

Algoritmos de consenso



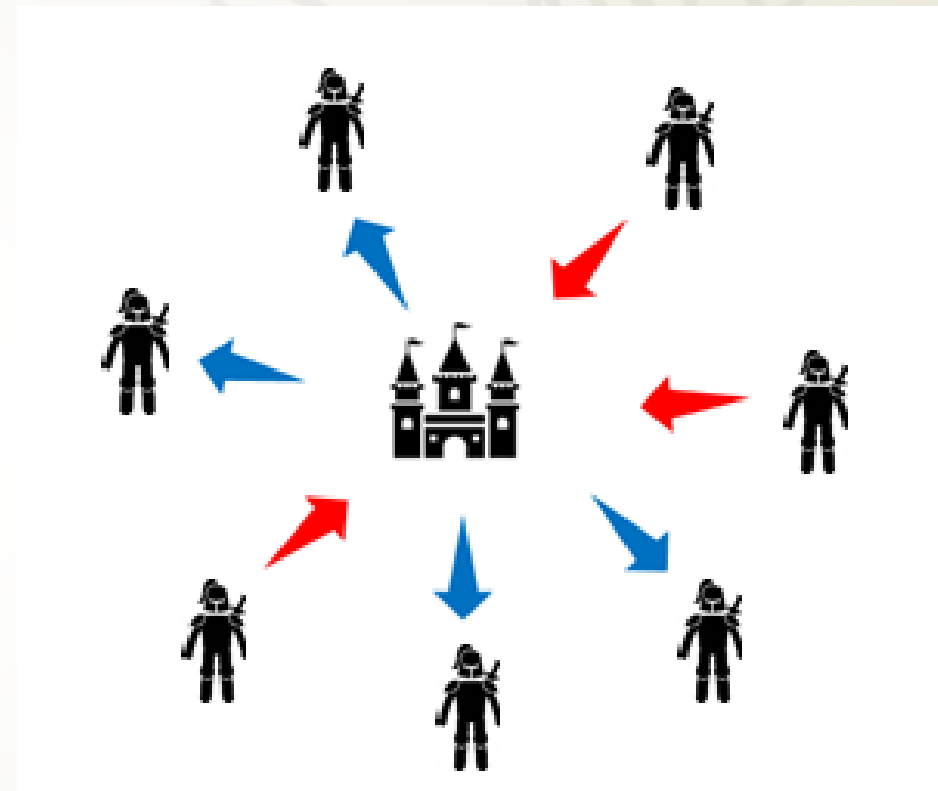
PUC Minas

Aplicações Descentralizadas e Blockchain

Prof. Carlos Leonardo dos S. Mendes

O problema dos generais Bizantinos

- ▶ Uma cidade está cercada pelas tropas do império bizantino.
- ▶ Pelotões comandados por generais estão em diferentes pontos e, devido ao difícil relevo, a comunicação entre eles se dá por mensageiros.
- ▶ O ataque só terá sucesso se coordenado (a maioria dos pelotões precisa atacar juntos).
- ▶ O mensageiro precisa atravessar a cidade para enviar a mensagem de ataque.



L. Lamport; R. Shostak; M. Pease (1982). "The Byzantine Generals Problem". *ACM Transactions on Programming Languages and Systems*. 4 (1): 382–401 [Aqui](#)

O problema dos generais bizantinos

- ▶ O problema reduzido a dois generais.
- ▶ Esse foi o primeiro problema de comunicação de computador considerado ser **insolúvel**.



Como chegar ao consenso?

Falha bizantina

“A falha bizantina é uma condição presente em sistemas distribuídos quando um ou mais componentes falham e não há informações precisas sobre a falha ou se o sistema opera de forma correta.”


- O termo **falha bizantina** vem do conhecido ***problema dos generais bizantinos***.
- Em uma falha bizantina, um componente do sistema pode se apresentar de forma inconsistente, apresentando sintomas diferentes diante de diferentes observadores.
- As falhas bizantinas são complexas de serem detectadas e resolvidas por sistemas de detecção de falhas, porque um componente pode estar gerando dados arbitrários, mas se apresentando como correto.

A tolerância a falhas bizantinas


- ▶ Exemplos de falhas bizantinas:
 - Discovery Spache Shuttle, voo STS-124
- ▶ Exemplos de sistemas tolerantes a falhas bizantinas:
 - Sistema de controle de voo do Boeing 777 e 787.
 - Sistema de controle de voo da cápsula SpaceX Dragon.

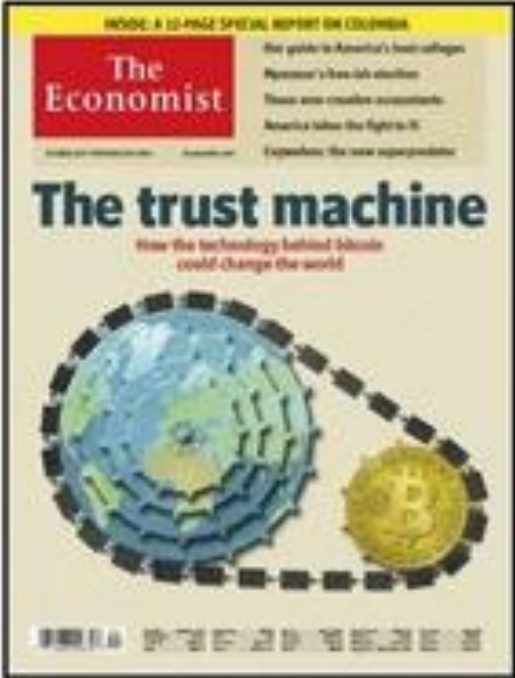


Algoritmos de consenso






Decentralized Consensus






$$r = f(w^m)$$

-  **r**esistance to new ideas
-  **n**umber of employees
-  **m**anagement levels

Nov 17th 2016

@CaisseDesDepots — @pdewost



Algoritmos de consenso

- ▶ Algoritmos de consenso são elementos críticos em uma rede de blockchain responsáveis por estabelecer um acordo entre os nós sobre o estado corrente da rede.
- ▶ Principais categorias de algoritmos:
 - **PoW (Proof of Work)**
 - **PoS (Proof of Stake)**
 - DPoS (Delegated Proof of Stake)
 - PoH (Proof of History)
 - PoA (Proof of Authority)
 - PoB (Proof of Burn)
 - PBFT (Practical Byzantine Fault-Tolerance)
 - PoC (Proof of Capacity)
 - PoET (Proof of Elapsed Time)



Algoritmo de prova de trabalho (PoW)

- O **algoritmo de prova de trabalho** ainda é um dos mecanismos de consenso mais utilizado em protocolos de blockchain.
- A ideia surgiu em 1992 e consiste em requerer uma **prova de resolução de uma função computacional** antes de um usuário enviar emails, permitindo aos destinatários mitigar *spams*.
- A resolução da função computacional teria pouco custo para um remetente legítimo, mas um alto custo para um envio em massa (*spam*).
- Uma das primeiras implementações é o algoritmo HashCash proposto por Adam Back em 1997.
- Nas plataformas de blockchain, o **esforço computacional** dos algoritmos de prova de trabalho usados é **considerável**.

Algoritmo de prova de trabalho (PoW)

Vantagens

- Um algoritmo PoW é muito seguro contra ataques, pois requer enorme energia computacional.
- É mais racional usar a energia computacional para minerar do que para atacar a rede (teoria dos jogos).
- O algoritmo não favorece quem tem mais criptomoedas, evitando concentração da rede nos mais “ricos”.

Desvantagens

- O PoW requer hardware altamente especializado com gastos energéticos enorme.
- Os cálculos realizados são “inúteis” no sentido que não podem ser usados em nenhuma outra funcionalidade.
- Embora seja muito seguro contra ataques, ainda está sujeito ao ataque de 51%.

Algoritmo de prova de participação (PoS)

- O **algoritmo de prova de participação** usa um processo de **eleição pseudo-randômico** para selecionar o **nó validador** que irá gerar o próximo bloco.
- No algoritmo PoS, os usuários que desejarem participar do processo de criação de blocos precisam **imobilizar** uma certa quantidade de **criptomoeda** na rede.
- Quanto **maior o valor imobilizado, maior a chance** de se tornar o validador do próximo bloco.
- Para não favorecer apenas os nós “mais ricos”, alguns outros fatores também são usados no processo de seleção: **randomização e tempo de imobilização**.
- É comum usar o termo “forjado” para os blocos criados com o algoritmos que não são PoW, ao invés do termo “minerado”.

Comparativo entre os principais algoritmos de consenso

Row	Consensus algorithms	Cryptocurrencies	Algorithm	Genesis Block	Rank	Market CAP (\$)	TPS	Block Time Minutes	Mining reward
1	PoW	Bitcoin	SHA256	January 3, 2009	1	180,207,092,238	7	10	12.5 BTC
		Ethereum	Ethash (KECCAK256)	July 30, 2015	2	22,757,000,420	15	0.25	2
		Litecoin	Scrypt	October 8, 2011	5	4,587,952,794	28	2.3	25
		Monero	Cryptonight	April 18, 2014	11	1,268,871,523	30	2	4.9
		Zcash	Equihash	October 28, 2016	28	348,443,197	27	2	10
2	PoS	Waves (LPoS)	LPoS	June 12, 2016	55	100,304,755	100	1	Non-mineable
		Qtum	POS 3.0	December 26, 2016	36	202,601,750	70	2	Non-mineable
		Nxt	SHA256	November 24, 2013	175	16,162,355	100	1	Non-mineable
		Blackcoin	Scrypt	February 24, 2014	500	4,569,548	0	1	Non-mineable
		Nano	Blake2b	February 29, 2016	45	123,741,646	7000	Instant	Non-mineable
3	DPoS	EOS	DPoS	July 1, 2017	7	3,641,735,649	4000	0.5	Non-mineable
		Cardano	Ouroboros (DPoS)	December 26, 2017	12	1,266,573,741	257	0.33	Non-mineable
		TRON	DPoS	August 28, 2017	13	1,186,299,015	2000	0.05	32 TRON
		Lisk	DPoS	January 30, 2016	47	118,714,644	3	0.284	Non-mineable
		BitShares	DPoS	July 19, 2014	58	91,575,735	100000	0.05	Non-mineable
4	PBFT	Ripple	N/A	April 11, 2013	3	12,010,477,031	1500	0.06	Non-mineable
		Stellar	N/A	April 6, 2016	10	1,410,189,643	1000	0.08	Non-mineable
		Zilliqa	Keccak	January 12, 2018	79	59,022,911	0	45s to 4 m	Non-mineable
5	PoC	Burst	Shabal256	August 11, 2014	190	14,417,212	80	4	460
6	DAG	IOTA	Curl-P	October 21, 2015	17	788,711,735	1000	Instant	Non-mineable
		Byteball (Obyte)	DAG	September 5, 2016	262	17,301,594	10	0.5	Non-mineable
		Travelflex	DAG	December 2, 2017	1374	163,648	3500	1	30.00 TRF
7	PoA (Hybrid PoW/PoS)	Dash	X11	January 19, 2014	16	850,165,302	56	2.5	2.09
		Decred	BLAKE256	December 15, 2015	32	233,089,579	14	5	18.22
		Komodo	Equihash	September 1, 2016	67	80,699,867	100	1	3.00 KMD
		Peercoin	SHA-256	August 19, 2012	373	7,844,163	0	10	37.36 PPC
		Espers	HMQ1725	April 28, 2016	1026	625,199	0	5	5000
8	dBFT	NEO	RIPEMD160	October 17, 2016	20	650,866,809	1000	0.25	Non-mineable
9	Pol	NEM (XEM)	Ed25519	March 31, 2015	26	403,570,701	10000	1	Non-mineable
10	PoB	Slimcoin	Dcrypt	May 07 2014	2661	16,195	0.00003	1.5	50.00 SLM

Novembro de 2019



PUC Minas



PUC Minas