CST4125: Blockchain Development
Week: 10
Title: Consensus Algorithms

Dr Ian Mitchell

Bedfordshire,
UK

2023

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Office Hours

- During term time only
- When: Autumn Term: Mondays 1100-1300hrs
- Please read notifications or emails
- There are occassions that these could be arranged online, e.g., due to industrial action or inclement weather

Deadlines

				Feedback	
Description	Submission	Weight	Deadline	Formative	Summative
1. Hyperledger	MyLearning	50%	18th December 2022	LW11-12	12/01/2023
2. Ethereum	MyLearning	50%	2nd April 2023	LW23-24	24/04/2023
Resits	MyLearning	50-100%	1st July 2023	None	None
Deferals	MyLearning	50-100%	1st July 2023	None	None

Lecture

Aims

- Critically appraise consensus algorithms
- Review Business Network Archives (BNA)
- Formative Feedback
- Assessment Criteria

Objectives

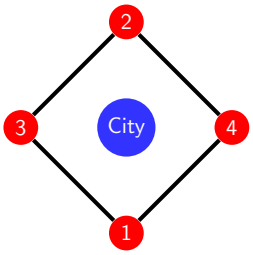
Knowledge

To explain different consensus algorithms [8, 5]

- Consensus?
- PoW: Proof-of-Work
- PoS: Proof-of-Stake
- PoET: Proof-of-Elapsed-Time
- PBFT: Byzantium Algorithms
- PoA: Proof-of-Authority

Consensus

- Byzantine Generals Problem [4]
- Reliable Complex System must cope with failure of one or more of its components
- A failed component may send conflicting information
- Each division commanded by its own General
- The Generals can communicate with each other
- Need a common plan of action
- Trust: Traitor in their midst preventing a consensus



Consensus

Conditions

1. All loyal Generals decide upon the same plan of action
2. A small number of traitors cannot cause the loyal Generals to adopt a bad plan

- Traitors can do what they wish
- Loyal Generals will all do what they are told
- $v(i)$ be information communicated by the i^{th} General. So, you have $v(1), v(2), \ldots, v(n)$, where there are n Generals

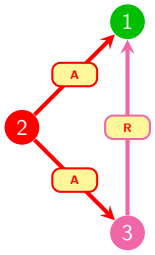
Byzantine Generals Problem

3-node

- 3 nodes, messages 'A' and 'R' for Attack and Retreat, respectively

Byzantine Generals Problem

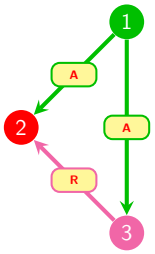
3-node



- 3 nodes, messages 'A' and 'R' for Attack and Retreat, respectively
- 1 and 2 are loyal; 3 is disloyal.
- 2 sends the same message to 3 and 1
- 3 changes the message and sends to 1
- 1 receives conflicting messages

Byzantine Generals Problem

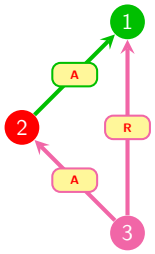
3-node



- 1 and 2 are loyal; 3 is disloyal.
- 1 sends the same message
- 3 changes the messages and sends to 2
- 2 receives conflicting messages

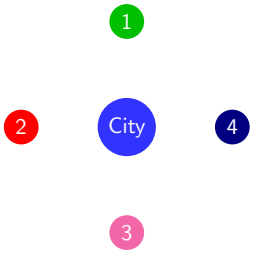
Byzantine Generals Problem

3-node

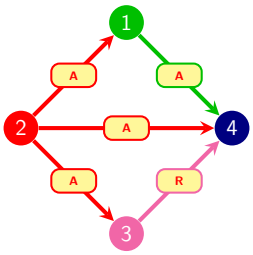


- 1 and 2 are loyal; 3 is disloyal.
- 3 sends different messages to 1 and 2
- 2 forwards the message to 1
- 1 receives conflicting messages
- needs to be $3m + 1$, where there are m traitors
- informal proof, formal proof [6]

Byzantine Generals Problem
4-node

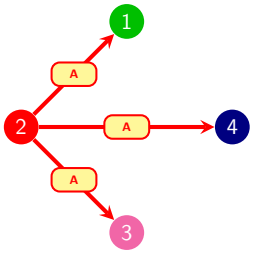


Byzantine Generals Problem
4-node



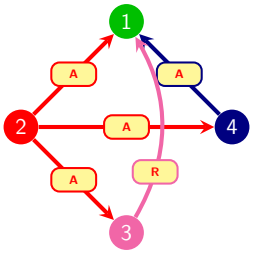
- 4 nodes, messages 'A' and 'R' for Attack and Retreat, respectively
- 1, 2 and 4 are loyal; 3 is disloyal.
- 2 sends the same message to all nodes
- 3 changes the message and sends to 4
- 4 receives conflicting messages
- 4 acts on majority of messages 'A'

Byzantine Generals Problem
4-node



- 4 nodes, messages 'A' and 'R' for Attack and Retreat, respectively
- 1, 2 and 4 are loyal; 3 is disloyal.
- 2 sends the same message to all nodes

Byzantine Generals Problem
4-node



- 4 nodes, messages 'A' and 'R' for Attack and Retreat, respectively
- 1, 2 and 4 are loyal; 3 is disloyal.
- 2 sends the same message to all nodes
- 3 changes message and sends to 1
- 1 receives conflicting messages
- 1 acts on majority of messages 'A'

PBFT? [2]

- cited alot, least understood
- [4] 5500+
- [2] 2500+
- n number of nodes in network
- $[0, 1, 2, 3, ..., n-2, n-1]$
- f Max. bad nodes the network can tolerate

PBFT Equations

$$n = nodes \tag{1}$$

$$f = \frac{n-1}{3} \tag{2}$$

$$n = 3f + 1 \tag{3}$$

Consensus
adapted from [8]

- Initial state is agreed
- Users agree to the consensus model
- Every block is linked to the previous block
- Users can verify the process independently
- Nodes are deterministic

Voting-based adpated from [3]

- Users vote for nodes to commit to blockchain
- Vote is weighted and tied to stake
- Nodes with most votes, publish blocks
- Publishing nodes become trustworthy
- Untrustworthy publishing nodes become disreputable and receive less votes

Round-Robin

- Permissioned
- nodes take turns in publishing blocks
- Timeout limits on unavailable nodes, when it is their turn
- low resources
- not suited to permissionless networks
  - Malicious nodes could add more nodes to increase their probability of selection
  - Take over the blockchain system

Proof-of-Work

- Challenge to solve a very difficult puzzle
- Extremely hard to solve
- Very easy to verify correctness of solution
- Combination lock
- Use of a nonce

PoW

$$H_a(d+n) < h \quad (4)$$

where H is hashing function; a is hashing algorithm (e.g. SHA256); $d$is data; n is nonce; and h is a result of a hashing function usually starting with 4 zeroes.

PoW

- Waste of Energy
- Resource intensive
- Application-Specific integrated circuit - ASIC
  - 1kH/s - 1,000 hashes per second
  - 1MH/s - 1,000,000 hashes per second
  - 1GH/s - 1,000,000,000 hashes per second
  - ASIC chip around 30GH/s
- solving puzzle is difficult, checking the puzzle is easy
- Bitcoin rewards miners
- No reward?
  - Rely on transaction fees
  - Less miners and open to 51% attacks
  - Change in consensus algorithm?
- High latency of TX validation

PoW: Example

- $d = 0$
- $H(0)$ = 5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9
- target = **0**feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9
- $n = 1$
- while $(H(d+n) < target)$
  - n++
- $H(00x1)$ = 6fbc24c863cad03d71238d38f725383eb79804b1adf05b05511470f18ac66129
- $H(00x2)$ = 9eb14f1909e80b0005ea1531e91a315401e5f788e0c5e7f1b7c24f3d2c92e5a4
- $H(00x3)$ = 5e847f40960c2fe8fcaf2bf7b11df0cc012f73c59d52cd2ee8f5ee44b2711e85
- ⋮
- $H(00x48)$ = 0529f9d44d1ec54ce86601d63aac3a094ac90577b175e024058190a6ec062873
- target= **000**ceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9
- $H(00x80)$ = 00021397ccc9e4e75258c17ac7d651674999ea72c6d3f6dfdae55ca8a2174420

Proof of Stake, PoS

- Nodes are validators, not miners
- Validate a TX, to earn TX fee
- Each node has a stake value
- Usually, stake cannot be spent
- Nodes are selected proportionately to the stake value
- Randomness where stakes are equal
- Example:
  - Node A has 200 MDXCoins
  - Node B has 100 MDXCoins
  - Node A is twice as likely to be selected to validate the TX
  - Upon doing so Node A receives the transaction fee
- Many variations on this, Proof-of-Deposit

PoS

Random Selection

- Ratio between stake:all cryptocurrency
- 1% stake of the entire blockchain results in being selected 1% of the time
- 51% stake results in 51% selection

Multi-round Voting

- Byzantine Fault Tolerance PoS [1]
- Select several staked nodes
- Staked users cast a vote
- Elected creates block

Coin Age

- Older stakes are more likely to get selected than younger stakes
- Age is reset after selection
- Fatigue

Delegate Systems

- users vote for nodes to become publishing nodes
- voting power is proportionate to stake
- incentivised to not act maliciously
- rewards and reputation

Coin-Age

- cryptocurrency a coin may have a 28 day max age
- Proof-of-stake
- Stakes with older coins have higher probability of being selected
- Reset
- Larger stake, plus older coins increases probability of being selected
- Hoard older coins?

Coin-Age

- cryptocurrency a coin may have a 28 day max age
- Proof-of-stake
- Stakes with older coins have higher probability of being selected
- Reset
- Larger stake, plus older coins increases probability of being selected
- Hoard older coins?
- Built-in max probability of being selected.

Proof-of-Stake, PoS

- Less energy spent
- No miners
- Does mean bigger stakes, have more probability of being selected
- low latency of TX validation
- Speeds up block creation

Proof-of-Elapsed-Time, PoET

- All nodes are validators
- Random allocation of wait time
- The node with the shortest wait times validates the TX
- Permissioned blockchain
- Low latency of TX validation
- Speeds up block creation [7]
- Does depend on size of block and data in transaction
- Scalability is still an issue (1K transactions per second)

Consensus

trust and resource relationship

increase level of trust $\propto$ decrease in resource intensive algorithm

Problem Definition: 12%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
Problem Definition, PD (12%)	Specification	No Spec.	Spec., present	Spec. is not conducive to BC	Unrelated or missing spec. components	Spec. conducive to BC, all components explained and coherent	1	/4
	Flowchart, FC	No use of FC in [8]	FC applied, no explanation.	All components of FC applied, some explanation.	All components of FC applied correctly but does not match spec/UCD.	All components of FC applied correctly and matches spec/UCD	1	/4
	Use Case Diagram, UCD.	No UCD	Incoherent UCD	Misaligned UCD and PD. Assumptions left uncommented	No include or extend relationships. Assumptions commented	Aligned and complete UCD with comments and assumptions	1	/4

Data Modelling: 16%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
Data Model (16%)	Participants	No participants	Lacking and/or incorrect participants. Incorrect data types used. Unidentified.	Irrelevant participants. Correct identification. Lacking any assumptions. Opportunities to use more appropriate data types missed.	Participants lacking UCs and incomplete assumptions. Structurally sound.	Correct participants, data structures, assumptions and matching UCs	1	/4
	Assets	No assets	Lacking and/or incorrect assets	Irrelevant assets. No enum or concepts.	Assets unrelated to participants or no assets with the capability of state change	Some of the assets must at least be 3 of the following: have a state capable of change, relevant, complete and related to participants	1	/4
	Transactions, TX	No TX	Vague TX	TX not updating state	TX without ownership	participant specific TX	1	/4
	Comments	No comments	Auto-generated comments only (headers only), no clarifying comments	Vague, incorrectly placed and/or unexplanatory comments	Explanatory and identifiable comments, but incomplete. Too verbose and high comment to code ratio	Complete, concise and succinct comments	1	/4

Access Control Language: 12%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
Access Control Language, ACL, (12%)	Participants	No ACL. Basic ACL, admin access only & automatically generated code	ACL has too few rules	ACL has contradictions or allows unauthorised access to transactions or assets. There is no difference between participant access	ACL order is incorrect	ACL is implemented correctly	1	/4
	Ordering, Comments and listing	No listing or basic ACL, admin access only & automatically generated code	Syntax errors for ACL.	Rules are disorganised and need re-ordering. Inclusion of commented out rules	Rules are in correct order, but lack ideal names, descriptor values and comments. No line numbers.	Correct order and appropriate names, descriptors values and comments	1	/4
	Conditions	Auto-generated rules only. Admin access to all.	No conditions and simple rules only	Conditions applied incorrectly.	Identifier conditions applied correctly	Conditions to check status or lists and of a higher order of difficulty	1	/4

Business Logic: 32%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
Business Logic (32%)	Queries	No Queries	Queries but don't execute	Irrelevant Queries	Relevant Queries without relationships	Relevant Queries with relationships	1	/4
	Transactions	No Transactions	BL - run time execution	BL code accessing assets and participants, with no restriction, or comments directing to ACL	BL code accessing TX with restrictions, but not acknowledged	Acknowledged rules and restrictions and code accessing both assets and participants correctly	2	/8
	API	No use of promises	BL code not executing	BL code duplicating ACL	No extensive use of API and promises	Extensive use of API and Promises and complexity used to aid the update of state correctly	3	/12
	Initialise	No initialisation or automatic population of values in registry	Initialisation present but not working	Initialisation only partial, e.g., only completes assets and not participants	All assets and participants populated but incorrectly, e.g., data is misaligned	All assets and participants populated correctly	1	/4
	Comments	No comments	Non-explanatory comments	Partial explanatory comments	Overly commented	Fully explanatory comments	1	/4

Presentation: 12%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
Presentation (12%)	Slide Content	No slides	Incoherent presentation and not demonstrating the understanding of the coursework. Cluttered and/or illegible slide content	Coherent but poor content coverage. Less than 5 mins in length. Uncluttered. Some Illegible slide content, especially screenshots	Less than 9 mins or greater than 10 mins. Clear figures and screenshots. Coherent but not explaining all points required	Between 9-10 mins in length, clear and readable slides and addresses all items	1	/4
	Transaction, TX	No Demonstration	Demonstration of successful TX	Demonstration of unsuccessful TX due to ACL	Demonstration of unsuccessful Demonstration due to BL	All demonstrations completed	1	/4
	Structure	No structure	No headers *and* footers, slide numbers	No headers *or* footers, slide numbers	Headers, Footers and numbers but incorrect	All slides consistent with correct information in headers and footers.	1	/4

Documentation: 8%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
Report (8%)	English	Many sentences rendered nonsensical and many misspellings	Some sentences rendered nonsensical and a few misspellings	Sentences with poor grammar, written in first or second person, and a few misspellings	Good grammar, not written in third person. A few grammatical and spelling mistakes.	Written in third person. A few grammatical or spelling mistakes	1	/4
	Template	No structure followed	No numbering but structure present	Incorrect frontmatter or backmatter, but main matter correct structure. No figure, listing or table captions.	No citations or references, or incorrect bibliography style applied	Correct template, citations/references, numbering and template compliance.	1	/4

Business Network Archive: 8%

Criteria	Sub-criteria	0	1	2	3	4	W	Σ
BNA (8%)	Execution	Errors	Run-time errors	No errors (4)			1	/4
	BNA format	None	ACL	Node.js	CTO	Structure	1	/4

Summary

Criteria	PoW	PoS	Hybrid PoW/S	PoET
Efficiency	No	Yes	No	Yes
H/w	Very Important	None	Important	None
Speed	Poor	Good	Poor	Good
Example	BitCoin	NextCoin	BlackCoin	HyperLedger

References I

[1] Jean-Paul Bahsoun, Rachid Guerraoui, and Ali Shoker. "Making BFT protocols really adaptive". In: *2015 IEEE International Parallel and Distributed Processing Symposium*. IEEE. 2015, pp. 904–913.

[2] Miguel Castro, Barbara Liskov, et al. "Practical Byzantine Fault Tolerance". In: *OSDI*. Vol. 99. 1999, pp. 173–186.

[3] Sawtooth Hylperledger. *PBFT consensus - a pending Sawtooth RFC*. https://github.com/hyperledger/sawtooth-rfcs. Version 0.1. [accessed: 04-AUG-19].

[4] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine generals problem". In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3 (1982), pp. 382–401.

[5] Giang-Truong Nguyen and Kyungbaek Kim. "A Survey about Consensus Algorithms Used in Blockchain.". In: *Journal of Information processing systems* 14.1 (2018).

References II

[6] Marshall Pease, Robert Shostak, and Leslie Lamport. "Reaching agreement in the presence of faults". In: *Journal of the ACM (JACM)* 27.2 (1980), pp. 228–234.

[7] Parth Thakkar, Senthil Nathan, and Balaji Vishwanathan. "Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform". In: *arXiv preprint arXiv:1805.11390* (2018).

[8] Dylan Yaga et al. *Blockchain technology overview*. Tech. rep. National Institute of Standards and Technology, 2018.