, and relevance [61]. Among numerous centrality definitions, we shall investigate four most critical ones, over simple, undirected graphs.

(i) *Degree centrality*,  $C_D(v)$ , is the degree  $d(v)$  of each vertex  $v$ , normalized by the total number of vertices  $|V|$  in the network. Formally,  $C_D(v) = \frac{d(v)}{|V|}$ . (ii) *Closeness centrality* is the average distance of each vertex to all others in a connected network. Formally,  $C_C(v) = \frac{|V|}{\sum_{u \in V, u \neq v} \text{dist}(v, u)}$ . Here,  $\text{dist}(v, u)$  is the length of the shortest path between  $u$  and  $v$ . (iii) *Betweenness centrality*. Considering that flows move via the shortest distances, the load in a vertex  $v$  is given by the total number of shortest paths passing through  $v$ . Formally,  $C_B(v) = \frac{2}{(|V|-1)(|V|-2)} \sum_{(u, w)} \frac{\eta(u, v, w)}{\eta(u, w)}$ . Here,  $\eta(u, v, w)$  is the number of shortest paths connecting vertices  $u$  and  $w$  that pass through vertex  $v$ , and  $\eta(u, w)$  is the total number of shortest paths between  $u$  and  $w$ . The sum is taken over all pairs  $(u, w)$  of distinct vertices, and the term outside the sum is for normalization in  $(0, 1)$ . (iv) *PageRank*. The basic idea is to transform the network's adjacency matrix  $A$  such that its elements represent the probability transition (via random walk) between a pair of vertices. The network can represent a Markov chain in which each vertex is a state, and the PageRank is calculated by the power method:  $\pi^T = \pi^T G$ , where  $G = \epsilon P + (1 - \epsilon)/|V|$ . Here,  $P$  is the transition probability matrix, given by  $P_{u, v} = A_{u, v} / \sum_v A_{u, v}$ . The original version of the algorithm [17] considers  $\epsilon = 0.85$ .

Since computing betweenness and closeness centralities are expensive over large networks, we estimate the most central vertices from the densest region of the graphs. Specifically, by following [15], we identify the most central vertices from the innermost core of the largest strongly connected component in each network. In Figures 4, we plot centrality values of these vertices for both ContractNet and TokenNet. We observe that relative ranking of these vertices according to different centrality measures remains about the same. The Kendall's tau coefficients between rankings according to (degree, closeness), (degree, betweenness), and (degree, PageRank) over ContractNet are 0.99, 0.85, and 0.93, respectively. Those values for TokenNet are 0.41, 0.77, and 0.95, respectively. In summary, *high-degree vertices in blockchain networks are generally also most central based on betweenness, closeness, and PageRank metrics.*

## 5 GLOBAL NETWORK PROPERTIES: MEASUREMENTS AND ANALYSES

We study the following summary features of the networks.

### 5.1 Reciprocity and Assortativity

In network science, *reciprocity* is a measure of the likelihood of vertices in a directed network to be mutually linked. A classical way to define the reciprocity  $r$  is using the ratio of the number of arcs pointing in both directions to the total number of arcs.

$$r = \frac{(\# \text{ arcs in simple, directed graph} - \# \text{ arcs in simple, undirected graph}) \times 2}{\# \text{ arcs in simple, undirected graph}}$$

With this definition,  $r = 1$  is for a purely bidirectional network, while  $r = 0$  is for a purely unidirectional one. Real networks have intermediate reciprocity values between 0 and 1. Reciprocity has been shown to be crucial to classify and model directed networks, understand the effects of network structure on dynamical processes, explain patterns of growth in out-of-equilibrium networks (in case of the Wikipedia or the World Trade Web), and study the higher-order structures such as correlations and triadic motifs [7, 14, 30, 31, 49, 78–80]. Reciprocity also provides a measure of the simplest feedback process, e.g., the tendency of a vertex to respond to another vertex stimulus in a communication network.

Table 8: Reciprocity, assortativity: simple, directed networks

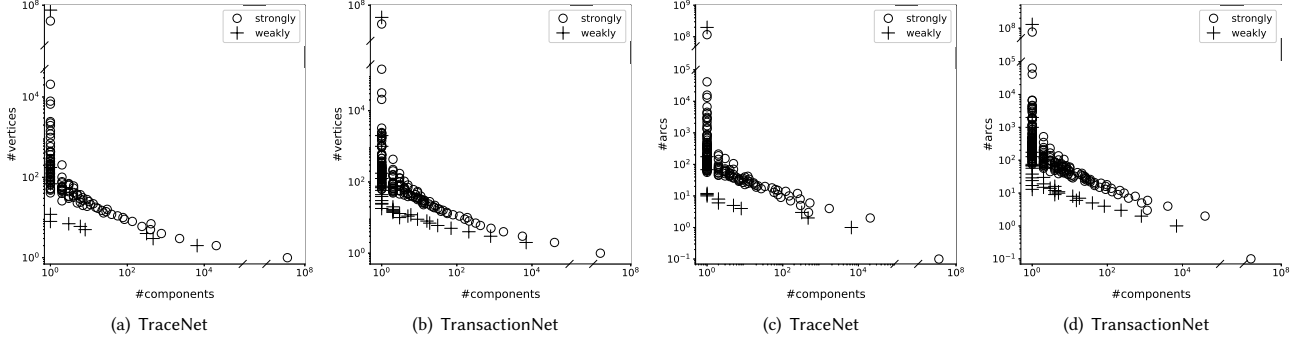
Network (#vertices, #arcs)	Reciprocity	Assortativity
TraceNet (76M, 198M)	0.06	-0.13
ContractNet (11M, 22M)	<b>0.21</b>	<b>-0.64</b>
TransactionNet (46M, 130M)	0.03	-0.12
TokenNet (30M, 95M)	0.03	-0.13

In Table 8, we report reciprocity of the four blockchain networks using their simple, directed versions, that is, multiple arcs from a source to a target vertex (having the same direction) are counted only once. We find that *the reciprocity value of ContractNet is significantly higher than that of the three other networks: about 4 times than that of TraceNet, and 10 times than that of both TransactionNet and TokenNet. This confirms that smart contracts rely more on each other in order to process a transaction or a trace, and thus resulting in a lot of bidirectional arcs.* In fact, smart contracts are similar to functions, and it is normal for these vertices to return something back to the smart contract that has evoked a call on it.

The concept of *assortativity* was introduced by Newman [56], and is extensively studied in network science. Generally, the assortativity of a network is determined for the degree of the vertices in the simple, undirected network. Assortativity,  $\rho$ , lies in the range  $-1 \leq \rho \leq 1$ . A network is said to be *assortative* (i.e.,  $\rho$  tends to 1) when high-degree vertices are, on average, linked to other vertices with high degree, and low-degree vertices are, on average, linked to other vertices with low degree. A network is said to be *disassortative* (i.e.,  $\rho$  tends to -1) when, on average, high-degree vertices are linked to vertices with lower degree, and vice versa.

**Table 9: Connected components characterization of blockchain networks**

Simple, directed networks (#vertices, #arcs)	# Strongly connected components	Largest strongly connected component (#vertices, #arcs)	# Weakly connected components	Largest weakly connected component (#vertices, #arcs)
TraceNet (76M, 198M)	35 215 962	40M, 116M	7 324	76M, 192M
ContractNet (11M, 22M)	<b>9 013 144</b>	2M, 4M	12 555	11M, 20M
TransactionNet (46M, 130M)	15 560 831	30M, 76M	8 181	46M, 128M
TokenNet (30M, 95M)	16 980 001	13M, 56M	54 271	30M, 94M

**Figure 5: Vertices count (a, b) and arcs count (c, d) distribution in connected components of blockchain networks**

In Table 8, we present the assortativity values of our blockchain networks, using their simple, undirected versions. Following [56], we employ the following equation that computes the degree assortativity  $\rho$  of an observed network.

$$\rho = \frac{|E|^{-1} \sum_i j_i k_i - [ |E|^{-1} \sum_i \frac{j_i + k_i}{2} ]^2}{|E|^{-1} \sum_i (j_i^2 + k_i^2) - [ |E|^{-1} \sum_i \frac{j_i + k_i}{2} ]^2}$$

$j_i, k_i$  are the degrees of the vertices at the ends of the  $i$ -th arc, with  $i \in [1, |E|]$ ,  $|E|$  is total number of arcs. *All four networks have assortativity values  $< 0$  (Table 8). Moreover, ContractNet’s assortativity is the most negative, while the other three networks have similar measures. Having negative assortativity values implies that there are relatively more scenarios of addresses with different degrees transacting with each other.* For example, consider those addresses associated with mining pools, exchanges, and mixers. They tend to have higher degrees as they provide various services and actions to other users in the network. *For ContractNet, disassortativity is more prevalent – this means that there exist some generic smart contracts that are used by many other smart contracts.* As an example, “decentralized exchanges” (which are smart contracts) do cryptocurrency exchanges in a decentralized manner, and tend to have many other smart contracts using their services. Thus, the disparity in the degree of such smart contracts and the other smart contracts transacting with it results in higher disassortativity of ContractNet.

Our empirical results confirm previous findings that *non-social networks tend to be disassortative* [56]. Social networks are typically more assortative than non-social networks (i.e., possessing positive vertex degree correlations): Well-connected individuals associate with other well-connected individuals, and poorly-connected individuals associate with each other [58]. *Unlike social networks, as shown earlier, all our blockchain networks are disassortative.*

## 5.2 Strong and Weakly Connected Components

An undirected graph is “connected” if every pair of vertices in the graph is connected. A *connected component* is a maximal connected subgraph of  $G$ . In a directed graph, a component or a subgraph

is called *weakly connected* if replacing all of its directed arcs with undirected arcs produces a connected (undirected) subgraph. A component is *strongly connected* if it contains both-way directed paths between every pair of vertices. The *strongly connected components* are the maximal strongly connected subgraphs in a directed graph. Analogously, the *weakly connected components* are the maximal weakly connected subgraphs in a directed graph. Finding connected components is a well-known problem in network clustering, community detection, compression, and entity resolution.

We characterize connected components from simple, directed version of four networks in Table 9. We observe the following trends – (1) *The counts for the weakly connected components are significantly lesser than respective counts for the strongly connected components. This is primarily because there are less bidirectional arcs between majority of vertex pairs.* (2) *The largest weakly connected component has comparable size to that of the original network. This indicates that the largest weakly connected component spans a significant portion (about 98-99% of vertices and arcs) of the entire network.* (3) *ContractNet’s strongly connected components count is the least among that for all other networks, indicating relatively stronger connectivity within smart contracts, and also the presence of bidirectional arcs between many pairs of smart contracts.*

We next analyze the distribution of counts for vertices and arcs in all our connected components – (1) Figure 5 show that both these distributions follow power-law: *few large components followed by a long-tail of remaining small components.* There is only one weakly connected component having around  $10^7 \sim 10^8$  vertices and arcs, respectively (the largest one), whereas we find about  $10^7$  singleton components, each with one vertex only. (2) We find certain *similarities with the structure of Web and social networks* as follows. Prominent studies [18, 52] have shown that the Web and social networks consist of a single, large strongly connected component (SCC), and other groups of vertices can either reach the SCC or can be reached from the SCC. Analogously, *blockchain graphs also have a single, large SCC, and about 98% of the remaining vertices can either reach this SCC, or can be reached from the SCC.*

### 5.3 Core Decomposition

The  $k$ -core of a (simple, undirected) graph is a maximal subgraph in which every vertex is connected to at least  $k$  other vertices within that subgraph. The set of all  $k$ -cores of a graph, for each  $k$ , forms its *core decomposition* [64]. The *core index* of a vertex  $v$  is the maximal  $k$  for which  $v$  belongs to the  $k$ -core. Core decomposition can be computed by iteratively removing the smallest-degree vertex and setting its core number as its degree at the time of removal. The concept of core decomposition is particularly appealing because (1) it can be computed in linear time [12, 22, 47], and (2) it is related to many of the various definitions of a dense subgraph, and it can be used to speed-up or approximate their computation [9, 21, 26].

Core decomposition for each weakly connected component can be performed independently of each other. In Table 10, we present the results corresponding to the largest weakly connected component of each network. We find that *both ContractNet and TokenNet have larger core indices for vertices in the innermost cores, indicating higher density of their innermost cores*. Furthermore, ContractNet’s innermost core has the maximum size, implying that a lot more vertices participate in denser substructures in this network.

**Table 10: Core decomposition (simple, undirected graphs)**

Largest Weakly Connected Component (#vertices, #arcs)	# Cores	Innermost core (#vertices, #arcs)
TraceNet (76M, 192M)	98	(221, 12 058)
ContractNet (11M, 20M)	<b>264</b>	<b>(1071, 143 352)</b>
TransactionNet (46M, 128M)	105	(682, 55 926)
TokenNet (30M, 94M)	<b>218</b>	(475, 57 124)

**Table 11: # triangles, transitivity (T), global cluster coeff. (C)**

	Largest strongly connected comp. (Simple, undirected)			Largest weakly connected comp. (Simple, undirected)		
	# Triangles	T	C	# Triangles	T	C
TraceNet	4 008 794	$10.0 \times 10^{-7}$	0.099	5 813 165	$1.2 \times 10^{-7}$	0.077
ContractNet	405 265	<b><math>38.0 \times 10^{-7}</math></b>	<b>0.212</b>	871 359	$6.7 \times 10^{-7}$	0.078
TransactionNet	1 908 138	$8.3 \times 10^{-7}$	0.064	4 550 517	<b><math>12.4 \times 10^{-7}</math></b>	0.100
TokenNet	2 803 894	$8.6 \times 10^{-7}$	0.209	5 296 640	$5.5 \times 10^{-7}$	<b>0.175</b>

### 5.4 Triangles, Transitivity, Clustering Coeff.

*Triangle counting* is a community detection algorithm that is used to determine the number of triangles passing through each vertex in a (simple, undirected) network  $G$ . The number of triangles of vertex  $v$  is defined as  $\Delta(v) = |\{\{u, w\} \in E : \{v, u\} \in E \cap \{v, w\} \in E\}|$ . Here,  $E$  denotes the set of arcs in the network  $G$ . A *triple*  $\gamma$  at a vertex  $v$  is a path of length two for which  $v$  is the center vertex. The number of triples of vertex  $v$ , having degree  $d(v)$ , is then defined as  $\gamma(v) = \binom{d(v)}{2}$ . The *local clustering coefficient* of a vertex is the likelihood that its neighbours are also linked. The computation of this score involves triangle counting. The *global clustering coefficient*  $C(G)$  is the normalized sum of those local clustering coefficients. Formally,  $C(G) = \sum_{v \in V} \frac{\Delta(v)}{\gamma(v)}$ . Here,  $V$  denotes the set of vertices in  $G$ . The *transitivity*  $T(G)$  of a network  $G$  is three times the number of triangles divided by the number of triples in the graph, that is,  $T(G) = \frac{\sum_{v \in V} \Delta(v)}{\sum_{v \in V} \gamma(v)}$ . Triangle count, clustering coefficient, and transitivity are used as features for classifying a website as spam/non-spam; to find community structure of a social network [25, 70].

Triangle counting is, however, quite expensive over large-scale networks. A simple approach is to traverse over all vertices and

check for existing arcs between any pair of neighbors. This algorithm, known as *vertex-iterator*, has running time  $O(\sum_{v \in V} \binom{d(v)}{2})$ . In our implementation, we, therefore, consider an *approximation algorithm* [63] that can estimate  $\Delta(v)$  in  $O(1)$  time. Informally speaking, this approximation algorithm samples triples with appropriate probability. It then checks whether an arc between the non-center vertices of the triple is present. We omit more details on this approximation algorithm due to interest of space.







In Table 11, we report the estimated count of triangles, transitivity, and global clustering coefficients for both the largest strongly and weakly connected components of our blockchain networks. In particular, we notice that the *transitivity is quite low*, for example, in the order of  $10^{-7}$ , against an expected value of  $10^{-1} \sim 10^{-2}$ , if connections were made at random ( $2m/n(n-1)$  for a random graph with  $n$  vertices and  $m$  arcs). This suggests that *in blockchain networks, there are forces at work that shy away from the creation of triangles*. Our empirical findings are consistent with past observations: *non-social networks have lower transitivity coefficients* [58]. This is in sharp contrast with social networks, where transitivity is much higher than what we expect by chance. In [57], this phenomenon is attributed to community structure, that is, social networks possess community structure, and other types of networks do not (or they possess it to a lesser degree). We, therefore, suspect that the lack of community structure creates lower transitivity in blockchain networks. Indeed, we find that *high-degree vertices are often “loner-star” [65], that is, connected to mostly low-degree vertices. This results in lack of community structure in blockchain graphs*.

In largest strongly connected component, ContractNet has the highest global clustering coefficient; whereas in largest weakly connected component, TokenNet possesses the highest clustering coefficient. It can be explained by their higher densities (Table 6).

### 5.5 Higher-Order Motifs Counting

Many networks exhibit rich, higher-order connectivity patterns at the level of small subgraphs, also known as *motifs* [13]. To this end, we count the occurrences of various motifs up to having five vertices inside dense substructures of blockchain networks. In particular, we show motif counts in the innermost core obtained from the largest strongly connected components in Table 12. Notice that *the most frequent motifs observed in blockchain graphs are primarily chain and star-shaped. In contrast, the counts for more complex patterns, e.g., cliques and cycles, are quite less*. We further report the density of a motif, which is defined as the ratio of its count to its count in a complete graph having same number of vertices as the innermost core. As expected, *the densities for more complex patterns are less, indicating lack of community structure in blockchain networks*.

**Table 12: Motif counts in ContractNet’s innermost core (simple, undirected). Motif density is the ratio of its count to its count in a complete graph having same number of vertices.**

	#	Motif density		#	Motif density
	13 669	$1 \times 10^{-1}$		2 214	$2 \times 10^{-2}$
	17 081	$3 \times 10^{-3}$		60 297	$9 \times 10^{-3}$
	387 816	$12 \times 10^{-3}$		2 578	$4 \times 10^{-4}$



**Table 13: # Articulation points, Adhesion, Cohesion, Average path lengths, Radius, Diameter (simple, undirected)**

	# Articulation points (% of all vertices)	Largest strongly conn. comp.		Largest weakly conn. comp.		Largest weakly connected component		
		Adhesion	Cohesion	Adhesion	Cohesion	Avg. path length	Radius	Diameter
TraceNet	1 214 137 (1.6%)	1	1	1	1	5.25	5 002	8 267
ContractNet	28 309 (0.2%)	1	1	1	1	5.94	14	27
TransactionNet	1 337 527 (2.9%)	1	1	1	1	5.33	5 002	8 267
TokenNet	75 513 (2.5%)	1	1	1	1	3.87	82	164

## 5.6 Articulation Points, Adhesion, Cohesion

A vertex in an undirected, connected graph is an *articulation point* if and only if removing it disconnects the graph. In a disconnected, undirected graph, removing an articulation point increases the number of connected components. They represent vulnerabilities in a network – single point failures that would split the network into more disconnected components. As shown in Table 13 over simple, undirected versions of our networks, having more vertices does not necessarily imply that the number of articulation points in the network would also be higher. *While TraceNet has approximately 30M more vertices than TransactionNet, it has fewer articulation points than the later. This could be because direct arcs between smart contracts are not included in TransactionNet. ContractNet has significantly less articulation points than the other networks. This indicates that ContractNet is tightly connected as compared to the other three graphs.* Smart contracts may rely on other common smart contracts to process a transaction. The direct arcs between smart contracts in ContractNet result in the least number of articulation points.

*Cohesion and adhesion* refer to the minimum number of vertices and arcs, respectively, that must be removed to disconnect a network. Clearly, they are defined over undirected, connected networks. Hence, instead of finding adhesion and cohesion of the entire blockchain graphs (disconnected), the largest weakly and the largest strongly connected components of each blockchain graph is considered in our analysis. We treat each largest weakly and every largest strongly connected component as a simple, undirected graph, and compute its cohesion and adhesion (Table 13). We observe that *our largest connected components are not at all structurally cohesive. Deletion of only one vertex or, only one arc, disconnects these components.* In case of cohesion, this is the highest-degree vertex in the respective component, such as Binance, a highly popular cryptocurrency exchange that provides a platform for trading.

## 5.7 Path Lengths and Diameter

Finally, we estimate the properties related to shortest paths between pairs of vertices, e.g., *average path lengths, radii, and diameters.* For these measurements, we consider simple, undirected version of our largest strongly and weakly connected components. Note that the *eccentricity* of a vertex  $v$  is the maximal shortest path distance between  $v$  and any other vertex. The radius of a network is defined as the minimum eccentricity across all vertices, and the diameter is the maximum eccentricity across all vertices. Due to high computational complexity associated with evaluating the actual radius and diameter, the numbers presented in Table 13 are from determining the eccentricity of 10 000 random vertices in each component [52]. Interestingly, *analogous to social networks, blockchain graphs are also small-world: The average shortest path length is only 4~6.* However, in both our larger networks, TraceNet and TransactionNet, there are vertices which are far apart, making the radius, diameter as 5 002 and 8 267, respectively.

## 6 STUDYING INDIVIDUAL TOKEN NETS

In our analysis thus far, TokenNet comprises of all tokens ‘hosted’ on Ethereum. However, each of the prominent tokens on Ethereum drives a significantly large groups of miners, exchanges, users, and contracts, indulging in the use, transfer, exchange, and creation of that particular token. In addition to the overall Ethereum network, these token subnetworks are quite interesting by themselves.

We consider three prominent, yet behaviourally distinct, tokens: Bancor (BNT), Binance Coin (BNB), and Zilliqa (ZIL) for independent network analysis. Bancor is a token underlying a decentralized exchange network, featuring a large number of smart contracts; Binance Coin is the utility token for Binance, one of the largest cryptocurrency exchanges in the world; and Zilliqa is a young coin backed by the Zilliqa cryptocurrency network, which rose to prominence within a short span of time. The network for each token is extracted from the original Ethereum TokenNet, filtered by their respective token marker. The basic characteristics of each token network is presented in Table 14. It is interesting to note that even though Bancor has way less number of users (vertices) in the subnetwork compared to Binance Coin, the number of interactions (arcs) is considerably higher. Zilliqa is a new token, but has evidently accumulated significant number of users and interactions.

**Table 14: Characteristics of individual token networks**

Token	# Vertices	# Arcs	# Self Loops	Density
Bancor	65605	1489059	7275	$7 \times 10^{-4}$
Binance Coin	357986	526010	146	$8 \times 10^{-6}$
Zilliqa	91250	334480	111	$8 \times 10^{-5}$

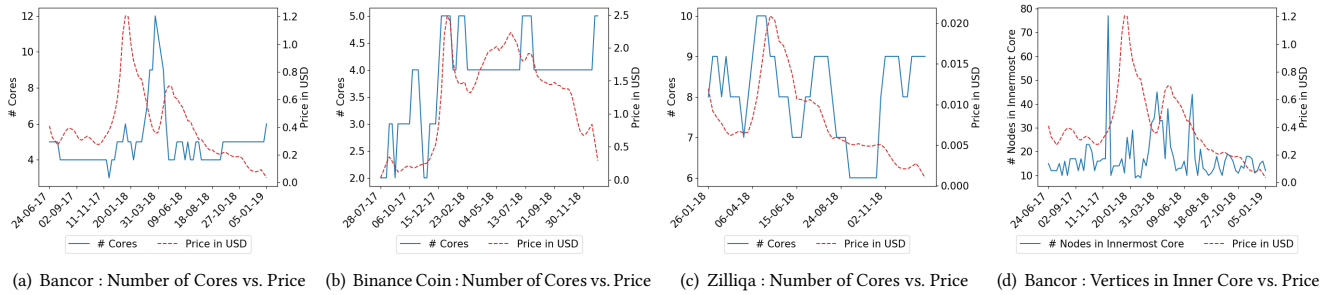
**Table 15: Core decomposition of individual token networks**

Token Network (#vertices, #arcs)	# Cores	Innermost Core (#vertices, #arcs)
Bancor (65605, 1489059)	14	(39, 338)
Binance Coin (357986, 526010)	7	(11, 41)
Zilliqa (91250, 334480)	13	(43, 391)

### 6.1 Core Decomposition

Two major points of interest in the evolution of the individual token subnetworks motivate our analysis: the amount of activity in the token network over time, and the size of the core community driving the token economy over time. To observe these features empirically, we study the  $k$ -core decomposition of each token sub-network, over time, and note – (i) the number of cores, (ii) the number of vertices in the innermost core, (iii) the number of arcs within the innermost core, and (iv) the number of arcs from vertices in the innermost core, in each token network. This temporal progression of *coreness* give us a measure for token activity, hence popularity, over time.

We show the overall  $k$ -core decomposition statistics for each token in Table 15, before proceeding with temporal analysis. Notice that both Bancor and Zilliqa have large core indices and their innermost cores are large, implying that more users participate in dense substructures in these networks, compared with Binance Coin.



**Figure 6: # cores and # vertices in innermost core of the token network vs. the price of the token in USD, over time.**

## 6.2 Temporal Progress of Core Decomposition

In the case of each of the aforementioned three tokens, BNT, BNB and ZIL, we adopt a sliding window technique for the  $k$ -core decomposition, with a window size of 3 weeks, and a slide stride of 1 week. This allows us to extract enough data from the token networks to create a meaningful token subgraph in each case. This results in a 2 week overlap between temporally consecutive token subgraphs, allowing us to see the long-term evolution of the network while minimizing the effect of sudden fluctuations in *coreness*. Still, we observe interesting rise and fall in the number of cores and number of vertices in the innermost core for the individual tokens, exhibiting a significant relation with the token market price in USD.

In the first three subfigures of Figure 6, we present the temporal evolution of the number of cores in the token subgraphs against the corresponding evolution of price of the token in the cryptocurrency market. We notice that for Bancor, the price change seems to drive the activity in the network, for Binance Coin the activity and price move simultaneously, while for Zilliqa, the activity in the network seems to precede the price fluctuations. In the last subfigure of Figure 6, we present the temporal evolution of the number of vertices in the innermost core of the Bancor token subgraph against the corresponding evolution of price of the token (other two cases not depicted due to space constraint). Here, on the other hand, we observe the peaks of activity in the core token community preceding the hike in price of the token. Detailed analysis in this line will be of interest in terms of forecasting the price of Ethereum backed tokens. Certain shifted correlations for the time series are as high as 0.6 for Bancor, 0.63 for Binance, and 0.72 for Zilliqa.

## 7 DISCUSSION AND CONCLUSIONS

In this work we investigated several local and global graph properties over four Ethereum blockchain networks (TraceNet, ContractNet, TransactionNet, and TokenNet), as well as in three prominent token subnetworks (Bancor, Binance Coin, and Zilliqa), and conducted a thorough experimental evaluation.

We find that these blockchain networks are very different from social networks. In case of both TraceNet and TransactionNet, Log-normal, Weibull, and Power-law with cut-off are better fit than the traditional power-law degree distribution. In all four blockchain networks, we have higher outdegree vertices (e.g., mining pools and mixers), as well as higher indegree vertices (e.g., ICO smart contracts). This characteristic is similar to the Web, consisting of both hub (having higher outdegrees) and authority (with higher

indegrees) vertices, and is unlikely in social networks, which usually have high correlation between indegrees and outdegrees. As a result, blockchain networks are disassortative, having very low transitivity. Moreover, most frequent motifs observed in blockchain graphs are chain and star-shaped. Complex patterns, such as triangles, cycles, and cliques occur less, indicating lack of community structure in blockchain networks. Removal of only the highest-degree vertex (e.g., Binance, a global cryptocurrency exchange) can disconnect the entire largest weakly connected components in these graphs.

In spite of the aforementioned differences, blockchain networks are surprisingly small-world and well-connected. Analogous to social networks, blockchain graphs have average shortest path lengths only 4~6. Similar to both social networks and the Web, blockchain networks contain a single, large strongly connected component (SCC), and about 98% of the remaining vertices can either reach this SCC, or can be reached from the SCC. In terms of the four different networks, we observe that ContractNet has more self-loops and bidirectional arcs (hence, higher reciprocity), while TokenNet has fewer of them. As a result, the MultiDigraph of ContractNet is denser, while the simple, undirected version of TokenNet is more dense. Both of them yield larger core indices for vertices in the innermost cores, indicating higher density of their innermost cores. Moreover, both ContractNet and TokenNet have smaller radius and diameter compared to our larger networks, TraceNet and TransactionNet.

**Future work.** Following our characterization of Ethereum into four different blockchain networks, there is ample opportunity for future work. Study of individual mining pools as complex self-contained evolving networks would be interesting, as would be an investigation on the interplay between mining pools to identify instances of selfish mining and mining strategies [35, 59]. Further analysis of the individual Token networks in terms of activity signatures and temporal evolution (like change in coreness) may lead to more accurate forecasting of trading behavior and token prices in the cryptocurrency market [67, 68, 71]. Identification of influential vertices and complex motifs (like cliques and cycles) in the blockchain networks may also lead to detection of fraudulent activities in the transaction and token networks of Ethereum. To facilitate such research directions, we open source our datasets: <https://github.com/sgsourav/blockchain-network-analysis>. Quite naturally, a similar line of measurements and analyses can be applied to other public blockchain platforms to unearth interesting phenomena within and across the Web of blockchain networks.

## REFERENCES

- [1] 2013. Elliptic. <https://www.elliptic.co/>.
- [2] 2018. Ethereum in BigQuery: a Public Dataset for Smart Contract Analytics. <https://cloud.google.com/blog/products/data-analytics/ethereum-bigquery-public-dataset-smart-contract-analytics>. Accessed: 2010-10-12.
- [3] 2019. Ethereum Tokens. <https://medium.com/linum-labs/ethereum-tokens-explained-ffe9df918008>. Accessed: 2010-10-12.
- [4] L. Adamic, O. Buyukkocuten, and E. Adar. 2003. A Social Network Caught in the Web. *First Monday* 8, 6 (2003).
- [5] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. 2007. Analysis of Topological Characteristics of Huge Online Social Networking Services. In *WWW*.
- [6] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu. 2017. Blockchain: A Graph Primer. *CoRR abs/1708.08749* (2017). arXiv:1708.08749 <http://arxiv.org/abs/1708.08749>
- [7] L. Akoglu, P. O. S. Vaz de Melo, and C. Faloutsos. 2012. Quantifying Reciprocity in Large Weighted Communication Networks. In *PAKDD*.
- [8] L. A. N. Amaral, A. Scala, M. Barthélemy, and H. E. Stanley. 2000. Classes of Small-world Networks. *Proc. Natl. Acad. Sci.* 97, 21 (2000), 11149–11152.
- [9] R. Andersen and K. Chellapilla. 2009. Finding Dense Subgraphs with Size Bounds. In *Algorithms and Models for the Web-Graph*.
- [10] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. 2006. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In *KDD*.
- [11] A.-L. Barabasi and R. Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512.
- [12] V. Batagelj and M. Zaveršnik. 2011. Fast Algorithms for Determining (Generalized) Core Groups in Social Networks. *Adv. Data Anal. Classif.* 5, 2 (2011), 129–145.
- [13] A. R. Benson, D. F. Gleich, and J. Leskovec. 2016. Higher-order Organization of Complex Networks. *Science* 353, 6295 (2016), 163–166.
- [14] M. Boguna and M. A. Serrano. 2005. Generalized Percolation in Random Directed Networks. *Phys. Rev. E* 72 (2005), 016106. Issue 1.
- [15] F. Bonchi, A. Khan, and L. Severini. 2019. Distance-generalized Core Decomposition. In *SIGMOD*.
- [16] V. Braitenberg and A. Schuz. 1991. *Anatomy of the Cortex: Statistics and Geometry*. Springer-Verlag, Berlin.
- [17] S. Brin and L. Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. In *WWW*.
- [18] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. 2000. Graph Structure in the Web. In *WWW*.
- [19] A. D. Broido and A. Clauset. 2019. Scale-free Networks Are Rare. *Nature Communications* 10, 2017 (2019).
- [20] G. Buzsáki and K. Mizuseki. 2014. The Log-dynamic Brain: How Skewed Distributions Affect Network Operations. *Nature Rev. Neurosci.* 15, 4 (2014), 264–278.
- [21] M. Charikar. 2000. Greedy Approximation Algorithms for Finding Dense Components in a Graph. In *Approximation Algorithms for Combinatorial Optimization*.
- [22] J. Cheng, Y. Ke, S. Chu, and M. T. Oszu. 2011. Efficient Core Decomposition in Massive Networks. In *ICDE*.
- [23] N. Christin. 2013. Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. In *WWW*.
- [24] A. Clauset, C. R. Shalizi, and M. E. J. Newman. 2009. Power-Law Distributions in Empirical Data. *SIAM Rev.* 51, 4 (2009), 661–703.
- [25] J.-P. Eckmann and E. Moses. 2002. Curvature of Co-links Uncovers Hidden Thematic Layers in the World Wide Web. *Proceedings of the National Academy of Sciences* 99, 9 (2002), 5825–5829.
- [26] D. Eppstein, M. Löffler, and D. Strash. 2010. Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time. In *Algorithms and Computation*.
- [27] L. Ermann, K. M. Frahm, and D. L. Shepelyansky. 2018. Google Matrix of Bitcoin Network. *The European Physical Journal B* (2018), 91–127.
- [28] M. Faloutsos, P. Faloutsos, and C. Faloutsos. 1999. On Power-law Relationships of the Internet Topology. *SIGCOMM Comput. Commun. Rev.* 29, 4 (1999), 251–262.
- [29] S. Ferretti and G. D'Angelo. 2019. On the Ethereum Blockchain Structure: A Complex Networks Theory Perspective. *Concurrency and Computation: Practice and Experience* (2019), e5493.
- [30] D. Garlaschelli and M. I. Loffredo. 2005. Structure and Evolution of the World Trade Network. *Physica A: Statistical Mechanics and its Applications* 355, 1 (2005), 138–144.
- [31] D. Garlaschelli, F. Ruzzenenti, and R. Basosi. 2010. Complex Networks and Symmetry I: A Review. *Symmetry* 2 (2010), 1683–1709. Issue 3.
- [32] A. Gionis and C. E. Tsourakakis. 2015. Dense Subgraph Discovery. In *KDD*.
- [33] M. S. Granovetter. 1973. The Strength of Weak Ties. *The American Journal of Sociology* 78, 6 (1973), 1360–1380.
- [34] A. Greaves and B. Au. 2015. *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. Technical Report. Stanford.
- [35] C. Grunspan and R. Pérez-Marco. 2019. Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks. *CoRR abs/1904.07675* (2019). arXiv:1904.07675 <http://arxiv.org/abs/1904.07675>
- [36] A. A. Hagberg, D. A. Schult, and P. J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Python in Science Conf*.
- [37] B. Haslhofer, R. Karl, and E. Filtz. 2016. O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs. In *SEMANTICS*.
- [38] H. Jeong, S. P. Mason, A.-L. Barabasi, and Z. N. Oltvai. 2001. Lethality and Centrality in Protein Networks. *Nature* 411, 41 (2001), 41–42.
- [39] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn. 2018. An Empirical Analysis of Anonymity in Zcash. In *USENIX Security Symposium*.
- [40] J. Kleinberg and S. Lawrence. 2001. The Structure of the Web. *Science* 294, 5548 (2001), 1849–1850.
- [41] D. Kondor, I. Csabai, J. SzÁijle, M. PÁssfai, and G. Vattay. 2014. Inferring the Interplay between Network Structure and Market Effects in Bitcoin. *New Journal of Physics* 16 (2014).
- [42] R. Kumar, J. Novak, and A. Tomkins. 2006. Structure and Evolution of Online Social Networks. In *KDD*.
- [43] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. 1999. Trawling the Web for Emerging Cyber-communities. In *WWW*.
- [44] X.T. Lee, A. Khan, S. Sen Gupta, Y.H. Ong, and X. Liu. 2020. Measurements, Analyses, and Insights on the Entire Ethereum Blockchain Network (Dataset). <https://github.com/sgsourav/blockchain-network-analysis>.
- [45] J. Leskovec and E. Horvitz. 2008. Planetary-scale Views on a Large Instant-messaging Network. In *WWW*.
- [46] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Aberg. 2001. The Web of Human Sexual Contacts. *Nature* 411 (2001), 907–908.
- [47] D. W. Matula and L. L. Beck. 1983. Smallest-last Ordering and Clustering and Graph Coloring Algorithms. *J. ACM* 30, 3 (1983), 417–427.
- [48] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. 2016. A Fistful of Bitcoins: Characterizing Payments among Men with no Names. *Commun. ACM* 59, 4 (2016), 86–93.
- [49] L. A. Meyers, M. E. J. Newman, and B. Pourbohloul. 2006. Predicting Epidemics on Directed Contact Networks. *Journal of Theoretical Biology* 240, 3 (2006), 400–418.
- [50] S. Milgram. 1967. The Small-World Problem. *Psychology Today* 1 (1967).
- [51] A. Mislove, H. S. Koppula, K. P. Gummadri, P. Druschel, and B. Bhattacharjee. 2008. Growth of the Flickr Social Network. In *WOSN*.
- [52] A. Mislove, M. Marcon, K. P. Gummadri, P. Druschel, and B. Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *SIGCOMM IMC*.
- [53] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, and N. Christin. 2018. An Empirical Analysis of Traceability in the Monero Blockchain. *PoPETs* 2018, 3 (2018), 143–163.
- [54] S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>.
- [55] M. E. J. Newman. 2001. The Structure of Scientific Collaboration Networks. *Proceedings of the National Academy of Sciences* 98, 2 (2001), 404–409.
- [56] M. E. J. Newman. 2002. Assortative Mixing in Networks. *Physical Review Letters* 89, 20 (2002), 208701.
- [57] M. E. J. Newman. 2003. Properties of Highly Clustered Networks. *Phys. Rev. E* 68 (2003), 026121. Issue 2.
- [58] M. E. J. Newman and J. Park. 2003. Why Social Networks are Different from Other Types of Networks. *Physical review E, Statistical, nonlinear, and soft matter physics* 68 3 Pt 2 (2003), 036122.
- [59] J. Niu and C. Feng. 2019. Selfish Mining in Ethereum. *CoRR abs/1901.04620* (2019). arXiv:1901.04620 <http://arxiv.org/abs/1901.04620>
- [60] A. G. Phadke and J. S. Thorp. 1988. *Computer Relaying for Power Systems*. John Wiley & Sons, Inc., New York, NY, USA.
- [61] F. A. Rodrigues. 2019. Network Centrality: An Introduction.
- [62] D. Ron and A. Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*.
- [63] T. Schank and D. Wagner. 2005. Approximating Clustering Coefficient and Transitivity. *J. Graph Algorithms Appl.* 9, 2 (2005), 265–275.
- [64] S. B. Seidman. 1983. Network Structure and Minimum Degree. *Social Networks* 5, 3 (1983), 269–287.
- [65] K. Shin, T. Eliassi-Rad, and C. Faloutsos. 2018. Patterns and Anomalies in K-cores of Real-world Graphs with Applications. *Knowl. Inf. Syst.* 54, 3 (2018), 677–710.
- [66] G. Siganos, S. L. Tauro, and M. Faloutsos. 2006. Jellyfish: A Conceptual Model for the AS Internet Topology. *Journal of Communications and Networks* 8, 3 (2006), 339–350.
- [67] S. Somin, G. Gordon, and Y. Altshuler. 2018. Network Analysis of ERC20 Tokens Trading on Ethereum Blockchain, In Complex Systems. *Springer Proceedings in Complexity*, 439–450.
- [68] S. Somin, G. Gordon, and Y. Altshuler. 2018. Social Signals in the Ethereum Trading Network. *CoRR abs/1805.12097* (2018). arXiv:1805.12097 <http://arxiv.org/abs/1805.12097>
- [69] M. Spagnuolo, F. Maggi, and S. Zanero. 2014. Bitlodine: Extracting Intelligence from the Bitcoin Network. In *Financial Cryptography and Data Security*.
- [70] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. 2011. The Anatomy of the Facebook Social Graph. *CoRR abs/1111.4503* (2011).
- [71] F. Victor and B. K. Lüders. 2019. Measuring Ethereum-based ERC20 Token Networks. In *Financial Cryptography and Data Security*.

- [72] F. Vogelsteller, V. Buterin, et al. [n.d.]. Ethereum Whitepaper. <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [73] S. Wasserman and K. Faust. 1995. *Social Network Analysis: Methods and Applications*. University Press.
- [74] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. 2009. User Interactions in Social Networks and Their Implications. In *EuroSys*.
- [75] G. Wood. [n.d.]. Ethereum: A Secure Decentralised Generalised Transaction Ledger. <https://github.com/ethereum/yellowpaper>.
- [76] S. Y. Yang and J. Kim. 2015. Bitcoin Market Return and Volatility Forecasting Using Transaction Network Flow Properties. In *SSCI*.
- [77] H. Yousaf, G. Kappos, and S. Meiklejohn. 2019. Tracing Transactions Across Cryptocurrency Ledgers. In *USENIX Security Symposium*.
- [78] G. Zamora-Lopez, V. Zlatic, C. Zhou, H. Stefancic, and J. Kurths. 2008. Reciprocity of Networks with Degree Correlations and Arbitrary Degree Sequences. *Phys. Rev. E* 77 (2008), 016106. Issue 1.
- [79] V. Zlatic and H. Stefancic. 2009. Influence of Reciprocal Edges on Degree Distribution and Degree Correlations. *Phys. Rev. E* 80 (2009), 016117. Issue 1.
- [80] V. Zlatic and H. Stefancic. 2011. Model of Wikipedia Growth Based on Information Exchange via Reciprocal Arcs. *EPL (Europhysics Letters)* 93, 5 (2011), 58005.