Stick a fork in it: Analyzing the Ethereum network partition

Lucianna Kiffer
Northeastern University
lkiffer@ccs.neu.edu

Dave Levin
University of Maryland
dml@cs.umd.edu

Alan Mislove
Northeastern University
amislove@ccs.neu.edu

ABSTRACT

As blockchain technologies and cryptocurrencies increase in popularity, their decentralization poses unique challenges in network partitions. In traditional distributed systems, network partitions are generally a result of bugs or connectivity failures; the typical goal of the system designer is to automatically recover from such issues as seamlessly as possible. Blockchain-based systems, however, rely on purposeful "forks" to roll out protocol changes in a decentralized manner. Not all users may agree with proposed changes, and thus forks can persist, leading to permanent network partitions.

In this paper, we closely study the large-scale fork that occurred in Ethereum, a new blockchain technology that allows for both currency transactions and smart contracts. Ethereum is currently the second-most-valuable cryptocurrency, with a market capitalization of over $28B. We explore the consequences of this fork, showing the impact on the two networks and their mining pools, and how the fork lead to unintentional incentives and security vulnerabilities.

1 INTRODUCTION

The use of blockchains—for example, in systems like Bitcoin and Ethereum—is an increasingly popular approach to building distributed systems. While currently most well-know as the basis for cryptocurrencies, blockchains have also been proposed to serve as the basis for domain registries [1], medical records [2], and asset registries [3], just to name a few. Blockchains are, at their core, a mechanism for storing state across a network of machines without any centralized trust. Combined with proof-of-work-based limits to generating blocks (called *mining*), blockchains can provide a tamper-proof way to store information.

Unfortunately, the inherent decentralization of blockchain systems poses a challenge in how network partitions (i.e., the inability of member nodes to communicate[1]) are handled. In traditional distributed systems, network partitions are generally a result of connectivity failures or software misconfigurations—i.e., are unintentional—and systems are typically designed to recover from such instances in as automatic a fashion as possible. Blockchain-based systems, however, rely on purposeful "hard forks" to roll out protocol changes in a decentralized manner. If not all users agree with proposed changes, or not all users upgrade to the new software, a persistent network partition can occur between the nodes that support the newer version of the protocol and those that do not. In this case, there will effectively be two separate systems running in parallel, each with its own state and history. In fact, such a persistent fork occurred [4] in Bitcoin in the summer of 2017, meaning each user who existed before the fork now has two "copies" of their wallet (one in the original Bitcoin, another in the new "Bitcoin Cash") that they can spend independently.[2]

Ethereum—a blockchain-based system that currently enjoys the second-highest market cap of all cryptocurrencies [16]—presents an interesting case study for examining how blockchain-based systems react to hard forks. Ethereum runs a blockchain of transactions similar to Bitcoin, but it does so to implement a *distributed virtual machine* that users can execute code on with a custom scripting language. Ethereum is of particular interest when it comes to forks because in July 2016 the system forked into two partitions, both of which still exist today; these two partitions represent two versions of the same currency and are dubbed *Ethereum* (ETH) and *Ethereum Classic* (ETC).

In this paper, we closely examine the dynamics around the ETH/ETC fork to better understand the behavior of the participants, and the impact a fork has on the security and market value of the overall system. Overall, we make the following observations: (1) Forks can lead to drastic, rapid partitions: ETC experienced a sudden loss of roughly 90% of the nodes in its network immediately after the fork. (2) Stabilization after forks can take days to occur: It took two days for ETC to resume producing blocks at the target rate; an influx of nodes re-joined ETC over the subsequent two weeks.

HotNets-XVI, November 30–December 1, 2017, Palo Alto, CA, USA

ACM ISBN 978-1-4503-5569-8/17/11…$15.00
https://doi.org/10.1145/3152434.3152449

[1] Here, a network partition means nodes can no longer communicate due to a portion of the nodes adopting a new protocol; this is in contrast to a "traditional" network partition where nodes are unable to send messages to each other at all.

[2] In March 2013, Bitcoin also experienced a short-lived fork due to inconsistent protocol versions that lasted approximately four hours before developers intervened [5].

(3) Forks can persist, with divergent behavior afterwards: ETC's mining power has held constant since the fork, while ETH's has increased tremendously. (4) The "market" between the two networks appears to be operating efficiently, with the expected return (in USD) on mining being almost identical between the two. (5) The fork unintentionally introduced a security vulnerability wherein attackers can rebroadcast transactions into the other network; this continues to this day, and we quantify this behavior. (6) Pool-based mining behavior in ETC has slowly reached a distribution similar to that in ETH.

As the 2017 Bitcoin fork [4] suggests, such network partitions are likely to be a fact of life with digital currencies based on public ledgers. Unfortunately, neither Ethereum nor Bitcoin were designed to operate under a persistent fork. We believe our study to be an important first step towards better anticipating and handling the unintended consequences of persistent forks.

2 BACKGROUND

Ethereum is one of the newest blockchain-based systems. The goal of such systems is typically to store state in a distributed and tamper-proof fashion. Blockchain-based systems were first introduced in the 2008 Bitcoin white paper by the pseudonym Satoshi Nakamoto [25]. At its core, a blockchain is a series of *blocks*, each containing a record of transactions (essentially, state changes) and a reference to the previous block. Generating a new block is secured by cryptopuzzles, where each peer must expend effort to "win" the right to generate a block. To ensure distributed agreement in the case where multiple blockchains exist, participants in the network choose to believe the chain that represents the most work (typically by picking the longest chain).

2.1 Ethereum and forks

Ethereum is different from previous blockchain systems. Like Bitcoin, Ethereum is a network of peers (called *miners*) who collectively store the entire blockchain containing every transaction that has ever taken place in the network. In Ethereum, the "currency" is called *ether*, and each block mined earns the winner 5 ether (Ethereum aims to produce a new block every 14 seconds). Unlike Bitcoin, Ethereum allows users to upload and run code, called *contracts*, using a Turing-complete language. Each operation the code executes, and each byte of memory the code uses, costs "gas", which is essentially a small fraction of an ether. These contracts can be arbitrarily complicated; they have their own ether balance, and can even transfer ether and call other contracts. For example, a user could implement a contract that would transfer ether to another user if some condition were met; the second user could then be assured that the first could not back out of the transaction. All of these interactions (the contract code, the ether balances, and the contract memory) are stored in Ethereum's blockchain.

Transient forks Due to the distributed nature of blockchain systems, two miners will occasionally mine a block before they are aware of the fact that the other did so as well. In this case, some of the miners will have different views of what the "real" new block is. However, this situation will ultimately be resolved: eventually, one of the two chains will become longer as future blocks are mined. Participants will switch over to the longer chain and abandon the shorter one, effectively removing the fork. This type of fork is termed a *transient fork* and is not of interest to our study, as the protocol has built-in mechanisms to address it.

Hard forks Systems that use blockchains are often under active development and occasionally need to update the software that nodes run. If these changes are backwards-compatible, they are often termed *soft forks*; in the cases where the chain is not backwards compatible (e.g., if the network messages or blockchain data change), they are often termed *hard forks*. These changes are often rolled out as new software versions. In the case of hard forks, it is typical for the developers to release the updated software but announce a specific block number (or time) at which the change will actually be activated; doing so provides users with sufficient time to upgrade their software before the change happens. Typically, hard forks are adopted by virtually the entire community; recent events have shown, however, that some forks are not universally adopted.

Ethereum, Ethereum Classic, and the DAO In this paper, we look closely at the Ethereum (ETH) and Ethereum Classic (ETC) fork, which was caused by a hard fork in July 2016 and has persisted to this day. In order to understand this fork, one needs to understand the DAO (Distributed Autonomous Organization).

In brief, in April 2016, a collection of smart contracts making up the DAO was created to operate as a decentralized crowdfunding platform for Ethereum projects. Any user could send ether to the DAO in exchange for voting power over which projects to fund. However, in June 2016 a malicious user exploited a vulnerability in one of the DAO's contracts and began to funnel money out of the DAO; the attacker obtained roughly $50M worth of ether. The vulnerability was in one of the DAO contracts, and not a bug in Ethereum itself; as a result, from Ethereum's perspective, the contract calls were all perfectly valid. Regardless, the attack resulted in many users potentially losing a significant investment, including many Ethereum developers.

To resolve this loss, a hard-fork was proposed which would edit Ethereum's code to effectively *erase* the attacker's transactions from the blockchain. There was a large debate among the Ethereum community of whether this fork should take place; one side argued that the attack was completely legal under the blockchain system where "code is law", and the other side argued that the attack put the faith in Ethereum in jeopardy. Ultimately, the hard fork went into effect on July 20, 2016. The network that accepted the hard fork kept the

name Ethereum (ETH), and the network that rejected the hard fork (and kept the attacker's transactions) was called Ethereum Classic (ETC).

Other Ethereum forks Although the DAO fork was the most controversial, it is not the only fork Ethereum has experienced. For instance, ETH had a hard fork on November 22, 2016 to increase the cost of a particular contract call, thereby addressing a potential denial-of-service attack [10–12]. Additionally, ETC forked on January 13, 2017 to incorporate similar defenses and to add replay protection [11]. ETC's fork lasted much longer than ETH's—3,583 blocks versus 86—likely due to ETC's smaller network size, so any subgroup working on a fork was more noticeable. In both cases, the forks were eventually resolved by the branch supporting the protocol changes winning out and the other dying off. The original DAO fork, on the other hand, persists to this day.

2.2 Related work

While existing measurement studies have focused on the Ethereum network (and other cryptocurrency networks) under normal operation or under simulated adversarial attacks, none to our knowledge have looked at how Ethereum responded to the large-scale fork. For example, Anderson et al. [14] analyzed the current (June 2016) use of three cryptocurrencies including Ethereum, looking at statistics of the kinds of transactions that were taking place in each. Others have used network statistics to evaluate the constraints of scaling decentralized blockchain protocols [17].

There has been significant complementary work studying the Bitcoin network, including studies of how information propagates in the Bitcoin network [18], properties of repeated subgraphs [26], and overall transaction patterns [23]. There has also been significant work looking at privacy and anonymity in cryptocurrencies, including examinations of how transaction histories can be used to de-anonymize addresses [15], studies of mixing services meant to anonymize addresses [24], and studies of how network traffic can deanonymize Bitcoin addresses [20].

The potential for network partitions is not unique to blockchains, as traditional distributed systems also have state they wish to keep consistent, and aim to largely handle inconsistencies in as automated a way as possible. There is a long line of work focusing on detecting and preventing inconsistencies [27] as well as building fault-tolerant systems [13, 21]. The distinction between blockchain systems and traditional distributed systems is that the shared state is designed to be tamper proof, even if a significant fraction of the network is malicious. Ethereum does use Kademlia's peer-to-peer protocol [22] to find peers to communicate with, but this is not a part of the blockchain consensus protocol.

3 ANALYSIS

We now present our examination of the Ethereum fork. We first look closely at the activity immediately surrounding the fork itself—looking at the behavior of miners and the protocol overall—before looking at the long-term impacts and dynamics of ETH and ETC.

3.1 Datasets

To collect data, we ran full Ethereum nodes in both the ETH and ETC networks.[3] As part of each node's participation in the network, it downloaded and verified all blocks in the blockchain. We then exported all block and transaction information from the nodes and processed it in a separate database. In order to collect ETH and ETC exchange rates, we relied on data from `coinmarketcap.com` (and verified their reported exchange rates against other marketplaces including Coinbase, Coingecko, and Coindesk).

We note that all of the data we have collected is publicly available; after all, Ethereum implements a public ledger. However, our study remains novel as it entails a direct, empirical comparison between the dynamics of the two resulting networks.

3.2 Short-term fork dynamics

We begin by examining the time that immediately surrounded the July 2016 hard fork. In Figure 1, we present the number of blocks each network mined per hour, the average difficulty per block, and the time delta between blocks in seconds. We can immediately make two key observations.

First, we can see that the fork itself caused the ETC network to suddenly experience a massive drop in nodes. This can be observed in the number of blocks per hour, which falls close to 0 for almost a day. The underlying reason for this is that block generation is limited by the *difficulty* parameter, which is calculated based on the difficulty of the previous block: if the time between blocks is below the target of 14 seconds, the difficulty is raised; if the time between blocks is above 14 seconds, the difficulty is lowered, but there is a cap in the absolute difference in difficulty between two blocks. In the case of the fork, because the number of nodes in ETC fell instantaneously and dramatically, the difficulty calculation responded more slowly, as can be seen in the middle graph. Looking at the number of blocks generated per hour, it took almost two days before the difficulty calculation was able to fully adjust to the new network size; in the meantime, the average time delta per block spiked to over 1,200 seconds (almost two orders of magnitude higher than the target).

Second, looking beyond the immediate fork (and the two days it took for the difficulty to stabilize), we see an interesting interplay between the difficulty in the two networks. In particular, over the two weeks following the fork, we can see a decrease in the difficulty in ETH that is mirrored by an increase in the difficulty of ETC. This suggests that miners who originally "took" the fork and switched to ETH actually switched *back* to ETC. We are unable to verify this

[3] One can also download the blockchain from other sources (e.g., "block explorer" websites [6]) that participate in the network.

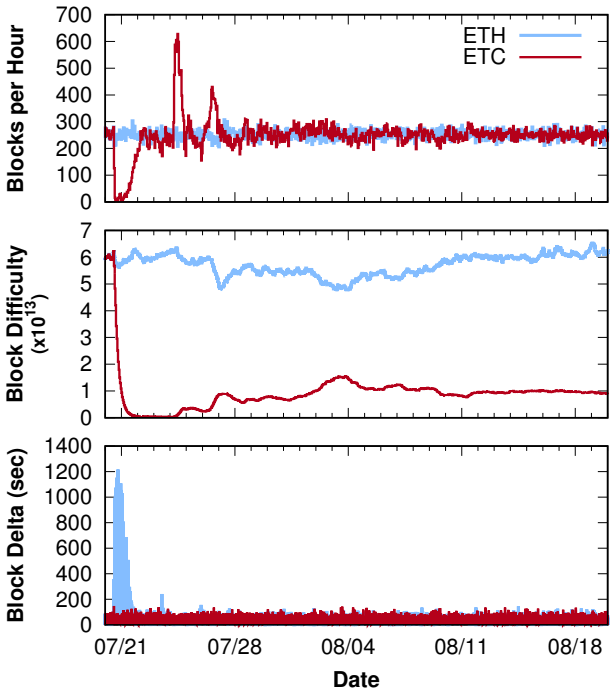

Figure 1: Blocks per hour (top), block difficulty (middle), and time delta between blocks (bottom) the month following the hard fork. Significant dynamics occurred immediately and two weeks after the fork.

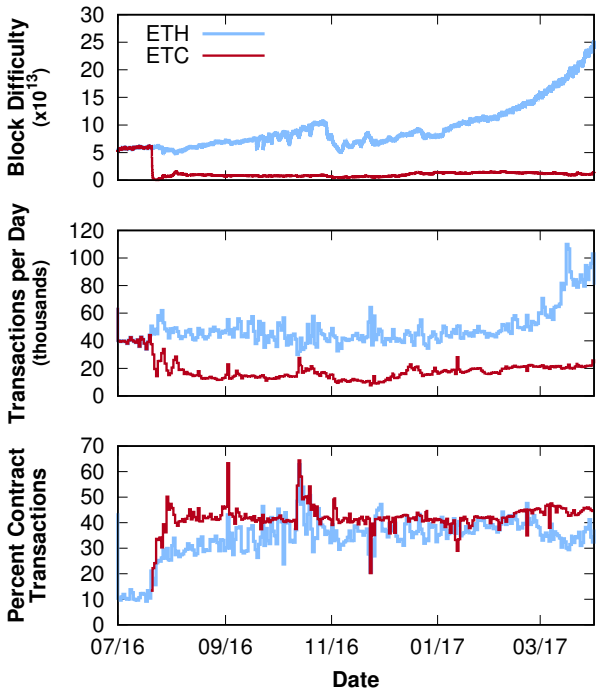

Figure 2: The overall difficulty per block (top), the number of transactions per day (middle), and fraction of transactions involving contracts (bottom) in the nine months since the fork.

hypothesis—the blockchain itself does not contain the identity of the miner, as most miners are members of mining pools—but the almost mirror-image behavior of the difficulty calculation strongly suggests that some miners may have changed their minds about which network(s) to join.

These observations together highlight that the network may be vulnerable in the time period immediately following the fork: an attacker may have been able to use the unexpected short-term dynamics of forks (e.g., the fact that many network parameters such as difficulty and neighbor lists are in flux) to interfere with the operation of the network.

3.3 Long-term fork dynamics

Next, we turn to examine the long-term impacts of the fork and the interactions between the two networks. We first look at how the two networks are used before examining whether any arbitrage opportunities exist and whether the fork introduced any unintended security vulnerabilities.

Network participant behavior We begin by examining the behavior of the two different networks. Figure 2 presents the block difficulty (top), total number of transactions per day (middle), and the percentage of transactions that are contract calls (bottom) for both networks since the fork.

We can first observe that the ETH network has substantially higher difficulty (roughly an order of magnitude), indicating that the vast majority of the mining power is focused on ETH. However, this skew is not seen to the same degree if we examine the number of *transactions* per day; the number of ETH transactions to ETC transactions was roughly 2.5:1 for most of the measurement study but increased to up to 5:1 in late March 2017. Moreover, we can see in the bottom graph that the fraction of transactions that were contract calls—rather than simple currency exchanges—was similar in the two networks until very recently. Overall, this discrepancy suggests that there is a difference in the way the two networks are being used, despite the fact that they are simply two variants of the same system; exploring these differences is an interesting topic for future work. The ETH network has significantly more mining, but fewer transactions per miner. We hypothesize that the press coverage that ETH received in March (e.g., due to its backing by large organizations [19]) led to an influx of speculation, which could explain the rise of transactions in ETH at that time.

Network efficiency Next, we examine whether miners are rational in their choice of which network to participate in. Recall that the hash function for Ethereum is actually different than the ones used in most other cryptocurrencies (e.g., Bitcoin). Thus, there is unlikely to be a significant amount of dedicated hardware-based mining capacity (e.g., ASICs).

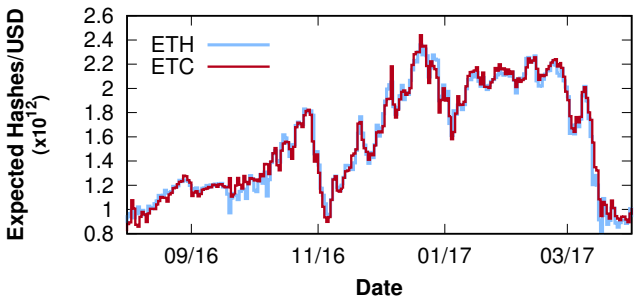

Figure 3: The expected "payoff" for mining in ETH and ETC, as calculated by the expected number of hashes a miner would need to calculate to earn 1 USD. We observe a strong correlation.

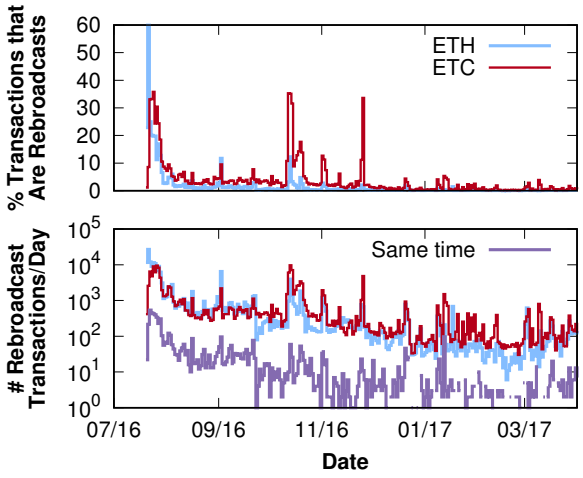

Figure 4: The number of rebroadcast transactions ("echos") in ETH and ETC (bottom), and the percentage of all transactions that these rebroadcasts represent (top). We see a high level of rebroadcasting initially after the fork, and it persists even to today. Most of the rebroadcasts were originally broadcast in ETH and then rebroadcast into ETC.

Thus, after the fork, potential miners had the choice of mining in ETH or ETC[4], and the rational choice of which to participate in is based on both the probability of winning in each (i.e., the inverse of the difficulty) and the exchange rate to traditional currencies.

To explore this, we calculate number of hashes that a miner would be expected to have to compute per USD, for both ETH and ETC. To do so, we divided the average number hashes to earn one ether (i.e., the difficulty divided by 5, as each block earns 5 ether) by the daily ETH/ETC to USD exchange rates [7]. The results of this calculation since the July 2016 fork are shown in Figure 3. We can immediately make three broad observations. *First*, there is a very strong correlation between the expected number of hashes per USD in ETH and ETC; in fact, the curves are almost identical. This suggests the market is very efficient, with the exchange rate being set (and miners making decisions) with the activity of both networks in mind. *Second*, we observe some interesting long-term dynamics: the drop in late October/early November is correlated with the launch of Zcash [28], which suggests that miners left both ETH and ETC to mine in Zcash (which also uses an ASIC-resistant hashing function). One possible explanation for why the hashes-per-USD rallied in November and December is that miners returned from Zcash, which is further reflected in the rise of difficulty in the top of Figure 2. Third, the drop in the expected number of hashes per USD in March is correlated with an increase in the market value of ether; while the difficulty increased slightly, the external value of ether increased much faster.

Security vulnerabilities We now turn to the security implications of the fork. Recall that the original Ethereum protocol was not explicitly designed to experience permanent forks, so no explicit precautions were put in place to deal with the consequences if such a fork occurred. One of the potential security issues that does arise is a form of double-spending: *rebroadcasted transactions*.

[4]Conversely, miners likely did *not* have the choice to switch from Ethereum to cryptocurrencies like Bitcoin that require miners to have specialized hardware to mine competitively

To understand how rebroadcasted transactions could take place, consider a user who owned 10 ether *before* the fork. Because the two forked systems share a common blockchain up until the point of the fork, after the fork, this user would suddenly have *both* 10 ETH and 10 ETC. Many users simply picked one of the two networks to participate in and ignored the other. However, because the message format did not change, any transaction in one of the networks could be *rebroadcasted* in the other; if the source account still had sufficient credit, it would be processed as a valid transaction. Thus, a user who wished to retain control over their fork-generated ETH and ETC should have generated two *new* addresses (one in each network) and transferred the respective ETH and ETC from the common address to the two new addresses. However, not all users did so, meaning the recipient of ETH (or ETC) could immediately rebroadcast the transaction in the ETC (or ETH) network to receive additional funds.

Ethereum Classic was not widely expected to survive, thus the consequences of having two chains were dealt with incrementally. Once it became clear that Classic was not dying off, the Ethereum community advised users on replay attacks, specifically, how to secure their funds by creating chain-specific addresses [8] and later implementing backwards-compatible *chain ids* that users could choose to include in their transactions so they could not be replayed in the other chain [11].[5]

[5]Both Bitcoin and Bitcoin Cash contain a version of chain id's that are not backwards compatible to prevent replay attacks in expectation that both chains would persist [9].

We now explore the extent to which rebroadcasts between ETH and ETC have occurred. Figure 4 shows the percentage of all transactions in ETH and ETC that were originally broadcast in the other network (top graph) and the overall number of such transactions per day (bottom graph). We say that there was an "echo" in ETH if we first saw that same transaction appear in ETC (and vice versa). We can see an initial spike immediately following the fork, followed by subsequent spikes in October and November which appear to correspond with spikes in contract transactions. Additionally, we can see that the overall number of rebroadcasts has fallen off, and yet there are still hundreds of daily rebroadcast transactions even today. It is important to note that not all such rebroadcasts are necessarily attacks, as the user may have intended for the transaction to execute in both networks.

Pool mining As a final point of analysis, we examine how the miners work together in *pools* to establish a more reliable payout method for the miners. In brief, if miners choose to mine on their own, their eventual payout (in terms of the number of blocks mined) is highly variable; mining is essentially a lottery, and a node may expend significant hashing effort and yet not win. To have more predictable payouts, miners instead mine collaboratively in pools: when one of the miners successfully mines a block, the miner transfers the ether mined to an account controlled by the pool, which programmatically splits the reward across all of the miners.[6]

Mining pools are very common in Bitcoin, and they have become common in Ethereum as well. We now examine how the pools are split across the two networks, and whether the distribution of mining power across pools is similar. To do so, we can examine the "winner" of each block, which contains the address to which the 5 ether award are transferred. In the case of a miner working in a pool, this will be the pool's address. Thus, we can immediately observe how much income each pool receives in aggregate each day.

Figure 5 shows the percentage of all blocks mined each day that are mined by the top 1, 3, and 5 pools in both ETH and ETC. Because pools are highly dynamic (pools come and go regularly), we calculate the top pools each day, rather than overall. We draw three interesting observations. *First*, we note that relative fraction of the blocks mined by the top ETH mining pools remains consistent over time; moreover, these ratios are the same as before the fork, indicating that the top mining pools immediately and pervasively chose to migrate to ETH (we verified that the top mining pools' addresses before the fork are consistent across ETH, as well). *Second*, for several months after the fork, the top mining pools in ETC mined a considerably smaller fraction of the

[6]The curious reader may wonder how this protocol is secured from a malicious miner. To receive a share of the reward, typically miners must report their "best" mined blocks (i.e., all blocks above some difficulty target) mined using the pool's header; from this the pool can verify that the miner was, in fact, mining for the pool and payout rewards proportional to mining effort.

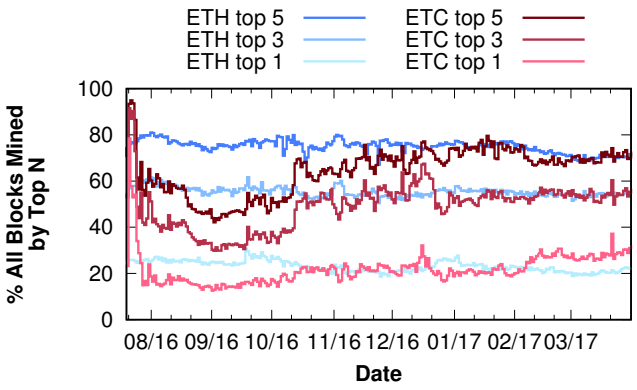

Figure 5: The percent of all mined blocks won by the top 1, 3, and 5 mining pools in ETH and ETC. Though mining pools in each network are distinct, the aggregate mining power distribution is remarkably similar.

blocks. Because the overall difficulty remained largely consistent over that time, this indicates that the sizes of the top mining pools themselves has increased. This coalescing of mining pools has been a relatively slow process, and leads us to our *third* observation: that eventually they have converged on the same relative ratios of mined blocks as the ETH (and pre-fork) mining pools. It is not clear why they have arrived at the same relative ratios—we speculate that it may be indicative of broader, more fundamental market trends, and is an interesting area of future work.

4 CONCLUSION

In this paper, we took the first look at how the Ethereum network responded to the large, persistent fork. We observed a number of interesting trends, most notably the consistency in mining payoff across both networks (despite significant speculation in the ETH network) as well as a consistent distribution of mining strength of pools. Our findings open up a number of interesting avenues for future work, such as exploring the transactions to detect malicious versus benign rebroadcasts, how miners actually moved between both chains, and examining whether the pool mining power distribution is a result of fundamental market trends.

The Ethereum/Ethereum Classic fork and the most recent Bitcoin/Bitcoin Cash fork highlight the fact that as the field of blockchains mature, existing systems will need to fork to upgrade to new standards/demands or risk dying off. If these kinds of forks become the standard, it is crucial to understand the vulnerabilities users and developers need to be concerned about. This paper presents a preliminary look at one persistent fork and is a first step in understanding a type of event that we are likely to see repeated.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This research was supported in part by NSF grants CNS-1409191, CNS-1490249, and CNS-1616234.

REFERENCES

[1] https://ens.domains.

[2] https://medium.com/mit-media-lab-digital-currency-initiative/medrec-electronic-medical-records-on-the-blockchain-c2d7e1bc7d09.

[3] https://chromaway.com/papers/A-blockchain-based-property-registry.pdf.

[4] https://www.bitcoincash.org/.

[5] https://github.com/bitcoin/bips/blob/master/bip-0050.mediawiki.

[6] https://etherscan.io.

[7] https://coinmarketcap.com/currencies/ethereum-classic/https://coinmarketcap.com/currencies/ethereum.

[8] https://blog.ethereum.org/2016/07/26/onward_from_the_hard_fork/.

[9] https://github.com/Bitcoin-UAHF/spec/blob/master/replay-protected-sighash.md.

[10] Contract code size limit #170. https://github.com/ethereum/EIPs/issues/170.

[11] Simple replay attack protection #155. https://github.com/ethereum/EIPs/issues/155.

[12] State trie clearing (invariant-preserving alternative) #161. https://github.com/ethereum/EIPs/issues/161.

[13] P. Agrawal. Fault tolerance in multiprocessor systems without dedicated redundancy. *IEEE Transactions on Computers*, 37(3):358–362, 1988.

[14] L. Anderson, R. Holz, A. Ponomarev, P. Rimba, and I. Weber. New kids on the block: an analysis of modern blockchains. *arXiv preprint arXiv:1606.06530*, 2016.

[15] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun. Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[16] V. Buterin. Ethereum: A next-generation smart contract and decentralized application platform. *URL https://github.com/ethereum/wiki/wiki/White-Paper*, 2014.

[17] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.

[18] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE, 2013.

[19] Enterprise ethereum alliance. https://entethalliance.org/.

[20] P. Koshy, D. Koshy, and P. McDaniel. An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*, pages 469–485. Springer, 2014.

[21] L. Lamport. The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2):133–169, 1998.

[22] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[23] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Internet Measurement Conference*, pages 127–140. ACM, 2013.

[24] M. Moser. Anonymity of bitcoin transactions: An analysis of mixing services. In *Münster Bitcoin Conference (MBC)*, 2013.

[25] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[26] D. Ron and A. Shamir. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[27] M. Yabandeh, N. Knezevic, D. Kostic, and V. Kuncak. Crystalball: Predicting and preventing inconsistencies in deployed distributed systems. In *Symposium on Networked Systems Design and Implementation*, pages 229–244. USENIX, 2009.

[28] Zcash. https://z.cash/.