Demo Abstract: Attack and Vulnerability Simulation Framework for Bitcoin-like Blockchain Technologies

Fabian Schüssler
Technical University of Munich
fabian.schuessler@in.tum.de

Pezhman Nasirifard
Technical University of Munich
p.nasirifard@tum.de

Hans-Arno Jacobsen
Technical University of Munich
arno.jacobsen@tum.de

Abstract

Despite the very high volatility of the cryptocurrency markets, the interest in the development and adaptation of existing cryptocurrencies such as Bitcoin as well as new distributed ledger technologies is increasing. Therefore, understanding the security and vulnerability issues of such blockchain systems plays a critical role. In this work, we propose a configurable distributed simulation framework for analyzing Bitcoin-like blockchain systems which are based on Proof-of-Work protocols. The simulator facilitates investigating security properties of blockchain systems by enabling users to configure several characteristics of the blockchain network and executing different attack scenarios, such as double-spending attacks and flood attacks and observing the effects of the attacks on the blockchain network.

CCS Concepts • **Networks → Peer-to-peer networks**;

Keywords Blockchain, Bitcoin, Simulation

1 Introduction

Understanding the security and vulnerability issues of different blockchain systems, such as Bitcoin, plays a significant role as the popularity of Bitcoin-like systems increases. In this work, we propose a simulation framework for simulating different types of attacks on Bitcoin-like blockchains. The proposed framework is a configurable blockchain simulator capable of conducting distributed large-scale network simulations and simulating different types of attacks, such as double-spending attacks and flood attacks, for deriving empirical insights on the behavior of the system under attack. A double-spending attack on Bitcoin represents a situation where the malicious miners try to outpace the longest valid chain. A transaction spam attack, also known as flood attack, represents a scenario where one or multiple nodes disrupt the normal operation of the network by issuing a large number of spam transactions. The spam transactions can increase the transaction costs and the transaction confirmation time.

Middleware Posters and Demos '18, December 10–14, 2018, Rennes, France
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Related Work

We built this work upon *VIBES*, a generic blockchain simulator for simulating the issuance of transactions, the block mining and the interaction of thousands of miners [5]. Our work extends VIBES with the latest implementation of Bitcoin and enables Bitcoin attack simulations. In addition to VIBES, other works attempted to simulate distributed ledger technologies [1, 2, 4]. However, other works focus solely on a single aspect of the Bitcoin network, where [1] simulates the block mining and effects of mining rewards and transaction fees, and [2, 4] focus on simulating the Bitcoin network and propagation of messages. In contrast to other works, we propose a simulator which addresses every aspect of a Bitcoin network, as well as offering a rich user interface for configuring the simulation environment and investigating the simulation results.

3 Technical Approach

Figure 1 demonstrates the architecture of our system which is based on the VIBES simulator. We make use of and improve several concepts introduced by VIBES, such as the fast-forward computing and the Coordinator Node. We also make use of VIBES primary blockchain configuration panel, which enables the user with providing several main simulation metrics such as the block size limit, block time, and transaction sizes.

We provide a web-based user interface where the user configures the different network and attack parameters. For demonstrating a double-spending attack, the user can configure the following simulation metrics: the attacker's hash-rate as a percentage of the network's total hash-rate (q), the attack duration as the number of mined blocks (d) and the number of desired transaction confirmations (n). Once the user provides all the simulation inputs, the simulation starts, and the system assigns every simulated miner node an *honesty attribute* based on the attacker's hash-rate which specifies if the node is *honest* or *malicious*. After the nodes started with mining on the genesis block, the network separates into an honest public blockchain and a private malicious blockchain. The attackers include their transactions for double-spending attempts into the public blockchain, and the malicious nodes

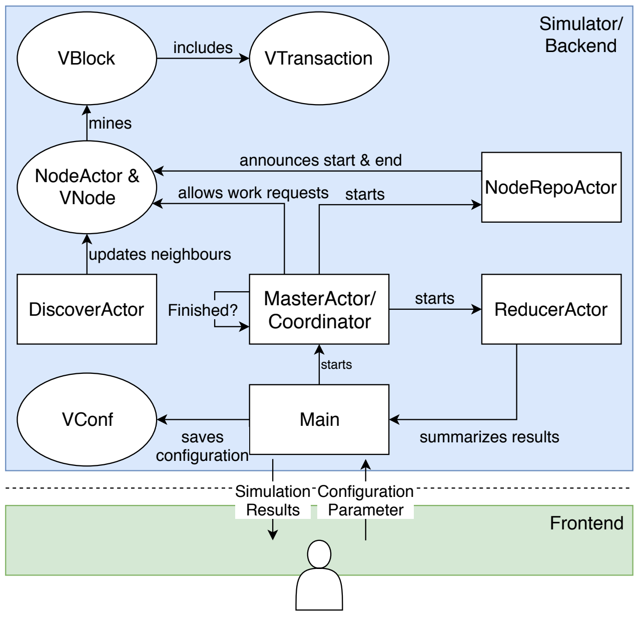


Figure 1. Architecture of the simulator.

start mining on the competing blockchain. The attack succeeds if the malicious blockchain is longer than the honest blockchain after the fraudulent transactions reached the required number of confirmations. For the attack to fail, the honest network's blockchain needs to stay longer until the end of the attack simulation. After the outcome of the attack is decided, both networks merge back together. The system visualizes the blockchain tree and the branch selection and shows the results of the simulation such as the attack success probability, and the maximal safe transaction value which is the maximal transaction value a merchant should accept; otherwise, it would be profitable for an attacker to attempt a double-spending attack. The system tests the theoretical and empirical outcomes of the double-spending attack against the calculated success probability r. It has to be taken into account that the simulation needs an attack duration to decide if the attack failed, but the Equation 1 does not factor in the attack duration, but we consider this effect negligible with $d = 50$. Table 1 compares the results of 101 simulations with the calculated double-spending attack success probability.

$$r = \begin{cases} 1 - \sum_{m=0}^{n} \frac{(m+n-1)!}{m!\,(n-1)!}(p^n q^m - p^m q^n), & \text{if } q < p \\ 1, & \text{if } q \geq p \end{cases} \quad (1)$$

Figure 2. Double-spending success probability where p is the honest network's percentage of the total hash-rate and $p = 1 - q$. [3]

For simulating a flood attack, the user requires to specify one parameter, the target transaction fee, which is the minimum fee that the attacker desires to gain from every transaction issuer in the network. In our implementation, the Coordinator sends requests to the nodes to create the transactions which are marked with a *transaction spam* attribute. In the case of a very short block generation time, transactions might not get propagated fast enough to the miner, who includes the transactions in the next block. To ensure that every transaction issuer in every block requires the targeted transaction fee, the attacker has a buffer of spam transactions in the transaction pool, which leads to maintaining the target fee in the event of very short block generation times despite the existence of transactions with lower fees in the transaction pool. In our simulations, a buffer of double the size of maximal transactions per block ensures the attacker's goal. After the simulation, the system provides a report containing several metrics such as the number of confirmed flood attack transactions, the attacker's spent transaction fees in Bitcoin and the number of confirmed transactions below the target fee.

Table 1. Outcomes of a double-spending attack and their simulated and expected probabilities with a sample size of 101 simulations ($n = 6$, $q = 30$, $d = 50$).

Outcome	Simulated Probability	Expected Probability
Attack successful	13.86%	15.645%
Attack failed	86.13%	84.355%
TOTAL	**100%**	**100%**

4 Conclusions

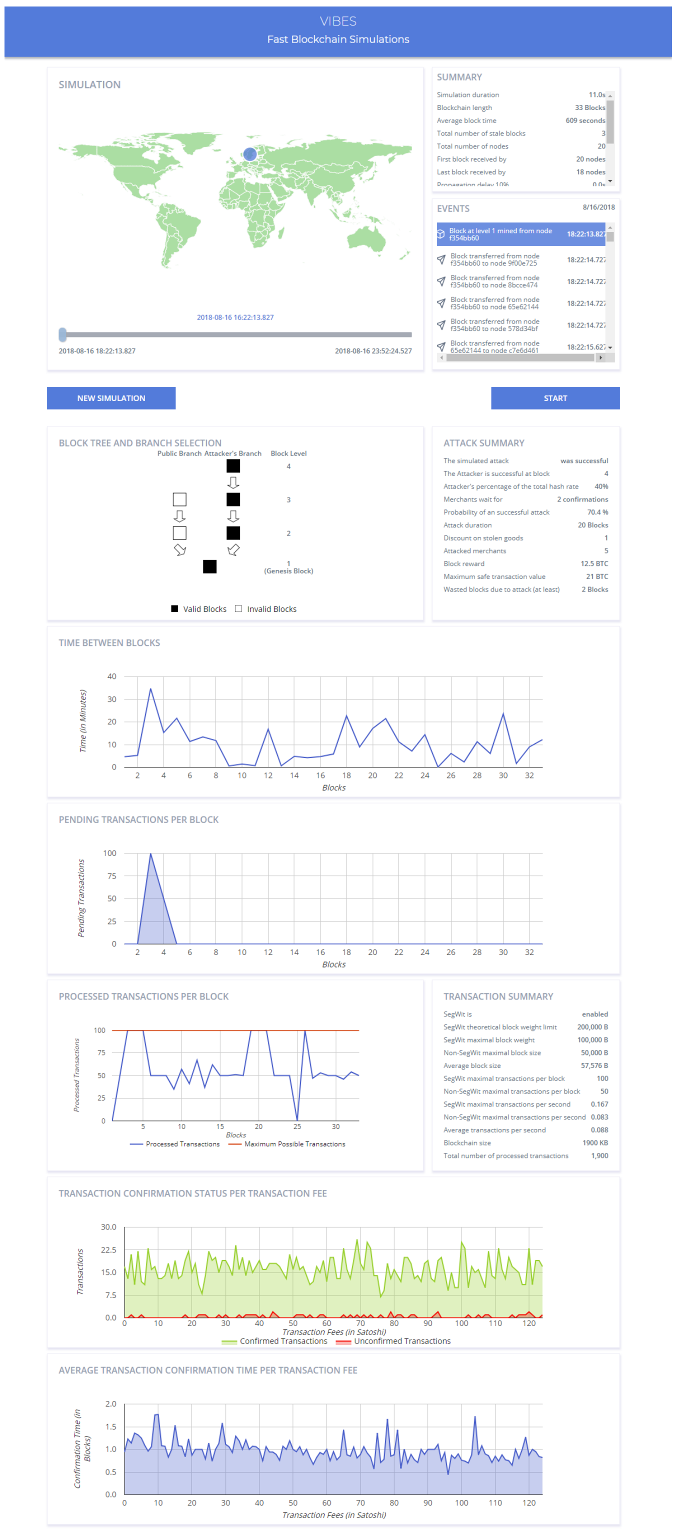
In this demo, we present an approach for simulating a few attacks on Bitcoin-like blockchain networks. Furthermore, the users can configure several simulation metrics, and the simulations outputs are displayed in detailed reports for the user to explore the security properties.

Acknowledgments

Alexander von Humboldt Foundation supported this project.

References

[1] M. Carlsten, H. Kalodner, M. Weinberg, and A. Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 154–167.
[2] T. Neudecker, P. Andelfinger, and H. Hartenstein. 2015. A simulation model for analysis of attacks on the Bitcoin peer-to-peer network. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 1327–1332.
[3] M. Rosenfeld. 2014. Analysis of Hashrate-Based Double Spending. *CoRR* (2014).
[4] M. B. Sarrias. 2015. *Bitcoin Network Simulator Data Explotation*. Master's thesis. Open University of Catalonia.
[5] L. Stoykov, K. Zhang, and H. A. Jacobsen. 2017. VIBES: Fast Blockchain Simulations for Large-scale Peer-to-peer Networks: Demo. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos*. ACM, 19–20.



SUMMARY

Simulation duration	11.0s
Blockchain length	33 Blocks
Average block time	609 seconds
Total number of stale blocks	3
Total number of nodes	20
First block received by	20 nodes
Last block received by	18 nodes
Propagation delay 10%	0.0s

EVENTS 8/16/2018

Block at level 1 mined from node f354bb60	18:22:13.827
Block transferred from node f354bb60 to node 9f00e725	18:22:14.727
Block transferred from node f354bb60 to node 8bcce474	18:22:14.727
Block transferred from node f354bb60 to node 65e62144	18:22:14.727
Block transferred from node f354bb60 to node 578d34bf	18:22:14.727
Block transferred from node 65e62144 to node c7e6d461	18:22:15.627

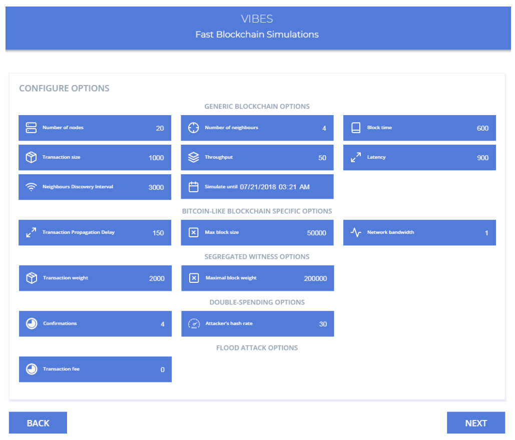
ATTACK SUMMARY

The simulated attack	was successful
The Attacker is successful at block	4
Attacker's percentage of the total hash rate	40%
Merchants wait for	2 confirmations
Probability of an successful attack	70.4 %
Attack duration	20 Blocks
Discount on stolen goods	1
Attacked merchants	5
Block reward	12.5 BTC
Maximum safe transaction value	21 BTC
Wasted blocks due to attack (at least)	2 Blocks

TRANSACTION SUMMARY

SegWit is	enabled
SegWit theoretical block weight limit	200,000 B
SegWit maximal block weight	100,000 B
Non-SegWit maximal block size	50,000 B
Average block size	57,576 B
SegWit maximal transactions per block	100
Non-SegWit maximal transactions per block	50
SegWit maximal transactions per second	0.167
Non-SegWit maximal transactions per second	0.083
Average transactions per second	0.088
Blockchain size	1900 KB
Total number of processed transactions	1,900

Figure 3. Simulation results of a double-spending attack.



CONFIGURE OPTIONS

GENERIC BLOCKCHAIN OPTIONS

Number of nodes	20
Number of neighbours	4
Block time	600
Transaction size	1000
Throughput	50
Latency	900
Neighbours Discovery Interval	3000
Simulate until	07/21/2018 03:21 AM

BITCOIN-LIKE BLOCKCHAIN SPECIFIC OPTIONS

Transaction Propagation Delay	150
Max block size	50000
Network bandwidth	1

SEGREGATED WITNESS OPTIONS

Transaction weight	2000
Maximal block weight	200000

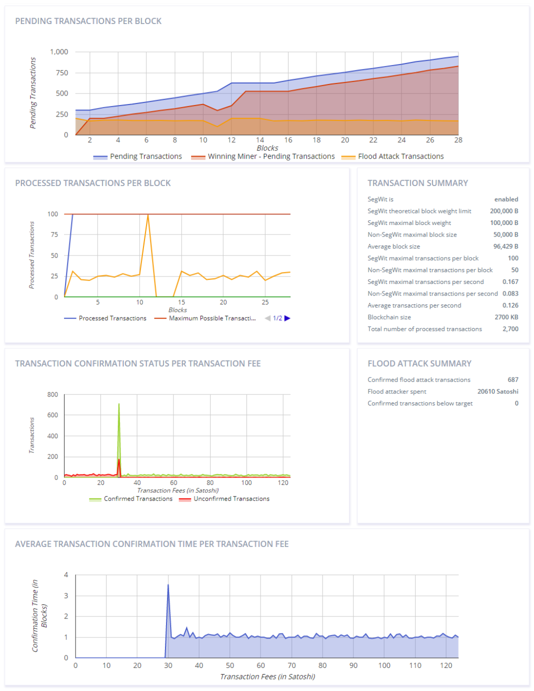
DOUBLE-SPENDING OPTIONS

Confirmations	4
Attacker's hash rate	30

FLOOD ATTACK OPTIONS

Transaction fee	0

Figure 4. The configuration options.


PENDING TRANSACTIONS PER BLOCK
Pending Transactions
Blocks
Pending Transactions
Winning Miner - Pending Transactions
Flood Attack Transactions
PROCESSED TRANSACTIONS PER BLOCK
Processed Transactions
Blocks
Processed Transactions
Maximum Possible Transacti...
1/2
TRANSACTION CONFIRMATION STATUS PER TRANSACTION FEE
Transactions
Transaction Fees (in Satoshi)
Confirmed Transactions
Unconfirmed Transactions
AVERAGE TRANSACTION CONFIRMATION TIME PER TRANSACTION FEE
Confirmation Time (in Blocks)
Transaction Fees (in Satoshi)

TRANSACTION SUMMARY

SegWit is	enabled
SegWit theoretical block weight limit	200,000 B
SegWit maximal block weight	100,000 B
Non-SegWit maximal block size	50,000 B
Average block size	96,429 B
SegWit maximal transactions per block	100
Non-SegWit maximal transactions per block	50
SegWit maximal transactions per second	0.167
Non-SegWit maximal transactions per second	0.083
Average transactions per second	0.126
Blockchain size	2700 KB
Total number of processed transactions	2,700

FLOOD ATTACK SUMMARY

Confirmed flood attack transactions	687
Flood attacker spent	20610 Satoshi
Confirmed transactions below target	0

Figure 5. Part of the simulation results of a flood attack.

5 Demonstration Remarks

Figure 3, Figure 4, and Figure 5 are snapshots of the simulation results of a double-spending attack, and some part of the simulation results for a transaction spam attack, and of the configuration panel, respectively, as described in Section 3. We developed the web frontend based on *React*[1] and the backend based on *Scala*[2] and *Akka*[3]. We offer the source code of both under this URL https://github.com/i13-msrg/vibes/tree/FabianSchuessler.

[1]https://reactjs.org/
[2]https://www.scala-lang.org/
[3]https://akka.io/