UNIVERSITY of NICOSIA

Week 3 – Session 6

Hierarchical and Alternative Blockchain Structures

BLOC 512: Blockchain Systems and Architectures

Session Objectives

- Scaling blockchains still remains a great challenge with many protocols proposing their own solutions. The linear structure of a blockchain, where each block links to the previous and every node is storing the full history of the ledger it is restricting the overall transaction throughput rate for the network.
- Several approaches exists in the literature ranging from Layer 2 solutions (e.g., Lighting, Plasma), to bigger blocks, and even replacing the linear blockchain architecture.
- In this session we will be briefly discussing ideas of replacing the linear blockchain in favor of a more scalable architecture. From the literature, several proposals suggest the replacement of a linear blockchain with Direct Acyclic Graphs and sharding.
    - We will be looking into IOTA as an indicative example.

Agenda

1. Intro to Sharding
2. DAGs: The case of IOTA
3. Conclusions
4. Self-Assessment Exercises

Sharding

First Layer Solutions

- Despite the unique characteristics of DLTs/blockchains, a key challenge of the technology has been scalability
- Early implementations of many protocols including Bitcoin performed relatively poor in terms of the number of transactions that can be processed per second - small and insufficient
- The blockchain trilemma as described by Vitalik states that trade-offs are inevitable between three important properties: decentralization, scalability, and security.
- The main challenge is to achieve scalability, security, and decentralization at the same time. Several solutions have been proposed in the literature, such as sharding, Directed Acyclic Graph, and the Lightning Network.
- In this session we will be briefly introducing Sharding and Direct Acyclic Graphs by looking IOTA as a tangible use-case.

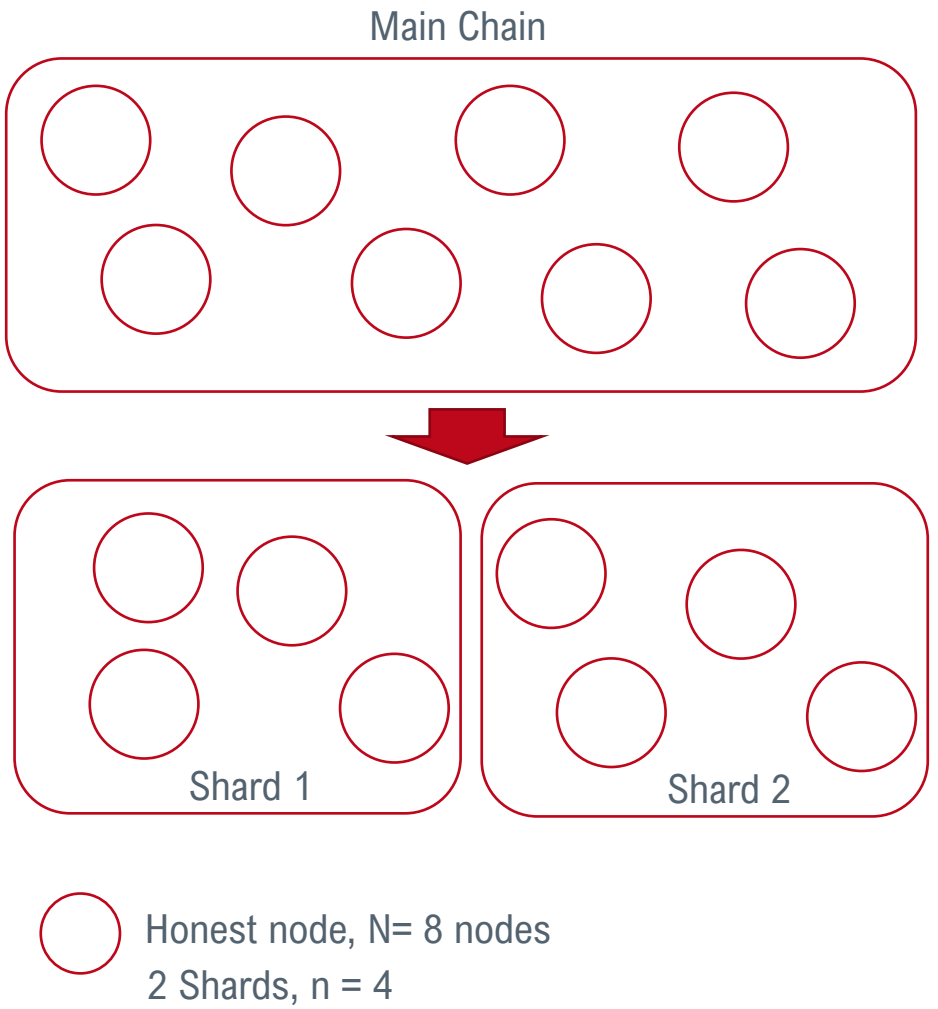
Scalability Factors

- Several factors impact blockchain scalability, these include:

  - **Throughput:** It is the number of confirmed transactions per second.
  - **Storage:** If all transactions are recorded on-chain, the data structure's size will considerably increase. In time the average block size in megabytes (MB) will increase yielding in an increase in the total size of the blockchain. This will create a demand on storage and time for downloading the entire chain.
  - **Cost:** Once a transaction is confirmed, transaction fees are paid to the miner (i.e., node that created the block where the transaction is included). Thus, it will be more economic for the user to conduct as many transactions as possible outside of the blockchain (i.e., off-chain) and then later record them as one transaction on-chain.
  - **Latency (aka confirmation time):** This refers to the time between submitting a transaction to the blockchain and the first confirmation of acceptance by the network. The greater number of transactions, the verification time increases; each transaction requires consensus and peer-to-peer verification.

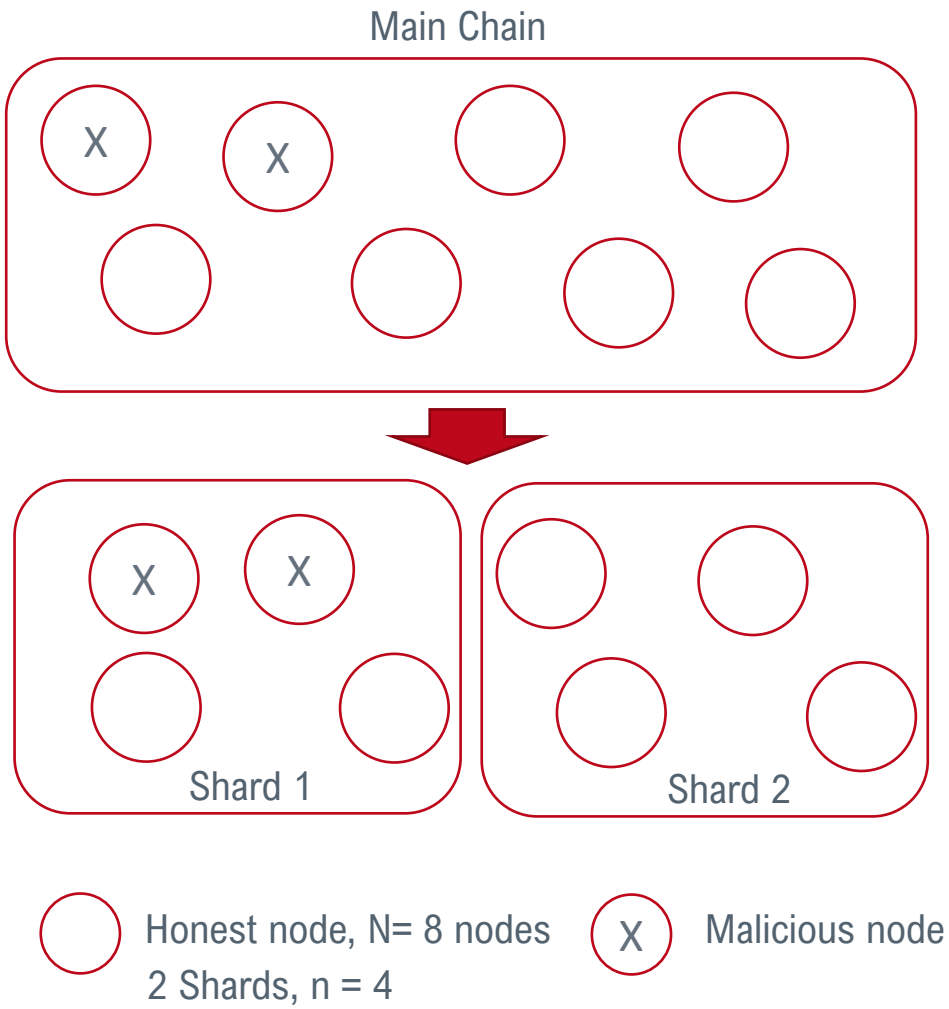
Layer 1 Solution: Sharding

- **Sharding:** The main idea of this technique is to divide or split the network into sub-committees, called ***shards***; each shard is responsible for processing a different set of transactions, rather than the entire network processing the same or all the transactions.
- This allows the network to scale with the numbers of shards, allowing the throughput and the storage to achieve high efficiency.
- **The catch:** potential compromise of security
  - A major challenge is understanding how work is divided in such a network. There must be safeguards in place to ensure that dishonest validators cannot overwhelm a particular shard.



Layer 1 Solution: Sharding

- **Security:** All shards need to satisfy the byzantine validator limit (aka, committee resiliency), which is the maximum percentage of malicious validators/nodes the shard can handle. For most BFT-like consensus algorithms, this limit is up to 33% on average. Beyond that limit the network is considered fundamentally insecure and unstable.
- The main challenge is the fact that even if the whole network falls well under that limit, a single shard could be compromised.
- For example, if we assume a network with 25% malicious nodes is split evenly into 2 shards and more than 33% of malicious nodes in the network end up in one shard, the network will be insecure. This is known as a ***single shard takeover attack.***
- **Committee resiliency:** The maximum number of malicious nodes that the committee can tolerate while still being secure.



Sharding-based Blockchain Protocols

- **Elastico:**
  - Proposed as the first public sharding-based blockchain protocol, that tolerates byzantine adversaries. It divides the network into multiple sub-committees where each is responsible for a separate set of transactions, aka shards. This proposal uses a PoW for committee formation and a BFT consensus for the intra-committee consensus. The number of shards grows almost linearly with the size of the network. For 1600 nodes, the transaction throughput reaches 40 tx/s.
  - These protocols can linearly scale intra-shard transactions, i.e., transactions that can be processed within a single shard, by adding more shards to the system. However, they encounter a performance bottleneck when it comes to cross-shard transactions (, i.e., transactions that require coordination from multiple shards)
  - Can only tolerate up to 25% of malicious/faulty nodes (total resiliency) and 33% of malicious nodes in each sub-committee (committee resiliency)
- **OmniLedger:**
  - The protocol makes use of a bias resistant randomness protocol to ensure security. Similarly, to Elastico, OmniLedger employs PoW for committee formation and a BFT variant for intra-committee consensus. OmniLedger uses a Byzantine shard atomic commit (known as Atomix) to deal with cross-shard transactions.
  - From the literature, and for 1800 nodes the network can handle up to 500 tx/s
  - Can only tolerate up to 25% of malicious/faulty nodes (total resiliency) and 33% of malicious nodes in each sub-committee (committee resiliency)

Sharding-based Blockchain Protocols

- **RapidChain:**
  - This shading-based protocol outperforms existing sharding algorithms (e.g., Elastico, OmniLedger) in terms of throughput and security. Cross-shard transactions in RapidChain rely on an inter-committee routing scheme which is based on the routing-algorithm of Kademlia.
  - RapidChain can tolerate up to 33% of malicious nodes in the network, and 50% of malicious nodes in each committee. For 1800 nodes it claims 4220 tx/s.
- **Zilliqa:**
  - Zilliqa's sharding design allows the network to process transactions in parallel and reach a high throughput. Nodes are randomly assigned to a shard. Each shard acts like a parallel chain that processes a set of transaction requests that do not exist in other shards. Note that in Zilliqa the state of the blockchain data is not divided just the network nodes.
  - After each shard comes to a consensus a "micro-block" is generated for that shard. Once this step finishes a committee combines the outputs from each shared and reaches consensus on it.
  - Zilliqa employs a practical byzantine fault tolerance protocol aka PBFT for consensus within each shard. Only effective with a small number of nodes, and it runs under the assumption that up to 33% of the nodes in each shard can be malicious.
  - The ability to process transactions in parallel due to the sharded architecture ensures that the throughput in Zilliqa linearly increases with the size of the network.

https://blog.zilliqa.com/https-blog-zilliqa-com-the-zilliqa-design-story-piece-by-piece-part1-d9cb32ea1e65

Sharding-based Blockchain Protocols

- **Harmony:**
  - This shading-based protocol divides not only the network nodes into shards but also the blockchain states into shards, scaling linearly in all three aspects of machines, transactions and storage. To prevent single shard attacks, they employ a re-sharding technique among a random number of nodes per shard. Each shard has a pre-defined number of nodes (i.e., 250) to provide a strong security guarantee against Byzantine behaviors.
  - Harmony claims the same intra and inter resiliency as Zilliqa, Elastico, and OmniLedger.
- **Ethereum Sharding 2.0:**
  - It is designed in three phases:
    1. **Beacon Chain:** which manages all shards in the network; more specifically, it applies consensus rules, rewards and penalties to validators, and manages validators and their stakes.
    2. **Shard Chains**: that enable parallel transactions.
    3. **State Execution:** this is where the operations of the entire system are executed; it introduces the concept of "Execution Environments (EEs)", which provides a smart contract functionality similar to Ethereum 1.0
  - Ethereum employs a mechanism referred to as "the receipt paradigm" in order to enable cross-shard communication. Under this paradigm, every transaction generates a receipt. These receipts will be eventually stored on the beacon chain via distributed shared memory; this means that receipts can be viewed by other shards - yet remain unable to be modified.
  - Ethereum 2.0 also implements the so-called Casper PoS consensus algorithm that will replace the current EthHash and enable the transition to Ethereum sharding 2.0.

Sharding-based Blockchain Protocols

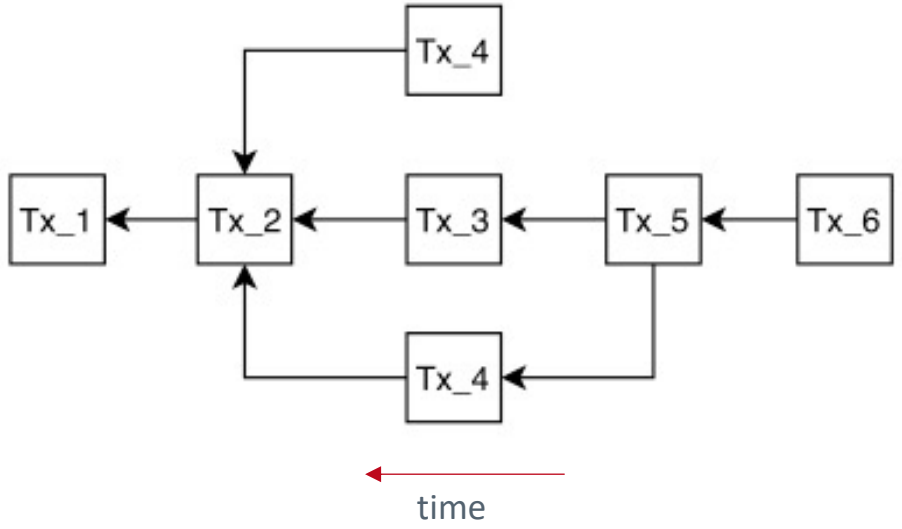
Protocol	Transaction Model	Consensus	Committee Resiliency	Total Resiliency
Elastico	UTXO	PoW, PBFT	33%	25%
OmniLedger	UTXO	BFT, PoW	33%	25%
RapidChain	UTXO	BFT, PoW	50%	33%
Zilliqa	Account	PoW, PBFT	33%	33%
Harmony	Account	PoS, BFT	33%	25%
Ethereum v2.0	Account	PoS, BFT	33%	33%

Table shows a quick overview of sharding-based blockchain protocols adapted from Hafid, A., Hafid, A. S., & Samih, M. (2020). Scaling blockchains: A comprehensive survey (here). IEEE Access, 8, 125244-125262.

Directed acyclic graphs (DAGs)

Intro

- Directed Acyclic Graph (DAG) is a tree data structure that differs from traditional blockchains.
- Instead of utilizing a linear sequenced blocks structure, transactions/blocks are structured in the form of a DAG.
  - In its simplest instantiation, a DLT network that employs a DAG organizes transactions as a tree branching out from one transaction to another.
- Such DLT networks consequently re-build the upper layer components of the architecture including consensus, incentives, etc.
- The promise of a DAG-based DLT architecture is to enable fast confirmation times (complete transactions within million seconds) and high scalability (attach transactions in parallel) without significantly compromising security.



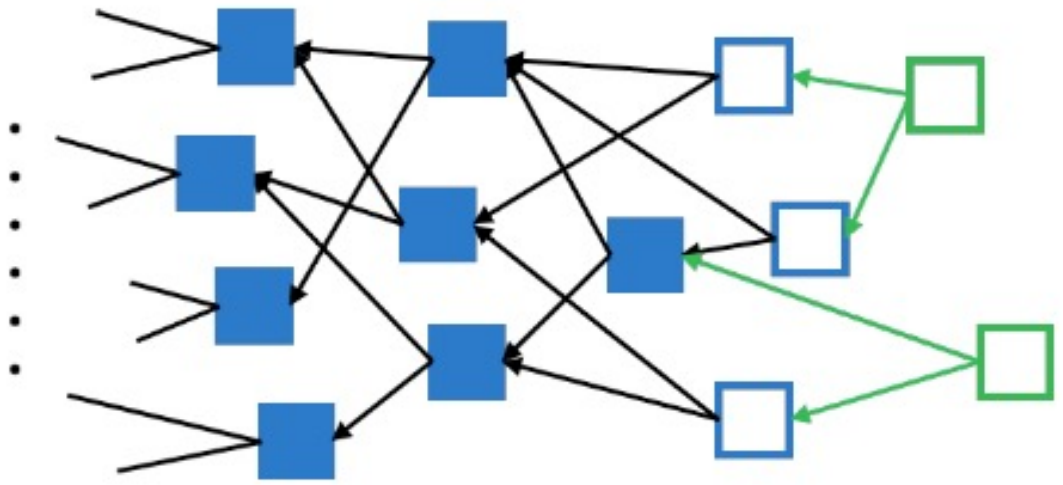
IOTA

- IOTA is a scalable, public distributed ledger. An architecture that is designed specifically for Internet of Things (IoT).
- The core feature of IOTA is the Tangle technology, which is a DAG adapted for storing transactions for a decentralized network.
- **Tangle:** IOTA's distributed ledger, does not comprise of transactions grouped into blocks and stored in an ordered sequence. Instead, each vertex represents an individual transaction tangled together with another transaction over an edge (i.e., a link) that represents the transaction hash.

https://www.iota.org/foundation/research-papers

IOTA

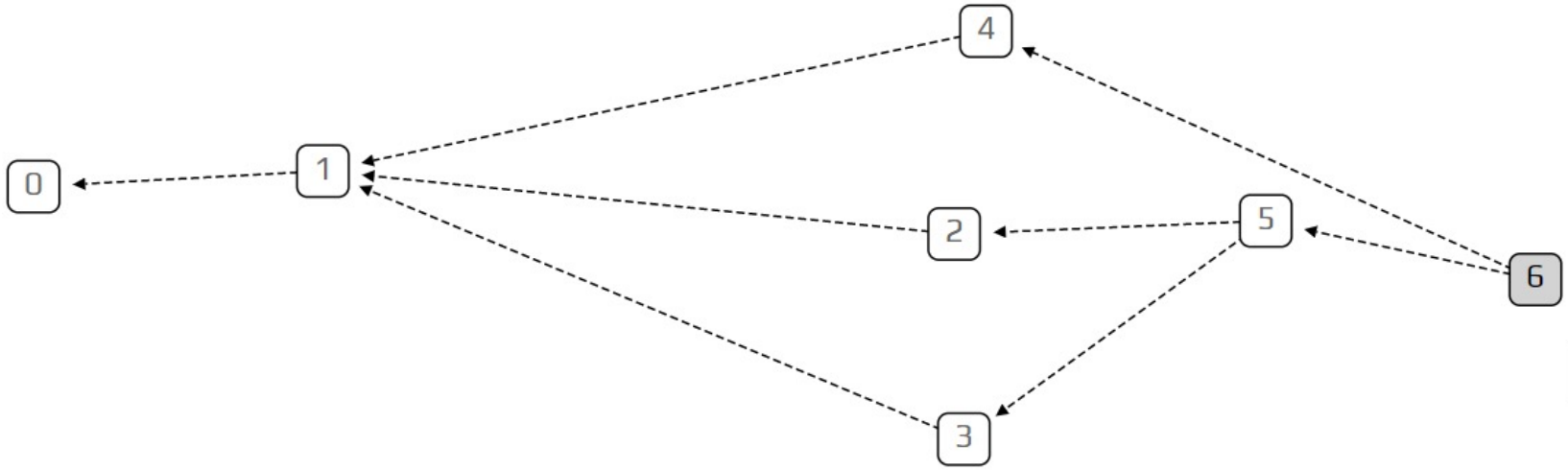
- **Submitting transactions:** To append a new transaction to the DAG, a user needs to approve (i.e., validate) two unapproved transactions (known as tips) of the DAG.
- If there is only one tip available, the user chooses an approved transaction instead. The user then attaches the hashes of the two chosen transactions to the new transaction, and works on a PoW puzzle before broadcasting it.
- When multiple conflicting tips (i.e., double-spending) are detected by a user, only one can be deemed valid and approved.
- In this case the user resorts to a tip-selection scheme to choose one that yields the highest acceptance probability in the long term.
- Other conflicting tips are considered invalid and then orphaned. Tangle currently employs a Markov-chain Monte Carlo (MCMC) based scheme to simulate the acceptance probabilities of each of the conflicting tips.



Green nodes: new transactions
Bordered blue: unapproved transactions
Solid blue: approved/confirmed transactions

https://www.iota.org/foundation/research-papers

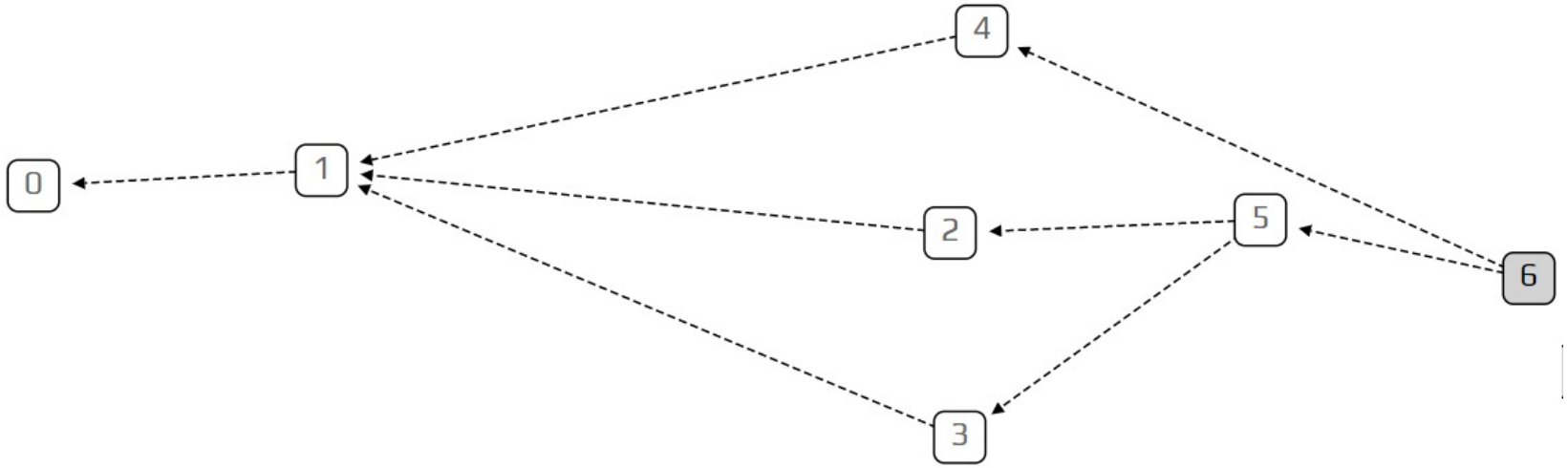
IOTA Simulation



Experiment with the IOTA Tangle Visualization
https://public-rdsdavdrpd.now.sh/

Github: https://github.com/iotaledger/iotavisualization

IOTA Simulation

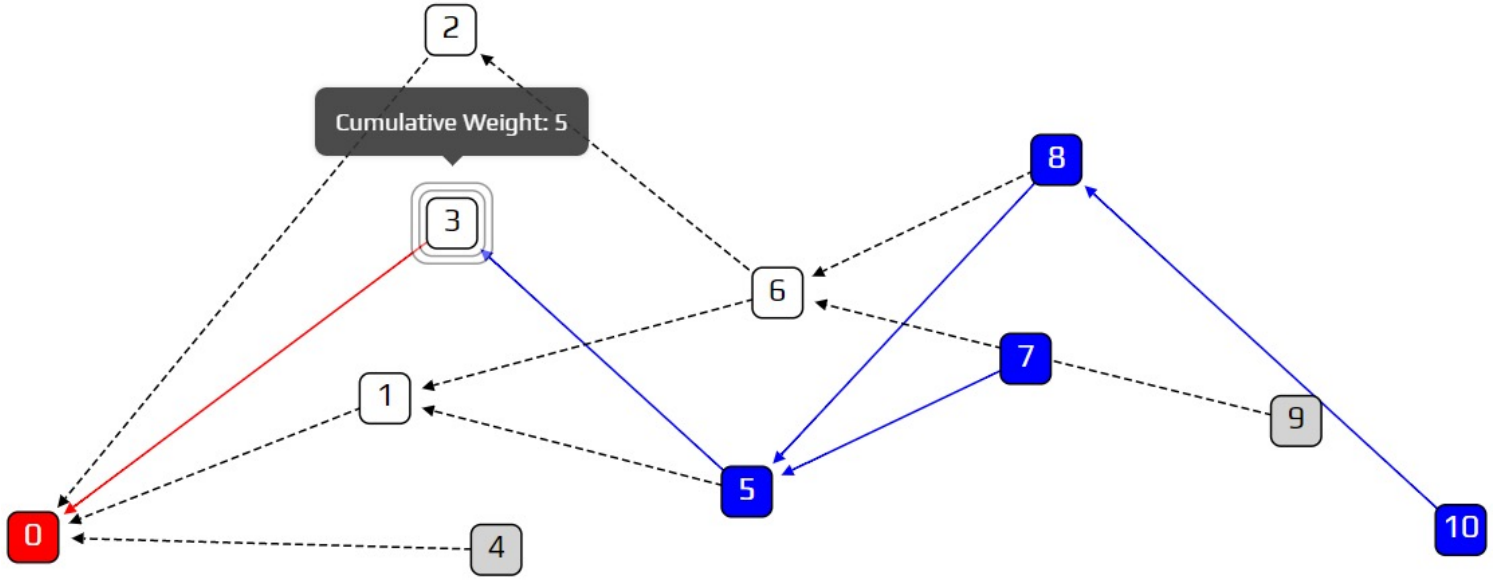


- **Experiment with IOTA tangle**
  - This simulator implements a random selection strategy aka uniform random tip selection.
  - All tips are marked with a gray square
  - When you put your mouse over a transaction, all transactions approved by that node are highlighted in red, and those which approved it are marked by blue.

Experiment with the IOTA Tangle Visualization
https://public-rdsdavdrpd.now.sh/

Github: https://github.com/iotaledger/iotavisualization

IOTA Simulation: Preventing Double Spends



- **Experiment with IOTA tangle simulator**
  - Challenge for honest users of IOTA: which branch should they approve?
  - How IOTA prevents double spends?

Experiment with the IOTA Tangle Visualization
https://public-qnbiiqwyqj.now.sh/

Github: https://github.com/iotaledger/iotavisualization

Conclusions

Conclusions

- Blockchain sharding is proposed as a promising solution to split the work and improve scalability, reduce latency, and process more transactions in less time. However, there are several challenges that increase their complexity e.g., the number of network shards, splitting the blockchain state, and how to provide strong guarantees against Byzantine behaviors.
- The idea of using a DAG has been utilized by many notable projects in the space e.g., IOTA, OByte, and Nano. These architectures support the idea that it is not important for all the nodes to have a global state but rather nodes should only need to know the local state that is relevant to them. Therefore, having enough connections (i.e., links) to other nodes to verify their local state is enough to confirm its validity.
- A limitation that DAG architectures phase is that are vulnerable to Eclipse attacks; possible when an attacker can monopolize the incoming connections of a victim node.

Glossary

Glossary

Data type: is a data item which is defined by the values it can take.
Data Structure: a data structure is a collection of data values, the relationships among them
Graph: graph is a data structure consisting of nodes that are connected by edges

Self-Assessment Exercises and Further Readings

Self-Assessment Exercises

- Obyte (previously Byteball) proposes a DAG-based blockchain architecture. Studying the details from the whitepaper, discuss the three main differences when compared with IOTA Tangle?

You are welcome to share your findings on the forums!

Further Reading

- Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018, May). Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE Symposium on Security and Privacy (SP)* (pp. 583-598). IEEE.
- Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016, October). A secure sharding protocol for open blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 17-30).
- Ullah, I., De Roode, G., Meratnia, N., & Havinga, P. (2021). Threat Modeling—How to Visualize Attacks on IOTA?. Sensors, 21(5), 1834.
- Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse attacks on bitcoin's peer-to-peer network. In 24th {USENIX} Security Symposium ({USENIX} Security 15) (pp. 129-144).
- Müller, S., Penzkofer, A., Kuśmierz, B., Camargo, D., & Buchanan, W. J. (2020, November). Fast probabilistic consensus with weighted votes. In Proceedings of the Future Technologies Conference (pp. 360-378). Springer, Cham.
- Guangmin, L. (2009, June). An improved Kademlia routing algorithm for P2P network. In 2009 International Conference on New Trends in Information and Service Science (pp. 63-66). IEEE.
- Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016, October). A secure sharding protocol for open blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 17-30).

UNIVERSITY *of* NICOSIA

Questions?

Contact Us:

Twitter: @mscdigital
Instructor's Email: christodoulou.kl@unic.ac.cy

Course Support:
Mark Wigmans - wigmans.m@unic.ac.cy
Marios Touloupos - touloupos.m@unic.ac.cy

IT & live session support: dl.it@unic.ac.cy