PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Marcos Antônio Lommez Cândido Ribeiro
Saulo de Moura Zandona Freitas
Bernardo Marques Fernandes
Eric Miranda Ferreira Guimarães

Nether

Descrição de arquitetura da blockchain

1 Objetivo

A blockchain será utilizada para salvar dados de um sistema de segurança distribuído, com o objetivo de mantê-los invioláveis e garantir sua integridade. O foco principal é garantir uma arquitetura segura e com tolerância a falhas.

2 Modelo de Consenso Híbrido

A blockchain usará um modelo híbrido de consenso que combina **Proof of Work (PoW)** e **Proof of Authority (PoA)**, garantindo segurança e mitigando problemas de eficiência de modelos PoW puros assim como a liberdade de mudança de lideres que o PoA não permite facilmente.

2.1 Nós Líderes

- Serão eleitos via PoW.
- Permanecerão como líderes por um tempo relativamente longo (até serem desligados ou até atingir um tempo X, quando novas eleições serão realizadas).
- Manterão salvos todos os dados históricos da blockchain durante o período em que forem líderes.
- Formarão uma topologia de **malha parcial**, em que os líderes têm conexões diretas uns com os outros.

2.2 Nós Subordinados

- Se conectarão a um único nó líder.
- Manterão um array contendo dados de outros líderes, mas sem a obrigação de manter uma cópia completa da blockchain. Eles terão apenas a parte que lhes é relevante e o último nó atualizado.

2.3 Topologia Geral

- Como dito lideres estarão organizados em uma topologia de **malha parcial**.
- Quando considerados como uma unidade, os nós líderes formarão uma topologia de **estrela**, conectando-se aos subordinados.
- Os líderes manterão uma lista de líderes e subordinados, com as conexões feitas via **P2P**.

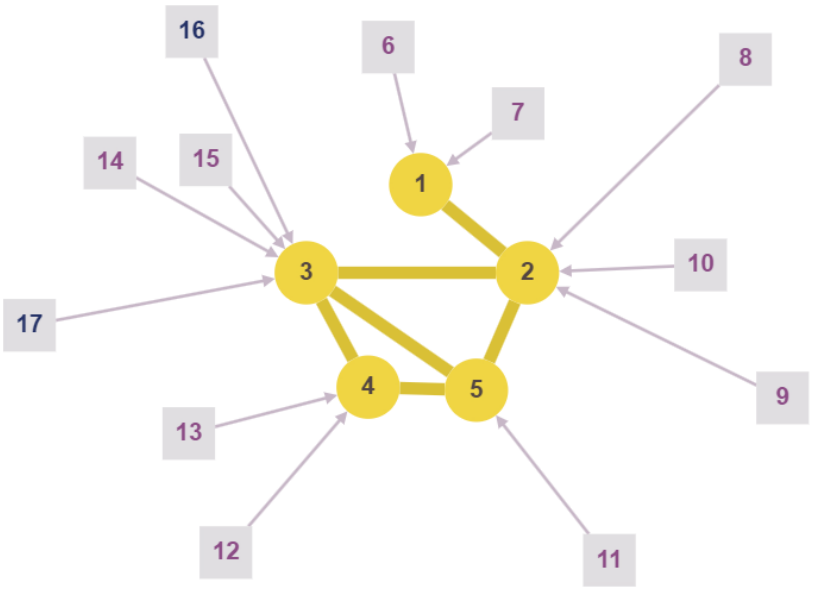

Figura 1: Topologia do Nether
Nós lideres em amarelo e nós subordinados em cinza

2.4 Eleições dos Nós Líderes

- As eleições serão iniciadas pelos líderes atuais, e os nós interessados poderão se candidatar avaliando seu próprio poder computacional.
- O primeiro nó a responder será eleito como líder.
- Para evitar condições de corrida, o primeiro líder a responder notificará os outros sobre o encerramento da eleição. Em casos de empate, a decisão será tomada de forma arbitrária.

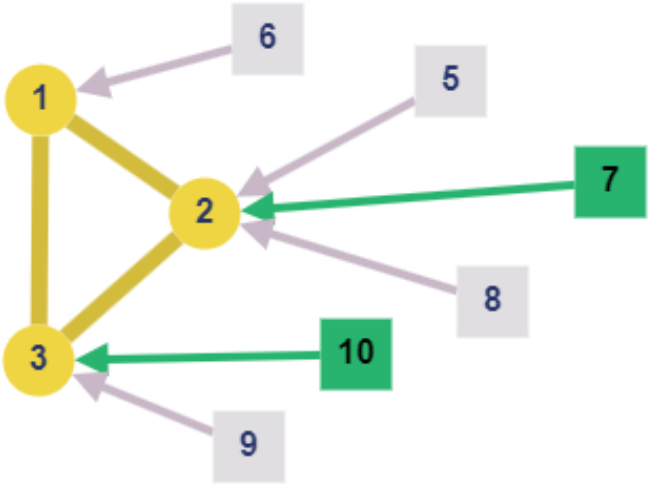

Figura 2: Processo de eleição, candidatos em verde

3 Segurança e Criptografia

3.1 Assinatura Digital e Validação

- Cada bloco será assinado digitalmente pelo líder que o adicionar à blockchain, utilizando o algoritmo **ECDSA (Elliptic Curve Digital Signature Algorithm)**.
- Cada nó tera uma **chave privada** escolhida durante o registro, a sua **chave pública** resultante sera seu próprio nome na rede e **ID**.
- Quando um dado for inserido na blockchain, o líder responsável assinará o bloco digitalmente mantendo sua chave publica no bloco. Os outros líderes validarão essa assinatura antes de adicionar o bloco. Em caso de invalidação o líder sera automaticamente desconectado.

3.2 Criptografia de Blocos

- A blockchain usará **AES-256** para o hash dos blocos. Onde cada bloco sera criptografado utilizando a hash do bloco anterior como chave.

4 Conexão de Nós na Blockchain

4.1 Primeira Inicialização de uma Nova Blockchain

- Um nó pode inicializar diretamente como líder para criar uma nova blockchain no primeiro momento de operação da rede.
- Uma rede devera possuir um nome/id único, para evitar desastres em caso de um nó de uma rede se conectar ao nó de outra blockchain compatível. Logo toda conexão inicial deve verificar a blockchain alvo.

4.2 Primeira Conexão de um Nó

- Um nó, ao se conectar pela primeira vez, precisará fornecer manualmente o endereço de outro dispositivo da rede.
- Será utilizada uma conexão **TURN/STUN** para realizar a descoberta do endereço.
- Um nó manterá uma lista de nós em stand by, que estarão aguardando conexão oficial na rede. Da mesma forma nós em stand by também podem possuir uma

lista de outros nós em seu aguardo, a inserção sera realizada em cadeia caso isso aconteça (Shutdown podem levar a estes casos).

- O nó conectado enviará uma lista de líderes ao novo nó, que se conectará ao líder com o menor valor de [**tempo de resposta + carga de trabalho**].

4.3 Lista de Seeds

- Após a primeira conexão, o nó manterá em memoria primaria e secundaria uma lista de **seeds** para reconexão, evitando a necessidade de repetir o processo manual.
- Caso a conexão com o líder seja perdida, basta se conectar a outro usando o mesmo critério.

4.4 Processo de Recuperação e Reconciliação

- Para garantir a resiliência da rede, cada nó manterá sempre o ultimo bloco valido pois o mesmo possui o dado do ultimo lider, alem de manter em memoria secundaria sua lista de peers e hierarquia, assim possibilitando a recuperação da blockchain em caso de *shutdown* parcial ou completo.
- Após um *shutdown*, um nó tentará se conectar inicialmente ao líder salvo no ultimo bloco inserido de sua copia da base de dados.
- Em caso de *shutdown* de um lider, o mesmo ao reconectar perdera o status de líder.
- Caso um lider em reconexão não encontre outros lideres no processo de recuperação a rede ira permanecer em status de espera, caso apos um tempo um líder não seja encontrado, sera realizado um consenso para retomada dos lideres anteriores ativos.
- Após recuperação da conexão Será iniciado um protocolo de **reconciliação de dados incoerentes**, garantindo que os dados perdidos ou divergentes sejam resolvidos.
- Se durante o processo de reconexão de um nó for detectado a existência de dados que não estão presentes nos lideres a reconciliação ocorrerá da seguinte forma:
  - A rede de maior tamanho será mantida.
  - Os nós "perdidos"serão analisados e reincorporados à rede maior.

5 Comunicação na rede

5.1 Conexão

- A conexão sera mantida através de um canal TCP peer-to-peer.
- Pings regulares serão realizados para testar a conexão
- Algum algoritmo especifico e mais robusto deve ser escolhido para manter a conexão dos nós lideres.

5.2 Comunicação Entre Nós Líderes

- A comunicação entre nós líderes utilizará o algoritmo de **Dijkstra** para encontrar o caminho mais curto na rede, prevenindo ataques de *man-in-the-middle*.
- A topologia será **totalmente conectada** dentro da mesma rede e **tunelada** ao se comunicar com redes externas.
- Em caso de desconexão de um líder, a malha será reconstruída usando a lista de nós previamente conhecida.

5.3 Comunicação Entre Líderes e Subordinados

- Lideres irão enviar broadcast para seus respectivos subordinados sempre que for necessário atualizar, contendo o novo bloco a ser adicionado. O mesmo deve ser validado pela chave, caso incoerência seja encontrada o nó deve desconectar e procurar outro para continuar.

6 Prova de Trabalho (Proof of Work)

6.1 Validação de Nós Líderes via PoW

- O desafio do **Proof of Work** será realizado fora da blockchain, em um programa modular plugável, permitindo personalização pela instituição que a estiver utilizando.
- O desafio será identificado por um **ID** derivado do **hash** associado ao programa do desafio.
- Consequentemente apenas "autoridades"terão conhecimento sobre o processo de eleição.

- O programa de desafio receberá como argumento padrão o número criptográfico do último bloco para utilizar de seed caso necessário.

6.2 Computação Paralela

- O paralelismo será aplicado no desafio do Proof of Work. Por ser um programa modular podemos facilmente fazer diferentes implementações de um mesmo problema, seja sequencial, paralela em thread, paralela em green thread ou por GPU, e até mesmo em diferentes linguagens.

7 Considerações Adicionais

- Precisa ser definido se a conexão sera TURN ou STUN.
- O algoritmo distribuído para a rede de lideres deve ser escolhido, talvez nem seja necessario.
- O tipo de desafio do **Proof of Work** precisa ser definido.
- É necessário simular uma rede local para rodar a blockchain em vários PCs simultaneamente.
- A blockchain armazenará **imagens** como parte dos dados, mas a arquitetura será independente dos tipos de dados inseridos.