Tutorial: Blockchain Consensus Unraveled: Virtues and Limitations*

Suyash Gupta, Jelle Hellings, Sajjad Rahnama, Mohammad Sadoghi
Exploratory Systems Lab, Department of Computer Science
University of California, Davis, CA 95616-8562, USA

ABSTRACT

Since the introduction of Bitcoin—the first wide-spread application driven by blockchains—the interest of the public and private sector in blockchains has skyrocketed. At the core of this interest are the ways in which blockchains can be used to improve data management, e.g., by enabling federated data management via decentralization, resilience against failure and malicious actors via replication and consensus, and strong data provenance via a secured immutable ledger.

In practice, high-performance blockchains for data management are usually built in permissioned environments in which the participants are vetted and can be identified. In this setting, blockchains are typically powered by Byzantine fault-tolerant consensus protocols. These consensus protocols are used to provide full replication among all honest blockchain participants by enforcing an unique order of processing incoming requests among the participants.

In this tutorial, we take an in-depth look at Byzantine fault-tolerant consensus. First, we take a look at the theory behind replicated computing and consensus. Then, we delve into how common consensus protocols operate. Finally, we take a look at current developments and briefly look at our vision moving forward.

CCS CONCEPTS

• **Information systems → Distributed database transactions**; • **Theory of computation → Distributed algorithms**; • **Computer systems organization → Dependable and fault-tolerant systems and networks**.

KEYWORDS

resilient transaction processing, permissioned blockchains, consensus, geo-scale, sharding, cluster-sending, Byzantine learning

ACM Reference Format:
Suyash Gupta, Jelle Hellings, Sajjad Rahnama, Mohammad Sadoghi. 2020. Tutorial: Blockchain Consensus Unraveled: Virtues and Limitations. In *The 14th ACM International Conference on Distributed and Event-based Systems (DEBS '20), July 13–17, 2020, Virtual Event, QC, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3401025.3404099

*This tutorial is based on the outline of our upcoming book on fault-tolerant transaction processing on blockchains [38]. Furthermore, this tutorial is an evolution of a tutorial presented at Middleware 2019 [34]. We have updated that tutorial with new and upcoming techniques and insights.

DEBS '20, July 13–17, 2020, Virtual Event, QC, Canada

ACM ISBN 978-1-4503-8028-7/20/07.
https://doi.org/10.1145/3401025.3404099

1 INTRODUCTION

Since the introduction of *Bitcoin*—the first wide-spread application driven by *blockchain*—the interest of the public and private sector in blockchain has skyrocketed. Recently, this interest has cumulated in the introduction of several blockchain-inspired database systems and blockchain fabrics [2, 3, 25, 26, 59, 60]. Blockchain-based systems have also been demonstrated to address challenges in various other fields such as the trade of valuable commodities [11, 15, 68], food production [29], managing land property rights [66], managing identities [4, 11, 66], supporting transparent aid delivery [4, 66], managing health care data [7, 31, 50], insurance fraud prevention [49], energy production and energy trading [68], and managing compliance with the GDPR [16].

In each of these systems and use cases, blockchain technology is used to improve data managament in one way or another. Blockchain technology—which is at the intersection of database systems, distributed systems, and cryptography—provides *promises for new directions in data management* by utilizing and combining underused techniques from distributed systems and cryptography in novel and powerful ways [10, 19, 47, 58].

The role of Bitcoin, Ethereum, and other cryptocurrencies in these developments cannot be understated: at their core, Bitcoin and Ethereum provided proof of the viability of blockchain techniques in non-trivial large-scale settings. At the core of these cryptocurrencies is the maintenance of a *blokchain* among a group of replicas. This blockchain holds an ordered record of all transactions and this record is secured against changes using cryptographic primitives. To assure that all replicas agree on the same set of transactions and maintain the same blockchain, new transactions are agreed upon via a consensus protocol (which are fault-tolerant counterparts of the classical two-phase and three-phase commit protocols used in distributed database systems [32, 42, 43, 69, 70, 72]). From our perspective, these technologies can strengthen data management in three vital directions:

(1) The blockchain structure in Bitcoin and Ethereum utilizes cryptographic primitives to harden against any unwanted changes (providing immutability). These techniques can also be used to secure a traditional database journal against tampering, which is a major step towards providing systems that provide reliable *data provenance*. Such tamper-resistant data storage can be used to irrefutably proof any claims about the current state of the data.
(2) Fault-tolerant consensus—used to replicate the transactions that are part of the blockchain—is designed to deal with *malicious behavior* of some of the replicas. The same techniques can also be used to deal with malicious behavior caused by the compromise (e.g., due to a cyberattack) of some of

the replicas in a system. In this way, blockchain technology promises one way to harden against cyberattacks on data-based services, which can reduce the huge societal and economic impacts of such attacks [33, 61–63, 73].

(3) Finally, fault-tolerant consensus can also be used as the technique for supporting *federated data management*, the collective management of a single database among various stakeholders. Federated data management is in itself a major step towards dealing with *data quality issues* arising from the non-federated interchange of information between various stakeholders and, as such, can reduce the huge negative economic impact of bad data [24, 48, 71]. Moreover, in this federated setting, the immutable and irrefutable structure of the blockchain can further help in *policing disputes* in cases where some stakeholders do not trust each other.

The explosion of blockchain based applications, products, and proof-of-concepts (see, e.g., [3, 10, 57, 78]), has led to the development of several different approaches towards consensus. We can roughly categorize these approaches into *permissionless blockchains*, as used in public settings by Bitcoin and other cryptocurrencies, and *permissioned blockchains*, which are better suited for managed private environments.

In *permissionless blockchains* such as Bitcoin, *Proof-of-Work*-inspired consensus algorithms are used to replicate data [41, 57, 78]. These algorithms require *limited communication* between replicas and can operate in unstructured peer-to-peer networks in which independent parties can join and leave at any time [67]. Proof-of-Work uses computationally complex puzzles to limit the influence any malicious party has on the evolution of the blockchain. At the same time, these puzzles incur a high *computational costs* on all parties, which has raised questions about the sustainability of the *energy consumption* of permissionless blockchain systems [17, 77]. Additionally, the complexity of Proof-of-Work puzzles causes relative long transaction processing times (minutes to hours) and significantly limits the number of transactions a permissionless blockchain can handle: in 2017, it was reported that Bitcoin can only process 7 transactions per second, whereas Visa already processes 2000 transactions per second on average [66]. Finally, the design of Proof-of-Work prevents scalability, as adding more computational power to the network will only increase the cost of Proof-of-Work puzzles. Other permissionless consensus algorithms based on Proof-of-Work, such as Proof-of-Space and Proof-of-Stake have similar limitations with respect to resource usage and transaction throughput.

Most data management use cases do not require the flexibility of unstructured peer-to-peer networks in which participants can join and leave at any time: data management systems are usually employed in a managed environment in which all participants are known, can be identified, and are vetted. This is exactly the setting for which *permissioned blockchains* are designed. In these blockchains, traditional Byzantine fault-tolerant replication techniques based on *consensus algorithms* such as PBFT are employed to accept, order, and execute client transactions among all replicas [5, 6, 13, 52, 53, 64, 75, 76]. The benefit of these consensus algorithms, compared to Proof-of-Work-based algorithms, is that they have low computational costs, low transaction processing times, and high transaction throughput. This is already exemplified in 2002 by *BFS*, a fault-tolerant version of the networked file system [44], which could already handle hundreds of transactions per second [13, 14].

In this tutorial, we will provide a deep dive into consensus protocols with a focus on data management. To do so, we take an in-depth look at Byzantine fault-tolerant consensus protocols, the main technique powering permissioned blockchains.

Our tutorial will focus on three avenues. First, we look at the theoretical framework in which permissioned blockchains operate. Then, we look at practical high-performance consensus protocols and at current developments. Finally, we look at the challenges in the design of future-proof high-performance permissioned blockchain systems that can deal with huge amounts of data, and provide our vision on future developments. These avenues are detailed in the following three sections. Finally, in Section 5, we provide a conclusion on the tutorial.

2 RESILIENT DISTRIBUTED COMPUTING

Blockchains are, at their basis, fully replicated distributed systems that aim to maintain data consistency. The well-known CAP Theorem puts restrictions on the types of failures these blockchains can deal with while guaranteeing continued services [8, 9, 30]. The CAP Theorem puts rather general limitations on the design of blockchains, however. More specific limitations are also known, as the Byzantine consensus problem and other related problems, such as the Byzantine agreement problem and the interactive consistency problem, have received considerable attention.

It is well-known that the Byzantine agreement problem can only be solved when using synchronous communication [28, 56, 74]. In a synchronous environment with $\mathbf{n}$ replicas of which $\mathbf{f}$ are Byzantine (e.g., malicious), Byzantine agreement requires that $\mathbf{n} > 3\mathbf{f}$ [20, 21]. When strong cryptographic primitives are available, this can be improved to $\mathbf{n} > \mathbf{f}$ [23, 54, 65] (although practical systems will still require $\mathbf{n} > 2\mathbf{f}$). Additionally, bounds on the amount of communication and the quality of the network are known [18, 20–23, 27].

3 PRACTICAL CONSENSUS PROTOCOLS

Having provided a theoretical background, we make the step toward detailing practical consensus protocols. We do so by a full coverage of the *Practical Byzantine Fault Tolerant* consensus protocol (PBFT) of Castro et al. [12–14]. Next, we also look at the lineage of consensus protocols that refine and improve PBFT. This detailed overview will cover many of the practical consensus protocols currently in use and, simultaneously, also covers recent developments. Our coverage will include protocols such as PBFT, HotStuff [79], Zyzzyva [5, 51, 52], FaB [55], SynBFT [1], RBFT [6], and PoE [35].

4 CHALLENGES AND OUR VISION

As outlined above, the approaches taken by permissionless and permissioned blockchains towards fault-tolerant replication have *benefits* and practical use cases. Unfortunately, fault-tolerant replication is *challenged* by the scalability and performance required by many modern big-data-driven applications. In specific, we see that there is no obvious way to scale up fault-tolerant replication:

adding more replicas will only increase the cost of replication and decrease the throughput of the system, even when using the most efficient consensus protocols.

We will close our tutorial by discussing recent steps toward the design of new fault-tolerant architectures that step away from the full-replicated nature of blockchains, this to increase scalability and the ability to serve big-data-driven applications. To put our vision in practice, we will first look at two low-level techniques, *cluster-sending* [45] and *delayed-replication* [46]. Next, we look at high-level designs enabled by these techniques to provide high-performance parallelized consensus [36, 37] and to provide high-performance consensus in sharded and geo-scale aware architectures [39]. Finally, we illustrate the architecture and design of ResilientDB, our state-of-the-art permissioned blockchain fabric [40].

5 CONCLUSION

In this tutorial, we present both a high-level overview of the role permissioned blockchains can play in data management and transaction processing. We show how data management and transaction processing can benefit from transaction processing, provide a detailed overview of relevant theory, techniques, and the current state-of-the-art, and, finally, discuss our vision for future directions for improvements of permissioned blockchains.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Small Business Innovation Research, under Award Number DE-SC0020455.

REFERENCES

[1] Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. 2018. Synchronous Byzantine Agreement with Expected $O(1)$ Rounds, Expected $O(n^2)$ Communication, and Optimal Resilience.

[2] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. CAPER: A Cross-application Permissioned Blockchain. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1385–1398. https://doi.org/10.14778/3342263.3342275

[3] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference*. ACM, 30:1–30:15. https://doi.org/10.1145/3190508.3190538

[4] GSM Association. 2017. Blockchain for Development: Emerging Opportunities for Mobile, Identity and Aid. https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2017/12/Blockchain-for-Development.pdf

[5] Pierre-Louis Aublin, Rachid Guerraoui, Nikola Knežević, Vivien Quéma, and Marko Vukolić. 2015. The Next 700 BFT Protocols. *ACM Transactions on Computer Systems* 32, 4 (2015), 12:1–12:45. https://doi.org/10.1145/2658994

[6] Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. 2013. RBFT: Redundant Byzantine Fault Tolerance. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 297–306. https://doi.org/10.1109/ICDCS.2013.53

[7] Burkhard Blechschmidt. 2018. *Blockchain in Europe: Closing the Strategy Gap*. Technical Report. Cognizant Consulting. https://www.cognizant.com/whitepapers/blockchain-in-europe-closing-the-strategy-gap-codex3320.pdf

[8] Eric Brewer. 2012. CAP twelve years later: How the "rules" have changed. *Computer* 45, 2 (2012), 23–29. https://doi.org/10.1109/MC.2012.37

[9] Eric A. Brewer. 2000. Towards Robust Distributed Systems (Abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*. ACM, 7–7. https://doi.org/10.1145/343477.343502

[10] Christian Cachin and Marko Vukolic. 2017. Blockchain Consensus Protocols in the Wild (Keynote Talk). In *31st International Symposium on Distributed Computing (Leibniz International Proceedings in Informatics)*, Vol. 91. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 1:1–1:16. https://doi.org/10.4230/LIPIcs.DISC.2017.1

[11] Michael Casey, Jonah Crane, Gary Gensler, Simon Johnson, and Neha Narula. 2018. *The Impact of Blockchain Technology on Finance: A Catalyst for Change*. Technical Report. International Center for Monetary and Banking Studies. https://www.cimb.ch/uploads/1/1/5/4/115414161/geneva21_1.pdf

[12] Miguel Castro. 2001. *Practical Byzantine Fault Tolerance*. Ph.D. Dissertation. Massachusetts Institute of Technology. http://hdl.handle.net/1721.1/86581

[13] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. USENIX Association, 173–186.

[14] Miguel Castro and Barbara Liskov. 2002. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Transactions on Computer Systems* 20, 4 (2002), 398–461. https://doi.org/10.1145/571637.571640

[15] Christie's. 2018. Major Collection of the Fall Auction Season to be Recorded with Blockchain Technology. https://www.christies.com/presscenter/pdf/9160/RELEASE_ChristiesxArtoryxEbsworth_9160_1.pdf

[16] Cindy Compert, Maurizio Luinetti, and Bertrand Portier. 2018. *Blockchain and GDPR: How blockchain could address five areas associated with GDPR compliance*. Technical Report. IBM Security. https://public.dhe.ibm.com/common/ssi/ecm/61/en/61014461usen/security-ibm-security-solutions-wg-white-paper-external-61014461usen-20180319.pdf

[17] Alex de Vries. 2018. Bitcoin's Growing Energy Problem. *Joule* 2, 5 (2018), 801–805. https://doi.org/10.1016/j.joule.2018.04.016

[18] Richard A. DeMillo, Nancy A. Lynch, and Michael J. Merritt. 1982. Cryptographic Protocols. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*. ACM, 383–400. https://doi.org/10.1145/800070.802214

[19] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. 2018. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering* 30, 7 (2018), 1366–1385. https://doi.org/10.1109/TKDE.2017.2781227

[20] D. Dolev. 1981. Unanimity in an unknown and unreliable environment. In *22nd Annual Symposium on Foundations of Computer Science*. IEEE, 159–168. https://doi.org/10.1109/SFCS.1981.53

[21] Danny Dolev. 1982. The Byzantine generals strike again. *Journal of Algorithms* 3, 1 (1982), 14–30. https://doi.org/10.1016/0196-6774(82)90004-9

[22] Danny Dolev and Rüdiger Reischuk. 1985. Bounds on Information Exchange for Byzantine Agreement. *J. ACM* 32, 1 (1985), 191–204. https://doi.org/10.1145/2455.214112

[23] D. Dolev and H. Strong. 1983. Authenticated Algorithms for Byzantine Agreement. *SIAM J. Comput.* 12, 4 (1983), 656–666. https://doi.org/10.1137/0212045

[24] Wayne W. Eckerson. 2002. *Data quality and the bottom line: Achieving Business Success through a Commitment to High Quality Data*. Technical Report. The Data Warehousing Institute, 101communications LLC.

[25] Muhammad El-Hindi, Carsten Binnig, Arvind Arasu, Donald Kossmann, and Ravi Ramamurthy. 2019. BlockchainDB: A Shared Database on Blockchains. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1597–1609. https://doi.org/10.14778/3342263.3342636

[26] Muhammad El-Hindi, Martin Heyden, Carsten Binnig, Ravi Ramamurthy, Arvind Arasu, and Donald Kossmann. 2019. BlockchainDB – Towards a Shared Database on Blockchains. In *Proceedings of the 2019 International Conference on Management of Data*. ACM, 1905–1908. https://doi.org/10.1145/3299869.3320237

[27] Michael J. Fischer and Nancy A. Lynch. 1982. A lower bound for the time to assure interactive consistency. *Inform. Process. Lett.* 14, 4 (1982), 183–186. https://doi.org/10.1016/0020-0190(82)90033-3

[28] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of Distributed Consensus with One Faulty Process. *J. ACM* 32, 2 (1985), 374–382. https://doi.org/10.1145/3149.214121

[29] Lan Ge, Christopher Brewster, Jacco Spek, Anton Smeenk, and Jan Top. 2017. *Blockchain for agriculture and food: Findings from the pilot study*. Technical Report. Wageningen University. https://www.wur.nl/nl/Publicatie-details.htm?publicationId=publication-way-353330323634

[30] Seth Gilbert and Nancy Lynch. 2002. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. *SIGACT News* 33, 2 (2002), 51–59. https://doi.org/10.1145/564585.564601

[31] William J. Gordon and Christian Catalini. 2018. Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability. *Computational and Structural Biotechnology Journal* 16 (2018), 224–230. https://doi.org/10.1016/j.csbj.2018.06.003

[32] Jim Gray. 1978. Notes on Data Base Operating Systems. In *Operating Systems, An Advanced Course*. Springer-Verlag, 393–481. https://doi.org/10.1007/3-540-08755-9_9

[33] Andy Greenberg. 2018. The Untold Story of NotPetya, the Most Devastating Cyberattack in History. https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/

[34] Suyash Gupta, Jelle Hellings, Sajjad Rahnama, and Mohammad Sadoghi. 2019. An In-Depth Look of BFT Consensus in Blockchain: Challenges and Opportunities. In *Proceedings of the 20th International Middleware Conference Tutorials*. ACM, 6–10. https://doi.org/10.1145/3366625.3369437

[35] Suyash Gupta, Jelle Hellings, Sajjad Rahnama, and Mohammad Sadoghi. 2019. Proof-of-Execution: Reaching Consensus through Fault-Tolerant Speculation. https://arxiv.org/abs/1911.00838
[36] Suyash Gupta, Jelle Hellings, and Mohammad Sadoghi. 2019. Brief Announcement: Revisiting Consensus Protocols through Wait-Free Parallelization. In *33rd International Symposium on Distributed Computing (DISC 2019) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 146. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 44:1–44:3. https://doi.org/10.4230/LIPIcs.DISC.2019.44
[37] Suyash Gupta, Jelle Hellings, and Mohammad Sadoghi. 2019. Scaling Blockchain Databases through Parallel Resilient Consensus Paradigm. https://arxiv.org/abs/1911.00837
[38] Suyash Gupta, Jelle Hellings, and Mohammad Sadoghi. 2020. *Fault-Tolerant Distributed Transactions on Blockchains*. (to appear).
[39] Suyash Gupta, Sajjad Rahnama, Jelle Hellings, and Mohammad Sadoghi. 2020. ResilientDB: Global Scale Resilient Blockchain Fabric. *Proceedings of the VLDB Endowment* 13, 6 (2020), 868–883. https://doi.org/10.14778/3380750.3380757
[40] Suyash Gupta, Sajjad Rahnama, and Mohammad Sadoghi. 2020. Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned. In *40th International Conference on Distributed Computing Systems*. IEEE.
[41] Suyash Gupta and Mohammad Sadoghi. 2018. *Blockchain Transaction Processing*. Springer International Publishing, 1–11. https://doi.org/10.1007/978-3-319-63962-8_333-1
[42] Suyash Gupta and Mohammad Sadoghi. 2018. EasyCommit: A Non-blocking Two-phase Commit Protocol. In *Proceedings of the 21st International Conference on Extending Database Technology*. Open Proceedings, 157–168. https://doi.org/10.5441/002/edbt.2018.15
[43] Suyash Gupta and Mohammad Sadoghi. 2020. Efficient and non-blocking agreement protocols. *Distributed and Parallel Databases* 38 (2020), 287–333. https://doi.org/10.1007/s10619-019-07267-w
[44] Thomas Haynes and David Noveck. 2015. RFC 7530: Network File System (NFS) Version 4 Protocol. https://tools.ietf.org/html/rfc7530
[45] Jelle Hellings and Mohammad Sadoghi. 2019. Brief Announcement: The Fault-Tolerant Cluster-Sending Problem. In *33rd International Symposium on Distributed Computing (DISC 2019) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 146. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 45:1–45:3. https://doi.org/10.4230/LIPIcs.DISC.2019.45
[46] Jelle Hellings and Mohammad Sadoghi. 2020. Coordination-Free Byzantine Replication with Minimal Communication Costs. In *23rd International Conference on Database Theory (ICDT 2020) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 155. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 17:1–17:20. https://doi.org/10.4230/LIPIcs.ICDT.2020.17
[47] Maurice Herlihy. 2019. Blockchains from a Distributed Computing Perspective. *Commun. ACM* 62, 2 (2019), 78–85. https://doi.org/10.1145/3209623
[48] Thomas N. Herzog, Fritz J. Scheuren, and William E. Winkler. 2007. *Data Quality and Record Linkage Techniques*. Springer New York. https://doi.org/10.1007/0-387-69505-2
[49] Matt Higginson, Johannes-Tobias Lorenz, Björn Münstermann, and Peter Braad Olesen. 2017. *The promise of blockchain*. Technical Report. McKinsey&Company. https://www.mckinsey.com/industries/financial-services/our-insights/the-promise-of-blockchain
[50] Maged N. Kamel Boulos, James T. Wilson, and Kevin A. Clauson. 2018. Geospatial blockchain: promises, challenges, and scenarios in health and healthcare. *International Journal of Health Geographics* 17, 1 (2018), 1211–1220. https://doi.org/10.1186/s12942-018-0144-x
[51] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. 2007. Zyzzyva: Speculative Byzantine Fault Tolerance. In *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*. ACM, 45–58. https://doi.org/10.1145/1294261.1294267
[52] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. 2009. Zyzzyva: Speculative Byzantine Fault Tolerance. *ACM Transactions on Computer Systems* 27, 4 (2009), 7:1–7:39. https://doi.org/10.1145/1658357.1658358
[53] Leslie Lamport. 1978. The implementation of reliable distributed multiprocess systems. *Computer Networks (1976)* 2, 2 (1978), 95–114. https://doi.org/10.1016/0376-5075(78)90045-4
[54] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* 4, 3 (1982), 382–401. https://doi.org/10.1145/357172.357176
[55] Jean-Philippe Martin and Lorenzo Alvisi. 2006. Fast Byzantine Consensus. *IEEE Transactions on Dependable and Secure Computing* 3, 3 (2006), 202–215. https://doi.org/10.1109/TDSC.2006.35
[56] Shlomo Moran and Yaron Wolfstahl. 1987. Extended impossibility results for asynchronous complete networks. *Inform. Process. Lett.* 26, 3 (1987), 145–151. https://doi.org/10.1016/0020-0190(87)90052-4
[57] Satoshi Nakamoto. 2009. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
[58] Arvind Narayanan and Jeremy Clark. 2017. Bitcoin's Academic Pedigree. *Commun. ACM* 60, 12 (2017), 36–45. https://doi.org/10.1145/3132259
[59] Senthil Nathan, Chander Govindarajan, Adarsh Saraf, Manish Sethi, and Praveen Jayachandran. 2019. Blockchain Meets Database: Design and Implementation of a Blockchain Relational Database. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1539–1552. https://doi.org/10.14778/3342263.3342632
[60] Faisal Nawab and Mohammad Sadoghi. 2019. Blockplane: A Global-Scale Byzantizing Middleware. In *35th International Conference on Data Engineering (ICDE)*. IEEE, 124–135. https://doi.org/10.1109/ICDE.2019.00020
[61] Dick O'Brie. 2017. *Internet Security Threat Report: Ransomware 2017, An ISTR Special Report*. Technical Report. Symantec. https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-ransomware-2017-en.pdf
[62] The Council of Economic Advisers. 2018. *The Cost of Malicious Cyber Activity to the U.S. Economy*. Technical Report. Executive Office of the President of the United States. https://www.whitehouse.gov/wp-content/uploads/2018/03/The-Cost-of-Malicious-Cyber-Activity-to-the-U.S.-Economy.pdf
[63] National Audit Office. 2018. Investigation: WannaCry cyber attack and the NHS. https://www.nao.org.uk/report/investigation-wannacry-cyber-attack-and-the-nhs/
[64] Michael Okun. 2016. *Byzantine Agreement*. Springer New York, 255–259. https://doi.org/10.1007/978-1-4939-2864-4_60
[65] M. Pease, R. Shostak, and L. Lamport. 1980. Reaching Agreement in the Presence of Faults. *J. ACM* 27, 2 (1980), 228–234. https://doi.org/10.1145/322186.322188
[66] Michael Pisa and Matt Juden. 2017. *Blockchain and Economic Development: Hype vs. Reality*. Technical Report. Center for Global Development. https://www.cgdev.org/publication/blockchain-and-economic-development-hype-vs-reality
[67] Bitcoin Project. 2018. Bitcoin Developer Guide: P2P Network. https://bitcoin.org/en/developer-guide#p2p-network
[68] PwC. 2016. Blockchain – an opportunity for energy producers and consumers? https://www.pwc.com/gx/en/industries/energy-utilities-resources/publications/opportunity-for-energy-producers.html
[69] Thamir Qadah, Suyash Gupta, and Mohammad Sadoghi. 2020. Q-Store: Distributed, Multi-partition Transactions via Queue-oriented Execution and Communication. In *Proceedings of the 23nd International Conference on Extending Database Technology*. OpenProceedings.org, 73–84. https://doi.org/10.5441/002/edbt.2020.08
[70] Thamir M. Qadah and Mohammad Sadoghi. 2018. QueCC: A Queue-oriented, Control-free Concurrency Architecture. In *Proceedings of the 19th International Middleware Conference*. ACM, 13–25. https://doi.org/10.1145/3274808.3274810
[71] Thomas C. Redman. 1998. The Impact of Poor Data Quality on the Typical Enterprise. *Commun. ACM* 41, 2 (1998), 79–82. https://doi.org/10.1145/269012.269025
[72] Dale Skeen. 1982. *A Quorum-Based Commit Protocol*. Technical Report. Cornell University.
[73] Symantec. 2018. Internet Security Threat Report, Volume 32. https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf
[74] Gadi Taubenfeld and Shlomo Moran. 1996. Possibility and impossibility results in a shared memory environment. *Acta Informatica* 33, 1 (1996), 1–20. https://doi.org/10.1007/s002360050034
[75] Gerard Tel. 2001. *Introduction to Distributed Algorithms* (2nd ed.). Cambridge University Press.
[76] Maarten van Steen and Andrew S. Tanenbaum. 2017. *Distributed Systems* (3th ed.). Maarten van Steen. https://www.distributed-systems.net/
[77] Harald Vranken. 2017. Sustainability of bitcoin and blockchains. *Current Opinion in Environmental Sustainability* 28 (2017), 1–9. https://doi.org/10.1016/j.cosust.2017.04.011
[78] Gavin Wood. 2016. Ethereum: a secure decentralised generalised transaction ledger. https://gavwood.com/paper.pdf EIP-150 revision.
[79] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus with Linearity and Responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*. ACM, 347–356. https://doi.org/10.1145/3293611.3331591