Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Lucas Stefan Abe

Arquitetura e escalabilidade das criptomoedas baseadas em blockchain

São Paulo
Novembro de 2018

Arquitetura e escalabilidade das criptomoedas baseadas em blockchain

Monografia final da disciplina
MAC0499 – Trabalho de formatura supervisionado.

Supervisor: Prof. Dr. Daniel Macedo Batista

São Paulo
Novembro de 2018

RESUMO

A primeira moeda digital totalmente descentralizada a ser criada foi o Bitcoin. Sua descentralização é atingida através de um sistema que funciona em uma rede *peer-to-peer* sem autoridade central, criptografia (daí o nome criptomoeda) e blockchain, um registro digital distribuído que armazena todas as transações que já ocorreram de forma praticamente imutável. Atualmente existem várias criptomoedas que se baseiam em blockchain, sendo que algumas delas funcionam não só como moeda mas também como uma plataforma de computação distribuída, como o Ethereum. Porém, um assunto que é muito discutido atualmente no meio é a escalabilidade. Se por um lado essas criptomoedas são descentralizadas, existe o problema de que elas só conseguem realizar uma quantidade baixa de transações por segundo, o que dificulta sua adoção em massa. O objetivo deste trabalho é estudar a arquitetura de criptomoedas como Bitcoin e Ethereum e fazer uma análise sobre possíveis soluções para a escalabilidade.

Palavras-chave: criptomoeda, blockchain, Bitcoin, Ethereum, escalabilidade, arquitetura, moeda digital, *peer-to-peer.*

ABSTRACT

The first fully decentralized digital currency to be created was Bitcoin. Its decentralization is achieved through a system that operates on a peer-to-peer network without central authority, encryption (hence the name criptocurrency) and blockchain, a distributed digital record that stores all transactions that have already occurred in a virtually immutable way. Currently there are several cryptocurrencies that are based on blockchain, some of which work not only as currency but also as a distributed computing platform, such as Ethereum. However, one subject that is widely discussed is scalability. Although these cryptocurrencies are decentralized, there is the problem that they are able to perform only a small amount of transactions per second, which hampers their mass adoption. The objective of this work is to study the architecture of cryptocurrencies such as Bitcoin and Ethereum and to analyze possible solutions for scalability

Keywords: cryptocurrency, blockchain, Bitcoin, Ethereum, scalability, architecture, digital currency, peer-to-peer.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização e motivação

O Bitcoin foi apresentado por um grupo ou pessoa usando o pseudônimo Satoshi Nakamoto através do artigo "*Bitcoin: A Peer-to-Peer Electronic Cash System*" (NAKAMOTO, 2008) e é tido por muitos como uma grande revolução tecnológica e econômica. Isso se deve ao fato de que o Bitcoin foi a primeira moeda digital a ser criada e que não é emitida por qualquer instituição financeira, seu controle é descentralizado através de uma rede *peer-to-peer* e também é seguro (até hoje não houve nenhum ataque à rede com sucesso) graças à criptografia e ao blockchain, uma estrutura criada inicialmente para o Bitcoin que permite o armazenamento de um registro público de todas as transações que já ocorreram de forma praticamente imutável através de um consenso entre participantes da rede.

Após a criação do Bitcoin, muitas outras criptomoedas apareceram, e com isso surgiram novas ideias de aplicações para o blockchain. Podemos citar o NameCoin como uma delas, que funciona como um banco de dados descentralizado para registrar nomes de domínios. Diante a criação de diversas criptomoedas para aplicações específicas surge o Ethereum, que funciona como uma plataforma de computação descentralizada, onde pode-se se criar os chamados *DAPPS* (aplicações descentralizadas) utilizando uma linguagem de programação turing completa para criar scripts que são chamados de contratos inteligentes.

Com a recente popularização das criptomoedas, devido principalmente à valorização econômica astronômica das mesmas, surge o problema da lentidão e altas taxas para realizar transações na rede (no caso do Bitcoin as transações chegaram a custar em média 55 dólares em dezembro de 2017) .

Isso se deve às limitações da arquitetura atual, o Bitcoin, por exemplo, tem um limite estimado de 7 transações por segundo, já o Ethereum tem um limite estimado de 20 transações por segundo.

Portanto, usar essas criptomoedas em micro transações ou para o desenvolvimento de aplicações descentralizadas, como é o caso do Ethereum, atualmente é inviável, visto que sistemas centralizados como o Visa processam em

média 2 mil transações por segundo, tendo capacidade para realizar até 56 mil (Visa Inc, 2015).

Então surge a necessidade de tornar essas criptomoedas escaláveis, de tal forma que se mantenha suas propriedades desejadas, no caso descentralizadas e seguras. Atualmente há um grande debate no meio, sendo que muitas criptomoedas atuais são ditas ser capazes de processar uma quantidade enorme de transações por segundo, porém sacrificam pelo menos uma dessas duas propriedades. Existem algumas abordagens promissoras para resolver tal problema, sendo que para o Bitcoin já existe uma implementação em fase de teste e o Ethereum possui algumas em desenvolvimento.

1.2 Objetivos

Esse trabalho tem o objetivo de estudar a arquitetura das criptomoedas com foco no Bitcoin e Ethereum, e também fazer uma breve análise qualitativa das soluções para a escalabilidade que estão sendo propostas atualmente.

1.3 Organização do documento

Este documento está organizado da seguinte forma: o Capítulo 2 apresenta os principais conceitos e componentes do Bitcoin, que é considerado a primeira criptomoeda e serve como base para as outras criptomoedas. O Capítulo 3 explica a arquitetura do Ethereum, focando na sua maior inovação em relação ao Bitcoin, os contratos inteligentes. No Capítulo 4 são apresentadas as principais propostas de solução para a escalabilidade das criptomoedas. Por fim, o Capítulo 5 finaliza o texto com uma conclusão geral sobre o que foi desenvolvido no trabalho.

2. BITCOIN

Em 2008 foi publicado o whitepaper do Bitcoin por uma pessoa (ou, possivelmente, um grupo de pessoas) com o pseudônimo Satoshi Nakamoto, onde é apresentado pela primeira vez um modelo de moeda digital descentralizada que é descrito como “Uma versão puramente peer-to-peer de dinheiro eletrônico que permite pagamentos online serem enviados diretamente de uma parte para a outra sem passar por uma instituição financeira” (Nakamoto, 2008). Após essa publicação do Bitcoin, começa sua implementação por Satoshi e outros colaboradores em um software de código aberto com o nome Bitcoin Core. Em 2009 o sistema entra no ar e está ativo desde então.

De acordo com Satoshi, apesar de os sistemas de pagamento online baseados em instituições financeiras funcionarem relativamente bem, existe o problema de serem baseados em um modelo de confiança. Isso gera alguns problemas como disputas entre compradores e vendedores pois as transações não são irreversíveis, sendo necessário o controle sobre as informações pessoais de quem os usa, nem todo mundo consegue abrir uma conta com facilidade, transações para outros países podem demorar dias e também é comum casos de fraudes.

Em contrapartida, o Bitcoin utiliza um modelo baseado em provas criptográficas em vez de confiança (daí o nome criptomoeda). As transações no Bitcoin são armazenadas de tal forma que é probabilisticamente impossível de revertê-las após um certo tempo e qualquer pessoa pode criar uma conta e usá-la sem necessidade de fornecer qualquer dado pessoal. Como o sistema funciona em uma rede peer-to-peer onde não existe autoridade central, foi necessário criar uma estrutura para armazenar as transações e evitar o problema de gasto duplo de bitcoin, que é chamado de blockchain. O sistema é seguro desde que os participantes honestos da rede possuam maior poder computacional do que os não honestos.

2.1 Visão geral

2.1.1 Clientes e carteiras

Para um usuário se conectar diretamente à rede do Bitcoin é necessário possuir acesso à internet e um software cliente. Esse software cliente pode ser o Bitcoin Core ou outra implementação que segue as regras definidas do Bitcoin. Basicamente existem dois tipos de clientes:

- *Full client:* Também conhecido como *full node* (nó completo), mantêm uma cópia atualizada de todas as transações que já ocorreram, portanto é essencial para o funcionamento do Bitcoin. Antigamente todo usuário utilizava esse tipo de cliente, mas com o crescimento do tamanho do blockchain (atualmente ocupa cerca de 200 GB), é mais usado por mineradores, que usam esse tipo de cliente para validar e inserir novas transações na rede. (Exemplo: Bitcoin Core)
- *Light client:* Também conhecido como *light node* (nó leve), só realiza o download dos cabeçalhos do blockchain e algumas transações para realizar uma forma simples de verificação de pagamento. Utilizado por softwares de carteira por consumir pouco espaço e largura de banda. (Exemplo: Electrum Bitcoin Wallet)

Uma carteira de Bitcoin é um software utilizado por um usuário final, ela facilita o envio e recebimento de bitcoins e também cria, gerencia e armazena os endereços e as respectivas chaves privadas do usuário, que funcionam como contas.

O endereço é usado para identificar uma conta do Bitcoin, já a chave privada é usada pelo software de carteira para assinar as transações. Cada endereço é derivado de uma chave privada, e os softwares de carteira podem gerar inúmeras chaves privadas. Geralmente, os softwares de carteira criam essas chaves através da geração de frases mnemônicas aleatórias em inglês de 12 a 24 palavras. Dessa forma, o usuário pode fazer um *backup* de suas contas anotando essa frase gerada, sem a necessidade de anotar todas as chaves privadas.

A maioria dos clientes do Bitcoin possuem funcionalidade de carteira e existem carteiras de diversos tipos, sendo elas classificadas como: *Desktop*, *Mobile*, *Web* e *Hardware*. A carteira considerada mais segura é a carteira de Hardware, pois a chave privada nunca é exposta em um ambiente online.

2.1.1 Transferência de bitcoins

Suponhamos que Raphael possui 1 bitcoin e quer enviar 0.3 bitcoin para Anne. Do ponto de vista do Raphael é simples, ele abre seu software de carteira, cria uma nova transação com o endereço de Anne, a quantidade e taxa que deseja pagar, clica em enviar e aguardar a rede confirmar a transação.

Então a carteira assina a transação com a chave privada de Raphael e faz sua transmissão para os nós da rede que registram a transação (mineradores) no blockchain através de um processo chamado mineração. Esse processo consiste em resolver problemas computacionas para incluir um novo bloco de transações. Quando o bloco com a transação de Anne é incluido no blockchain, Anne recebe a primeira confirmação do recebimento de 0.3 bitcoin. A figura abaixo resume esse processo.

Fonte: Estevão P.

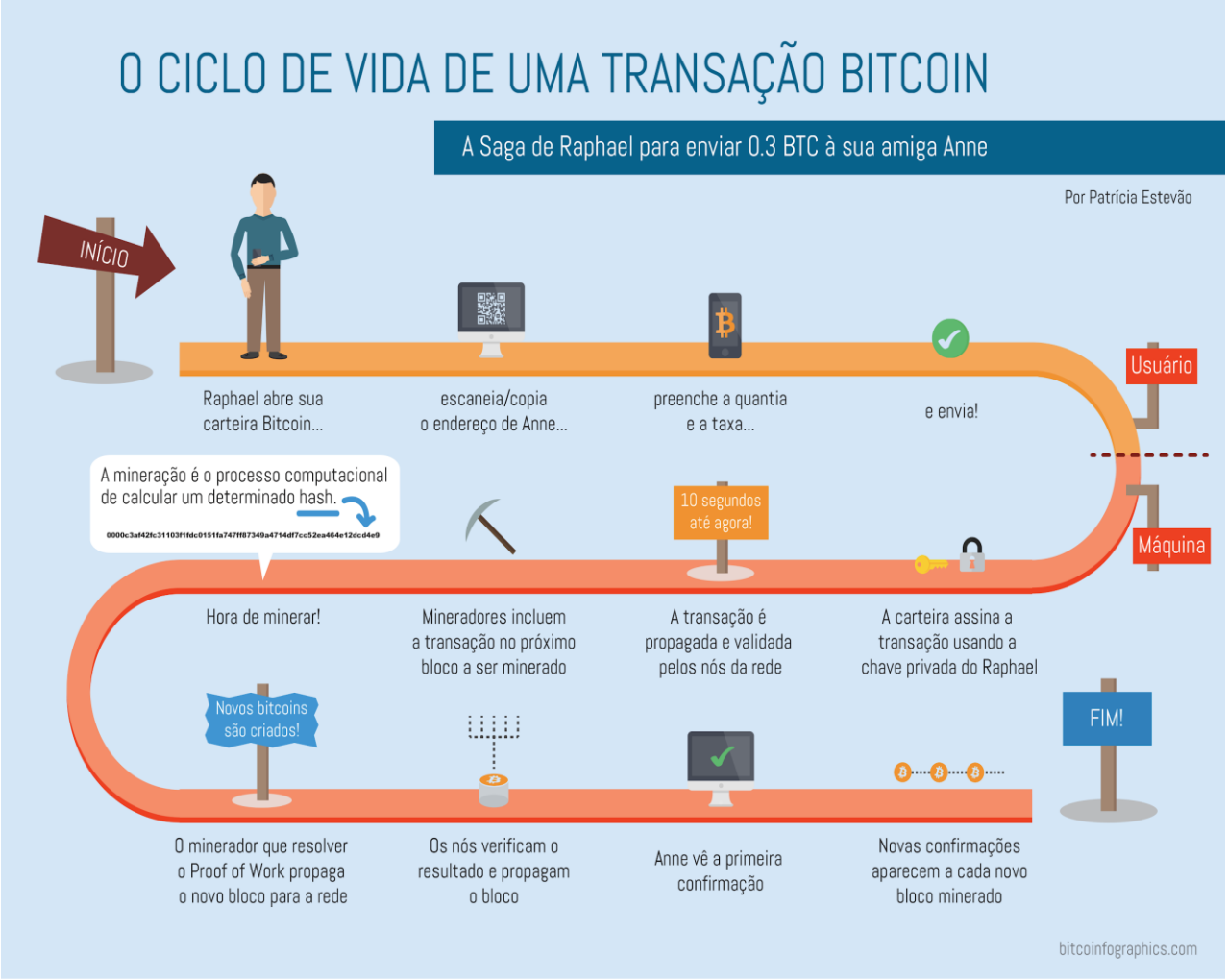

Figura 1 - O ciclo de vida de uma transação no Bitcoin

2.1.2 Como obter bitcoins

A forma mais comum de obter bitcoins é em sites especializados na compra e venda de criptomoedas, mais conhecidos como *exchanges*. Após a compra basta baixar algum software de carteira, criar uma conta e depositar no endereço da conta. Também há uma outra forma de obter bitcoins, através da mineração, porém é um processo muito complexo e custoso.

2.2 Hash criptográfico

Uma função de hash criptográfico é uma função de hash inviável de reverter. É um algoritmo matemático que mapeia dados de tamanho arbitrário (mensagem) para uma string de tamanho fixo (hash). Hashes criptográficos são muito utilizados em criptomoedas devido a quatro propriedades importantes:

- É determinístico, ou seja a mesma mensagem resulta no mesmo hash
- É inviável gerar uma mensagem através de seu valor de hash
- Uma pequena mudança na mensagem muda o valor de hash de tal forma que o novo valor parece não ter relação com seu valor antigo
- É inviável encontrar duas mensagens diferentes com o mesmo valor de hash

2.3 Criptografia de chave pública e endereços

As criptomoedas em geral baseiam-se no esquema de criptografia de chave pública ECSDA (*Elliptic Curve Digital Signature Algorithm*, em português, Algoritmo de Assinatura Digital de Curva Elíptica) para controlar o acesso das criptomoedas pelos usuários, ou seja, é um sistema de contas do Bitcoin.

Nesse esquema, existe uma chave pública e uma chave privada, sendo que a pública é derivada da chave privada através de uma função matemática irreversível, chamada de multiplicação de curva elíptica. Além disso, no Bitcoin, o endereço é criado aplicando funções de hashing criptográfico na chave pública. A relação entre chave privada, chave pública e endereço do Bitcoin é mostrada na Figura 2.

Fonte: ANTONOPOULOS A. M, 2017

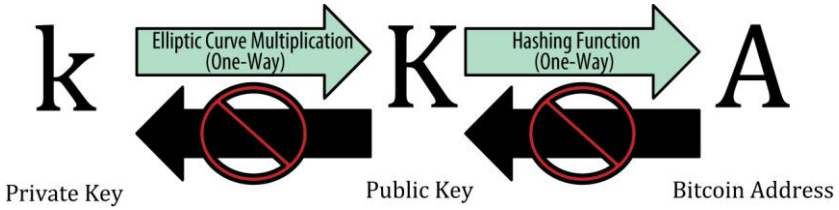

Figura 2 - Processo de derivação de chave pública e endereço, o processo reverso não é possível

Chave privada: Simplesmente um número de 32 bytes, que é gerado aleatoriamente nos softwares de carteiras, pois isso garante que outro usuário não terá a mesma chave privada, portanto segura[1]. A posse dessa chave é raiz para o controle de todos os fundos associados ao respectivo endereço do Bitcoin, portanto deve ser mantida em segredo. Ela é usada para realizar assinaturas digitais na transferência de bitcoins para um outro endereço.

Chave Pública: Um número de 65 bytes, que é usado para os nós da rede determinarem se a assinatura digital é legítima, sem a divulgação da chave privada, isso é possível devido a relação matemática entre chave pública e privada.

Endereço do Bitcoin: Usado para receber fundos, o endereço é derivado através da chave pública aplicando as funções de hash criptográfico SHA256 e RIPEMD160 e depois é aplicado uma codificação de string apenas com caracteres alfanuméricos (Figura 3).

[1] A quantidade de chaves privadas possíveis é 2^{256}, então um ataque por força bruta é probabilisticamente impossível se a senha for gerada de forma aleatória.

Fonte: ANTONOPOULOS A. M, 2017

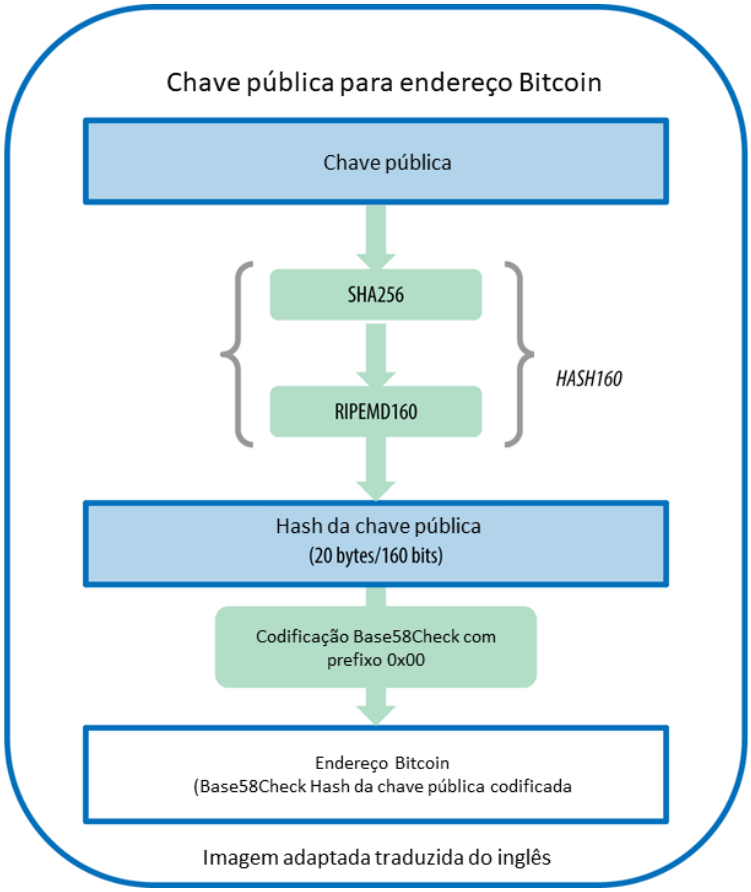

Figura 3 - Processo resumido de derivação de endereço através da chave pública

Exemplo de derivação par de chave pública-privada ECSDA e endereço:

Chave privada:

0x18e14a7b6a307f426a94f8114701e7c8e774e7f9a47e2c2035db29a206321725

Chave pública:

0x0250863ad64a87ae8a2fe83c1af1a8403cb53f53e486d8511dad8a04887e5b2352

Endereço do Bitcoin:

1PMycacnJaSqwwJqjawXBErnLsZ7RkXUAs

Observa-se que com a criptografia de chave pública é possível que qualquer pessoa crie uma conta de forma anônima e o usuário tem total controle dos seus fundos, o que tem suas vantagens e desvantagens em relação às instituições financeiras. A vantagem é a praticidade de se criar uma conta, a desvantagem é que no caso do usuário perder ou ter a sua chave privada roubada por terceiros, todos os

bitcoins serão perdidos sem chance de recuperação dos fundos. Os softwares de carteira geralmente geram diversas chaves privadas a partir de uma frase de 12 a 24 palavras, assim o usuário consegue fazer o *backup* de diversas chaves privadas somente com essa frase.

2.4 Unidades

A unidade de moeda do Bitcoin é o “bitcoin”. Abreviada como BTC, é divisível em até 8 casas decimais. Sua menor fração (10^{-8} bitcoin) é chamada de satoshi, em homenagem a seu criador. Sua grande divisibilidade é necessária pelo fato de que o Bitcoin tem um suprimento limitado de aproximadamente 21 milhões de bitcoins.

2.5 Transações

Todas as transações que ocorrem na rede são armazenadas no blockchain, validadas por mineradores (mais detalhes serão explicados na próxima seção) e também são visíveis para qualquer nó da rede. Cada transação possui um ou mais *inputs*, que representam a origem dos fundos e um ou mais *outputs* que são os destinos dos fundos.

Para autenticar a transação do usuário, ela é assinada digitalmente com a chave privada pela carteira. A assinatura e a chave pública também são anexadas no *input* (Figura 3).

Fonte : BARRERA, A.

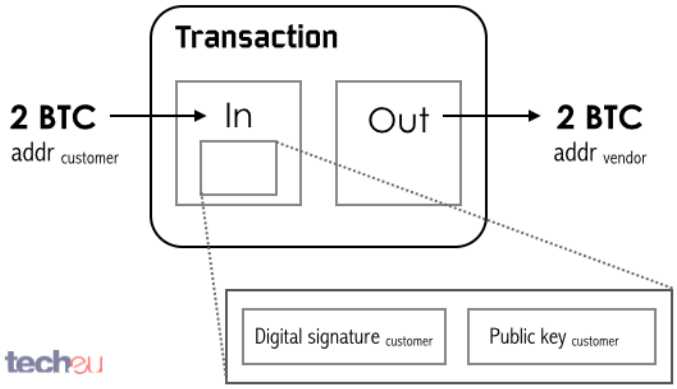

Figura 4 - Ilustração simplificada de uma transação de 2 BTC

As transações não existem individualmente, cada *input* de transação é ponteiro de um *output* de uma outra transação. Em outras palavras, o *input* usado em uma transação é o *output* de outra transação (Figura 4). Portanto, o blockchain armazena uma lista ligada de transações com origem na *coinbase transaction*[2].

Fonte: BARRERA, A.

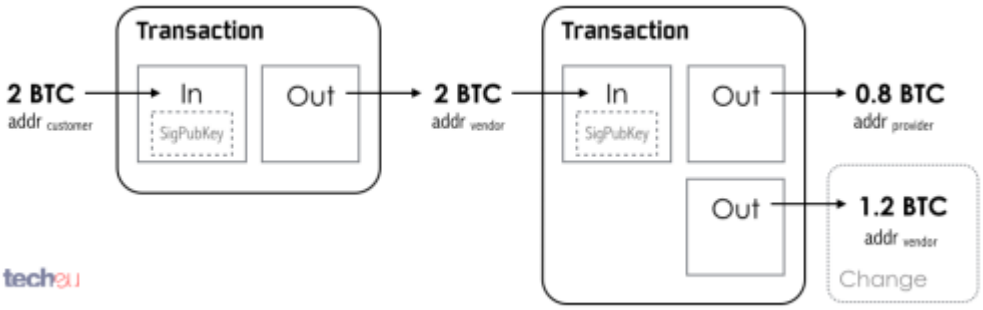

Figura 5 - Ilustração simplificada da ligação entre duas transações

Apenas o output de uma transação de bitcoins não gastos pode ser usado como *input* de outra transação, o nome desse tipo de output é UTXO (*Unspent Transaction Output*, em português, saída de transação não gasta). Além disso, os bitcoins dos *inputs* devem ser usados totalmente na transação. No caso do usuário não querer gastar todos os bitcoins de um *input*, o software de carteira adiciona um outro output com um endereço do usuário para retornar o "troco" (Figura 5). Também no caso de o usuário não ter bitcoins suficientes disponíveis em um endereço, a carteira utiliza vários endereços diferentes.

Fonte: BARRERA, A.

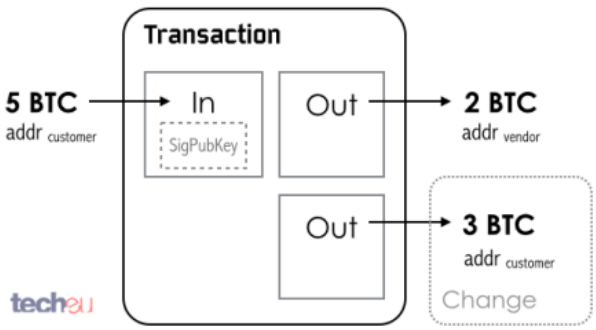

Figura 6 - Transação em que é usado um UTXO de 5 BTC com troco de 3 BTC

[2] Transação especial que só possui *output,* portanto emite novas unidades de bitcoin

O saldo de um endereço é a soma de todos os UTXO e a transação só é aceita se a soma dos bitcoins dos *outputs* for menor ou igual que a soma dos *inputs*. No caso de a soma dos inputs ser menor, os bitcoins restantes ficam como taxa para os mineradores.

2.6 Blockchain

O Blockchain é um registro público distribuído em vários nós pela internet de todas as transações do Bitcoin, que opera em uma rede peer-to-peer. O registro é formado por blocos, que armazenam conjuntos de transações em uma árvore de Merkle e outros dados importantes. Cada bloco é identificado por um hash gerado a partir de dados do seu cabeçalho e possui o hash do bloco anterior. Dessa forma, todo bloco tem uma referência para o bloco anterior, formando uma cadeia de blocos (Figura 6), daí o nome blockchain (blockchain veio da união das palavras em inglês “block chain”, que significam “cadeia de blocos”).

A inserção de novos blocos é feita por um processo chamado mineração, que consiste em resolver um problema computacionalmente difícil chamado Proof-of-Work (em português, prova de trabalho). Os nós da rede que realizam essa tarefa são chamados de mineradores, eles armazenam uma cópia completa do blockchain e recebem um incentivo econômico em forma de bitcoins quando inserem um novo bloco. Para alterar um bloco no blockchain é necessário alterar todos os blocos seguintes, e refazer todo o trabalho do Proof-Of-Work. Portanto, quanto mais blocos são adicionados após um bloco, menor a probabilidade de revertê-lo.

A criação dessa estrutura representa uma revolução na àrea de moedas digitais, pois ela resolve o problema de double spending (gasto duplo), que é o ato de gastar uma mesma moeda digital mais de uma vez, utilizando uma forma de consenso descentralizado sobre o estado atual das transações. Isso nunca foi possível antes, outras tentativas de criptomoedas anteriores ao Bitcoin falharam justamente por necessitar de uma entidade central para evitar tal problema.

Fonte: NARAYANAN, A. *et al,* 2016

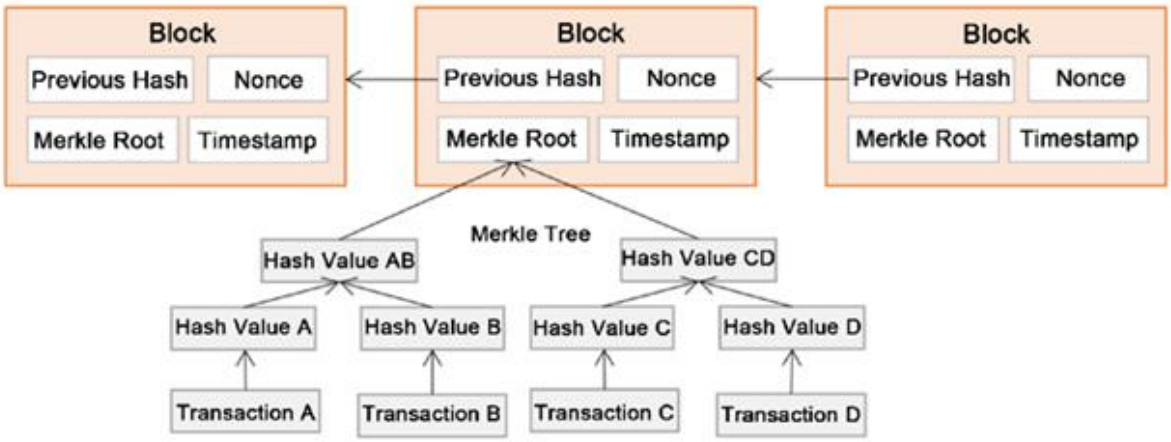

Figura 7 - Ilustração simplificada do blockchain

2.6.1 Estrutura de um bloco

Um bloco é composto de um cabeçalho, que contém metadados, seguido por uma longa lista de transações, sendo que seu tamanho não pode exceder 1 MB. O cabeçalho ocupa 80 bytes, enquanto uma transação possui em média 400 bytes e um bloco possui em média 1900 transações.Portanto, um bloco completo é em média 10000 vezes maior que o cabeçalho do bloco. Com exceção do bloco gênese (primeiro bloco do blockchain), todos os blocos possuem um ponteiro para o bloco anterior. Abaixo seguem os dados de um bloco:

***Block Size* (4 bytes):** Tamanho do bloco, em bytes

***Block Header* (80 bytes):** Vários campos do cabeçalho do bloco

***Transaction counter* (1-9 bytes):** Quantidade de transações no bloco

***Transactions (tamanho variável)*:** Transações registradas no bloco

O Block Header é uma string de 80 bytes com os seguintes campos:

Version (4 bytes): Um número de versão usado para rastrear atualizações de software/protocolo.

***Previous Block Hash* (32 bytes):** Uma referência para o bloco anterior da cadeia, é o hash *double-SHA256*[3] do cabeçalho do bloco anterior produzido no Proof-Of-Work.

[3] SHA256 é uma função de hash criptográfico, double-SHA256 é a função SHA256 aplicada duas vezes

***Merkle Root* (32 bytes):** Um hash da raiz da árvore de merkle das transações do bloco.

***Timestamp* (4 bytes):** O tempo da criação do bloco em *Unix Epoch*.

***Difficulty Target* (4 bytes):** O valor da dificuldade do algoritmo Proof-Of-Work, representado em uma codificação específica de ponto flutuante, seu valor está no intervalo $[0, 2^{256}]$

***Nonce* (4 bytes):** Um contador usado no algoritmo Proof-Of-Work para gerar hashes diferentes.

2.6.2 Árvore de merkle

Uma árvore de Merkle ou árvore de hash é um tipo de árvore para armazenar grandes volumes de dados e que permite uma verificação eficiente e segura de sua integridade. No Bitcoin são usadas para um nó da rede qualquer poder verificar se uma transação está incluída em um bloco de forma eficiente.

Sua construção é feita de baixo pra cima, aplicando recursivamente o algoritmo de hashing *double-SHA256* em pares de nós até só ter um hash, a raiz da árvore. Primeiro é aplicado o hashing nas transações para criar os nós folha da árvore. Depois, para criar os nós pais o hash dos pares de nós filhos são concatenados e o hashing é aplicado novamente. O processo continua até que só exista um nó no topo, a raiz de Merkle (Figura 7), que é armazenada no cabeçalho do bloco.

Fonte: ANTONOPOULOS A. M, 2017

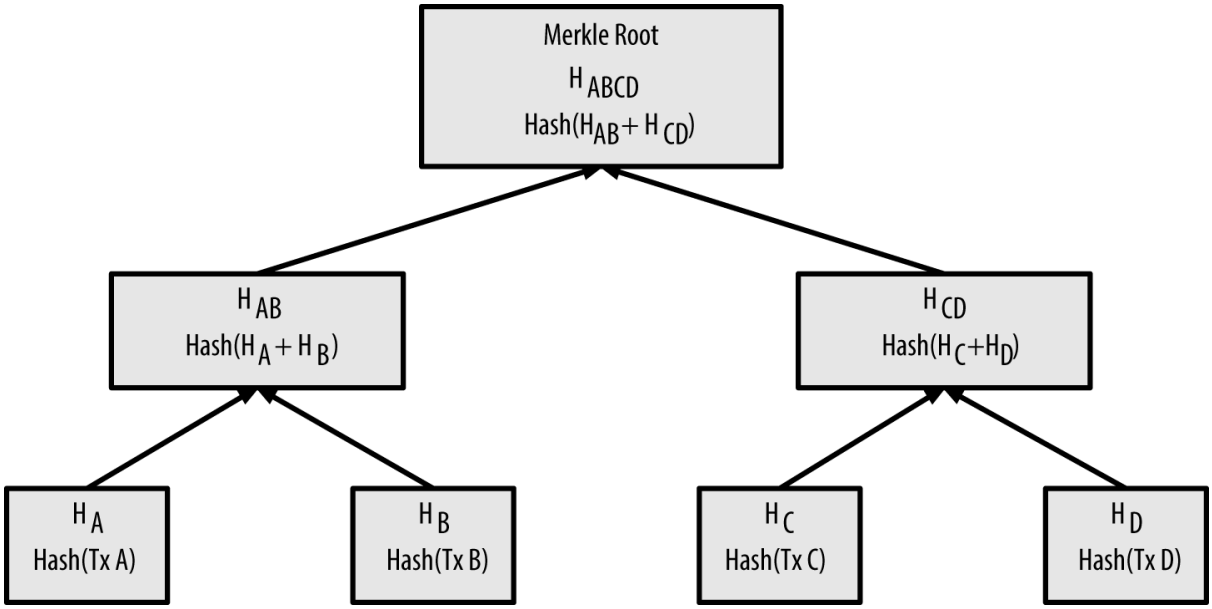

Figura 8 - Ilustração de uma árvore de merkle com 4 transações A, B, C e D.

.

Para verificar que uma determinada transação foi incluída no bloco, é computado o caminho de Merkle, também conhecido como prova de merkle. Isso requer uma computação de no máximo 2logn hashes, onde n é a quantidade de transações do bloco. Dessa forma, é possível ter light nodes (em português, nós leves). Ao contrário dos full nodes que baixam todo o blockchain, esse tipo de nó só baixa o cabeçalho dos blocos e alguns hashes para verificar suas transações. Uma outra propriedade importante dessa árvore é que se alguma transação do bloco é alterada, a raiz de Merkle também se altera.

2.6.3 Mineração

A mineração é o processo de validação, inserção de novas transações e também onde são emitidas novas unidades de Bitcoin no sistema. Tal processo é essencial para manter a segurança do Bitcoin, evitando problemas de gasto duplo, alterações no registro de transações e ataques de negação de serviço. O nome mineração é utilizado devido ao fato de que assim como os metais preciosos, o Bitcoin possui suprimento finito e também é necessário realizar um trabalho custoso para gerar novos bitcoins.

Os nós da rede que realizam esse processo são chamados de mineradores, eles executam um software cliente full node do Bitcoin e competem entre si para

inserir novos blocos. Como o blockchain é mantido por nós anônimos na rede, é necessário que o minerador prove que uma quantidade significante de trabalho foi investido para criar o bloco.

Essa prova é feita através do algoritmo Proof-Of-Work, que consiste em encontrar um *nonce* tal que o hash *double-SHA256* do cabeçalho do bloco seja menor do que um determinado valor de dificuldade (*Difficulty Target*). Portanto, é fácil para os outros nós da rede verificarem que o trabalho foi realizado, porém é computacionalmente difícil resolver o problema, o único método é por tentativa e erro até encontrar um nonce que forneça o hash adequado. A probabilidade de um minerador conseguir inserir um novo bloco é diretamente proporcional ao seu poder computacional em relação ao poder total da rede.

Quando um minerador resolve esse problema, é feito o *broadcast* do bloco criado para todos os nós da rede, que verificam se o bloco não quebra nenhuma regra do sistema do Bitcoin para então adicioná-lo em suas cópias do blockchain. Como o poder computacional da rede está em constante mudança, a dificuldade é reajustada para mais ou menos a cada 2160 blocos, para que a mineração de um novo bloco ocorra em média a cada 10 minutos.

Logo, o Proof-Of-Work é uma forma simples de determinar um consenso entre os participantes da rede sobre o estado das transações. A decisão da maioria é representado pela cadeia mais longa, assim os mineradores sempre trabalham na cadeia com maior prova de trabalho.

Naturalmente, os mineradores recebem um incentivo econômico em forma de bitcoins quando inserem novos blocos no blockchain. Atualmente são emitidos 12,5 BTC por bloco. A cada 210000 blocos (aproximadamente 4 anos) a emissão de moeda cai pela metade. Em 2020, a emissão cairá para 6,25 BTC e assim sucessivamente. Após 6,93 milhões de blocos criados (aproximadamente no ano de 2140) um pouco menos que 21 milhões de bitcoins serão emitidos no total (atualmente gira em torno de 17 milhões) e não será mais possível emitir bitcoins, portanto assim como os metais preciosos o Bitcoin possui um suprimento finito. Na figura 9 temos um gráfico do total de bitcoins criados com o passar do tempo.

Fonte: ANTONOPOULOS A. M, 2017

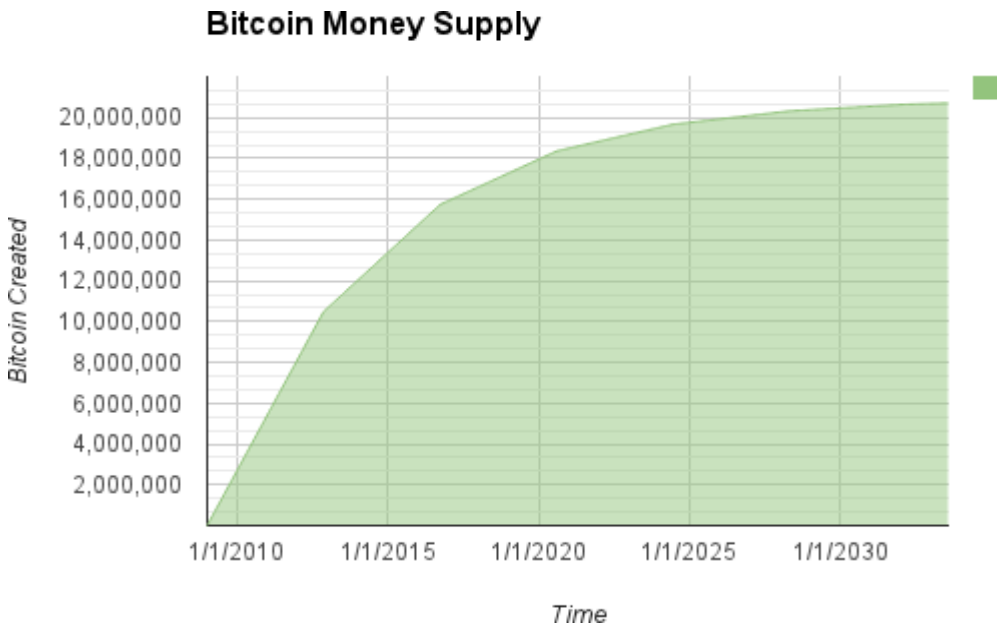

Figura 9 - Bitcoins em circulação em relação ao tempo

A emissão de novos bitcoins é feita pelos mineradores através de uma transação especial, chamada de *coinbase transaction*. Essa transação é construída pelo minerador, não possui input e é a primeira transação do bloco. O *output* dela vai para um endereço de escolha do minerador, e possui um campo de 100 bytes no qual pode ser inserido qualquer dado arbitrário.

Uma outra forma dos mineradores ganharem bitcoins é através da taxa de transação. Como dito anteriormente o excedente dos *inputs* de uma transação é dado para o minerador. No início, não era necessário pagar essas taxas, porém hoje quem não paga corre o risco de não ter sua transação realizada na rede pois os mineradores escolhem as transações com as taxas mais altas para montar o bloco.

2.6.4 Forks temporários e risco de ataque

Como o sistema do Bitcoin é descentralizado, pode existir mais de uma versão do blockchain em um determinado tempo em diferentes nós. Se dois mineradores transmitirem a mesma versão do próximo bloco simultaneamente, isso pode criar o chamado Fork (bifurcação) no blockchain. Esse Fork geralmente não dura muito tempo, eventualmente um minerador insere um bloco em uma das duas ramificações que se torna a cadeia mais longa. Quando um bloco é inserido no

blockchain, usa-se uma heurística de esperar pelo menos 6 confirmações da rede (blocos subsequentes adicionados) para considerar que ele faz parte do blockchain de fato.

Um agente malicioso que deseja atacar a rede pode tentar gastar seus próprios bitcoins mais de uma vez, esse ataque é conhecido como *double spending*. O processo acontece da seguinte maneira, o agente gasta seus bitcoins com algum serviço, espera a quantidade necessária de confirmações **z** requerida pelo prestador, e depois ele tenta modificar sua transação por outra. Assim, é necessário alterar seu bloco, e como seu hash muda, é necessário refazer todo a prova de trabalho dos blocos subsequentes também e ainda produzir uma cadeia mais longa que a honesta. Isso cria um Fork temporário entre a cadeia honesta e a não honesta e caracteriza uma condição de corrida entre os participantes honestos da rede e o agente não honesto, se o agente controla a maior parte do poder computacional da rede eventualmente ele irá criar uma cadeia maior que a dos nós honestos, senão a probabilidade de sucesso diminui exponencialmente em relação à quantidade **z** de confirmações (NAKAMOTO, 2008, p. 6-7).

2.7 Atualizações

No Bitcoin o software de referência para atualizações de protocolo é o Bitcoin Core, quando uma nova funcionalidade é implementada no software ou ocorre alguma alteração no protocolo, os nós da rede podem ou não atualizar seus clientes, assim cada nó vota se aceita ou não a atualização. Basicamente há dois tipos de atualizações: soft fork e hard fork.

Soft fork é uma alteração no protocolo Bitcoin que restringe as regras de consenso aplicado pelos nós (*full nodes* e mineradores). Essas mudanças são compatíveis com versões anteriores do Bitcoin Core e são de consenso de toda a comunidade, ou seja, os nós antigos do Bitcoin Core, aceitarão receber blocos criados por nós novos que atuarão de acordo com as novas regras implementadas pela nova versão. Nesse caso, apenas os mineradores terão que atualizar o software cliente. Usuários comuns podem continuar a utilizar clientes antigos, que agora passarão a aceitar os novos blocos que os mineradores encontrarem.

Um **hard fork** é uma alteração no protocolo do Bitcoin que deixa as regras do

protocolo menos restritas, ou seja, um bloco que antes era considerado inválido antes da atualização pode ser considerado válido pelos nós atualizados. Nesse caso, todos os nós da rede devem atualizar o cliente, caso isso não ocorra haverá uma bifurcação em dois blockchain incompatíveis (versão antes da atualização e depois da atualização) pois a versão antiga não aceitará os blocos da versão mais recente.

3. ETHEREUM

Apesar de ser muitas vezes associada com o Bitcoin, a tecnologia do blockchain tem muitas outras aplicações que vão além de apenas moedas digitais peer-to-peer. Tendo isso em mente, o Ethereum foi proposto em 2013 por Vitalik Buterin, e através de um *crowdfunding* sua primeira versão foi concluída em 2015 e está em constante desenvolvimento desde então.

Assim como o Bitcoin, o Ethereum possui uma criptomoeda que funciona em uma rede pública descentralizada utilizando o blockchain. Porém, existem algumas diferenças entre os dois, sendo que as distinções mais importantes se referem ao propósito e capacidades.

O Bitcoin oferece uma aplicação particular da tecnologia do blockchain, um sistema de moeda digital peer to peer que permite pagamentos online. Já o Ethereum, apesar de também ser usado como moeda digital, foi criado com o propósito de funcionar como uma plataforma de computação distribuída que executa programas chamados contratos inteligentes.

De acordo com o cofundador do Ethereum, Gavin Wood “O Bitcoin é primeiramente uma moeda, isso é apenas uma aplicação particular do blockchain. Porém está longe de ser a única aplicação. Tomando um exemplo passado de uma situação similar, o e-mail é um uso particular da internet, e com certeza ajudou a popularizá-la, porém existem vários outros usos”.

A criptomoeda minerada no blockchain do Ethereum é chamada de Ether, e além de ser usada como uma forma de pagamento, ela também é usada por desenvolvedores de aplicações para pagar por taxas de transações e serviços na rede.

3.1 Visão geral

3.1.1 O que é um contrato inteligente?

Contrato inteligente é um termo utilizado para descrever programas de computador que facilitam a troca de dinheiro, conteúdo, propriedade, ações ou qualquer outra coisa de valor. Quando é armazenado no blockchain, um contrato inteligente é executado automaticamente quando eventos específicos são atingidos (figura 10) sem possibilidade de censura, fraude, tempo de inatividade ou interferência de terceiros.

O contrato inteligente funciona da seguinte forma: Primeiro é definido entre as partes o que colocar no código do contrato, esse contrato então é ativado quando eventos determinados no contrato ocorrem, feito isso ele é executado pelos nós da rede e pode ocorrer uma transferência de valores (Figura 10).

Fonte : blockchainhub

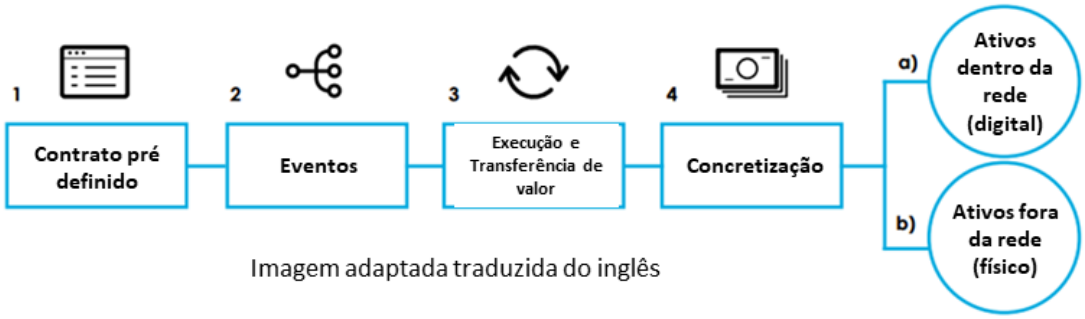

Figura 10 – Fluxo de execução de um contrato inteligente

3.1.2 Ethereum Virtual Machine

Antes da criação do Ethereum, as aplicações no blockchain eram feitas para realizar uma quantidade muito limitada de operações e para cada aplicação era necessário um blockchain totalmente novo ou construir algo no Bitcoin, o que era muito difícil e custoso para os desenvolvedores.

O núcleo da inovação do Ethereum é a EVM (Ethereum Virtual Machine, ou Máquina Virtual do Ethereum), um ambiente de execução que está presente nos nós da rede do Ethereum. Esse ambiente é responsável por rodar os contratos inteligentes, que são códigos compilados de linguagens de script turing completa.

A EVM tornou o processo de criar aplicações no blockchain muito mais fácil e eficiente. Em vez de criar um blockchain para cada aplicação, o Ethereum permite o desenvolvimento de inúmeras aplicações em uma só plataforma e com uma linguagem de programação simples para os desenvolvedores.

3.1.3 Aplicações descentralizadas

Atualmente, diversas aplicações descentralizadas, mais conhecidas como DAPPS (do inglês, decentralized applications) estão sendo desenvolvidas no Ethereum. Podemos citar os ICOs (Inicial Coin Offering, em português, oferta inicial de moeda) como um fator relevante para o grande número de startups que estão desenvolvendo na plataforma.

ICO é uma forma de crowdfunding para financiar tais aplicações ou outras criptomoedas que é muito utilizado no Ethereum, é semelhante a um IPO (Initial Public Offering). O processo funciona da seguinte forma, um contrato inteligente é criado com uma certa quantidade de tokens definido pela startup, feito isso os usuários que querem participar do crowdfunding enviam Ether para esse contrato da

startup, que por sua vez envia os tokens na conta do participante automaticamente. Esses tokens são um tipo de moeda que são criados dentro da rede pública do Ethereum, geralmente tem alguma utilidade na aplicação em questão, ou são utilizados para trocar por uma outra criptomoeda posteriormente. Bilhões de dólares já foram arrecadados dessa forma, assim, o ICO pode ser considerado como a primeira aplicação efetiva do Ethereum.

Alguns tipos de aplicações que estão sendo desenvolvidas no Ethereum e em outros blockchains:

- Sistema de votação
- Cadeia de suprimentos e logística
- Ativos digitais (*tokens*)
- Registro de documentos
- Jogos de aposta
- E-commerce
- Direitos autorais
- Armazenamento de dados
- *Trading*
- Sistema de identidade digital
- Empréstimos peer to peer
- Digitalização de ativo

3.2 Componentes do Ethereum

3.2.1 Sistema de contas

Diferente do Bitcoin, o Ethereum possui um sistema de contas para armazenar a sua moeda nativa, o ether. Cada conta possui um endereço e a transferência de valores e informações entre contas ocorre através de funções de transição de estado. Uma conta do Ethereum possui quatro campos:

- O **nonce**, um contador utilizado para garantir que cada transação seja processada apenas uma vez
- O **saldo de Ether** atual da conta
- O **código do contrato** da conta, se possuir
- O **armazenamento** da conta (vazio por padrão)

o Ether é o principal combustível do Ethereum, e é usado para pagar as taxas de transações. Em geral, há dois tipos de contas:

- **contas de propriedade externa (de usuários)**: Semelhantes às contas do Bitcoin, porém com informações sobre seu estado. Utiliza o sistema de chave pública e privada, sendo que o endereço é a própria chave pública do usuário.

- **contas de contrato:** Contas que possuem código (contratos inteligentes), esses contratos são ativados por transações e não possuem chave privada

Uma conta externa não possui código, e mensagens podem ser enviadas de uma conta externa. Em uma conta de contrato, toda vez que ela recebe uma mensagem seu código é ativado, permitindo a escrita e leitura no armazenamento interno e o envio outras mensagens ou ether para outras contas. Uma conta de contrato tem total controle do seu saldo de ether e outras variáveis persistentes.

Vale ressaltar que ao contrário do Bitcoin, existe de fato um campo que armazena o saldo de uma conta. Então os ethers não vão de uma transação para a outra como ocorre no Bitcoin, mas sim de conta para conta através da transição de estado nas contas causada pela transação.

3.2.2 Transações

O termo transação no Ethereum é designado para se referir a pacotes de dados assinados que contém uma mensagem e são enviados por uma conta de propriedade externa. Uma transação é composta pelos seguintes dados:

- O **destinatário** da mensagem
- A **assinatura digital** do remetente
- A **quantidade de ether** a ser transferida do remetente para o destinatário
- Um **campo dados** opcional
- O **limite de gas**, que representa a quantidade máxima de gás que uma transação pode consumir
- O **preço do gas** em ether, representando a taxa paga por passo computacional para os mineradores

Os primeiros três campos são padrões nas criptomoedas em geral. O **campo de dados** pode ser utilizado por um contrato para acessar dados. Por exemplo, se o contrato estiver funcionando como um sistema de registro de nome de domínio, então ele pode interpretar o dado tendo dois campos, o primeiro sendo o domínio para registrar e o segundo o endereço IP associado. Então o contrato leria o dado dessa mensagem e o armazenaria.

Quando uma transação é feita na rede, todos os nós realizam a sua computação que pode ser apenas uma transferência de ether entre contas externas ou computações de um contrato inteligente. Isso custa tempo e energia. **Gas** é o mecanismo usado para pagar os mineradores por esse serviço, cada passo computacional na EVM geralmente consome 1 unidade de gas e há uma taxa de 5 gas por byte da transação.

Assim, os mineradores recebem uma pequena quantidade de Ether por transação, o pagamento em ether é calculado como:

Pagamento (em Ether) = gas usado x preço do gas (em Ether)

Quando maior o preço do gás, maior vai ser a prioridade para a transação ser confirmada na rede, pois os mineradores costumam pegar as transações com taxas mais altas.

3.2.3 Mensagens

Contratos conseguem enviar mensagens para outros contratos. Diferente das transações, mensagens são objetos que apenas existem no ambiente de execução do Ethereum. Essencialmente uma mensagem é uma transação, exceto pelo fato de que ela é produzida por um contrato e não um usuário. Ela possui todos os campos de uma transação exceto o preço do gas.

A mensagem é produzida quando um contrato que está executando um código faz uma chamada para produzir e executar uma mensagem. Assim como uma transação, a mensagem faz com que o destinatário execute seu código. Portanto, contratos podem se comunicar com outros contratos da mesma forma que usuários.

Note que o limite de gás atribuído para uma transação se aplica para a transação e todas suas sub-execuções. Por exemplo, se um usuário envia uma transação para um contrato A com 1000 gas, A consome 600 gas e envia uma mensagem para uma conta B que consome 300 gas, então A ainda pode gastar 100 gas após a execução de B.

3.2.4 Transição de estado

Quando uma transação é executada no Ethereum, ocorre uma transição de estado no sistema. A função de transição de estado pode ser definida da seguinte forma:

- Checar se a transação é válida (assinatura correta, e nonce válido), se não , retorna um erro.
- Calcular a taxa de transação como limite de gas x preço do gas, e determinar o endereço do remetente pela assinatura. Subtrair taxa do saldo do remetente e incrementar seu nonce, se não tiver saldo suficiente retorna um erro
- Subtrair uma certa quantidade de gas por byte da transação.
- Transferir o valor da conta do remetente para o destinatário. Se a conta ainda não existe, criá-la. Se a conta do remetente for um contrato, rodar o contrato até terminá-lo ou até esgotar o gas
- Se a transferência falhar pelo fato de o remetente não ter saldo suficiente, ou esgotar o gas na execução do contrato, toda a mudança de estado é revertida menos o pagamento da taxa, que será adicionado na conta do minerador
- Caso contrário, o valor em Ether que sobrou de gas é devolvido para o remetente, e as taxas são pagas para o minerador.

3.2.5 Unidades

A menor unidade do Ethereum é denominada wei, e equivale a 10^{-18} Ether. As denominações definidas para as unidades são:

- wei
- szabo:10^{12} wei
- finney: 10^{15} wei
- ether: 10^{18} wei

3.3 Modelo de execução de código na EVM

A parte do protocolo responsável por processar as transações é a máquina virtual do Ethereum (EVM).

A EVM é uma máquina virtual que executa código turing completo. Sua única limitação é que a quantidade de computação que pode ser feita é limitada pela quantidade de gás dada.

Além disso, a EVM possui uma arquitetura baseada em pilha. Uma stack machine (máquina baseada em pilhas) é um computador que utiliza uma estrutura de pilha para armazenar valores temporários.

O tamanho de cada item na EVM é 256 bits, e a pilha tem um tamanho máximo de 1024 itens. A EVM possui memória, onde os itens são armazenados como um array de bytes, a memória é volátil.

A EVM também possui armazenamento. Diferente da memória, não é volátil e é mantido como uma parte do estado do sistema. O código de programa é armazenado separadamente, em uma ROM virtual que só pode ser acessado através de instruções especiais (Figura 11). Dessa forma, a EVM difere da arquitetura de Von Neumann, no qual o código do programa fica na memória ou no armazenamento.

Fonte: Adaptado de Wood G.

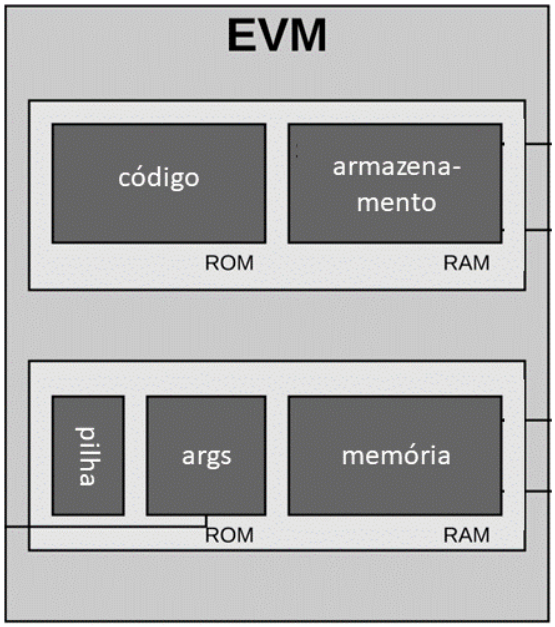

Imagem adaptada traduzida do inglês

Figura 11 – EVM

A EVM também possui sua própria linguagem chamada de “bytecode da EVM”. O código escrito por programadores para criar contratos inteligentes é tipicamente feito em uma linguagem de alto nível de script, que é compilado em bytecode.

Antes de executar uma computação em particular, o processador verifica se as informações necessárias estão disponíveis e são válidas.

No início da execução, a memória e pilha estão vazias e o contador do programa é 0. A EVM então executa a transação recursivamente, calculando o estado do sistema e o estado da máquina para cada loop. O estado do sistema é simplesmente o estado global das contas do Ethereum. O estado da máquina é composto por:

- Gás disponível
- Contador do programa
- Conteúdo da memória
- Número ativo de palavras na memória
- Conteúdo da pilha

Itens de pilha são adicionados ou removidos da parte mais à esquerda da série. Em cada ciclo, a quantidade de gás apropriada é reduzida do gás restante e o contador do programa é incrementado. No final de cada loop, existem três possibilidades:

1. A máquina atinge um estado excepcional (por exemplo, gás insuficiente, instruções inválidas, itens de pilha insuficientes, itens de pilha acima de 1024,

etc.) e, portanto, deve ser interrompida, com quaisquer alterações descartadas
2. A sequência continua a processar no próximo loop
3. A máquina atinge uma parada controlada (o final do processo de execução)

Supondo que a execução não atinge um estado excepcional e atinge uma parada "controlada" ou normal, a máquina gera o estado resultante, o gás restante após essa execução, o subestado acumulado e a saída resultante.

3.4 Contratos inteligentes

Como dito anteriormente, um contrato é escrito por programadores em uma linguagem de alto nível que posteriormente é compilado em bytecode, essas linguagens são geralmente referenciadas como “linguagem orientada a contrato” por ter um foco específico em contratos inteligentes. Atualmente existem várias linguagens de programação para a rede pública do Ethereum, sendo que as principais são:

- **Solidity**: Semelhante à linguagem Javascript. Atualmente, essa é a linguagem de script de contrato inteligente mais popular e funcional.
- **Serpent**: Semelhante à linguagem Python, e foi popular nos primórdios do Ethereum.
- **LLL:** Semelhante ao Lisp e só foi realmente usada bem no começo da história do Ethereum

No Ethereum, um contrato inteligente é gravado criando uma nova conta com algum código e fazendo o upload para o blockchain da Ethereum em uma transação.

A forma mais fácil de fazer esse upload é através da carteira oficial do Ethereum “Mist”, que possui uma interface gráfica simples.

Cada nó de mineração executará o contrato inteligente em seu computador usando sua máquina virtual Ethereum como parte do processo de mineração e chegará a uma conclusão sobre a saída. Então cada nó na rede Ethereum chegará à mesma conclusão, porque eles estão executando o mesmo código de contrato com a mesma informação fornecida.

Quando um bloco é minerado, o minerador vencedor fará o broadcast do bloco para o resto da rede, e os outros computadores irão validar esse resultado, depois adicionam o bloco às suas cópias do blockchain. É assim que o estado do blockchain do Ethereum é atualizado.

3.5 Ethereum x Bitcoin blockchain

Apesar do Ethereum e o Bitcoin serem ambos criptomoedas baseadas em blockchain, o Ethereum possui várias diferenças em seu blockchain por se tratar de uma versão mais sofisticada do Bitcoin, dentre elas podemos citar:

- **Tempo menor de criação de blocos:** O tempo médio entre criação de blocos no Ethereum é de 15 segundos, enquanto no Bitcoin esse valor é de 10 minutos. Isso significa que uma transação é confirmada mais rapidamente no Ethereum. No caso do Bitcoin, para considerar uma transação irreversível probabilisticamente, é necessário esperar 6 confirmações da rede (cerca de 1 hora), enquanto no Ethereum é necessário 12 confirmações (cerca de 3 minutos).
- **Blocos menores**: No Bitcoin, o tamanho máximo de um bloco é 1 MB, enquanto no Ethereum o tamanho do bloco é baseado em uma quantidade máxima de gas gasto por bloco. Atualmente esse valor é de 1500000 gas. Em termo de dados, o tamanho do bloco geralmente não passa de 2 KB.
- **Mais dados nos blocos**: Além de armazenar uma lista de todas as transações em cada bloco, o Ethereum também armazena o estado mais recente de cada conta criada.
- **Algoritmo de mineração**: O problema matemático que deve ser resolvido no processo de mineração do Ethereum é o Ethash, que é um pouco diferente do Bitcoin. Isso permite o uso de hardware comum para mineração como placas de vídeo, pois reduz a eficiência de hardwares específicos para mineração conhecidos como ASIC (Application Specific Integrated Circuits, ou seja, circuitos integrados de aplicação específica). Essa mudança tem em vista manter a rede descentralizada.
- **Suprimento infinito**: Diferente do Bitcoin, o Ethereum não possui um suprimento finito. Porém, a taxa de emissão de moedas diminui todo ano até chegar a um valor de aproximadamente 1% / ano.

3.6 Atualizações

O Ethereum é um sistema que está em constante desenvolvimento open source. Seu principal cofundador Vitalyk Buterin participa ativamente das atualizações do protocolo e junto com uma comunidade de desenvolvedores e pesquisadores trabalha para cumprir diversos objetivos.

As atualizações das versões do protocolo do Ethereum ocorrem através de hard forks, sendo que cada uma das versões do seu software cliente oficial tem um nome e implementam mudanças significativas no sistema. A próxima atualização planejada é Metropolis

Atualmente o mecanismo de consenso utilizado na rede é o proof of work, porém existe uma mudança prevista para o proof of stake, que é um mecanismo que consome menos energia pois não é necessário realizar computações custosas.

Além disso, um dos focos da equipe do Ethereum hoje é resolver o problema da escalabilidade. Como visto nas seções anteriores, há uma limitação na

arquitetura do Bitcoin e Ethereum, pois todos os mineradores da rede precisam validar cada transação e há limitações de intervalo de criação e tamanho do bloco, o que consequentemente acarreta em uma quantidade de transações por segundo limitada.

Existe uma proposta de solução para a escalabilidade no Ethereum que já está em fase de desenvolvimento e será discutida com detalhes no próximo capítulo.

4 ESCALABILIDADE

Atualmente as criptomoedas baseadas em blockchain realizam apenas uma quantidade limitada de transações por segundo, isso se deve à sua arquitetura atual. Vimos nos capítulos anteriores que essas criptomoedas possuem um limite de transações que podem ser incluídas em cada bloco e cada bloco é criado em um intervalo constante de tempo. Ainda há o fato de que cada nó da rede valida todas as transações, o que é muito ineficiente, porém faz com que o sistema seja seguro e descentralizado. O Bitcoin possui uma capacidade estimada de 7 transações por segundo, enquanto o Ethereum possui uma capacidade de 20 transações por segundo.

Essa limitação ficou evidente no ano de 2017, quando houve uma grande valorização no preço das criptomoedas e consequente aumento do uso da rede pelos mais diversos usuários. Em alguns períodos as transações no Bitcoin demoravam horas ou até mesmo dias para serem confirmadas dependendo do valor pago de taxa, a taxa média de uma transação chegou a custar 50 dólares.

O Ethereum sofreu do mesmo problema nesse ano. O episódio mais marcante foi quando apenas um jogo virtual baseado em contratos inteligentes chamado CryptoKitties congestionou a rede inteira.

Assim, é evidente que atualmente não há como ter uma adoção em massa dessas criptomoedas, visto que sistemas centralizados como o Visa tem capacidade de processar mais de 50 mil transações por segundo.

A seguir serão abordadas as principais propostas para resolver o problema da escalabilidade.

4.1 Aumento do limite do tamanho do bloco

O limite de 1 MB no tamanho do bloco foi introduzido no Bitcoin em 2010 (de um limite anterior de 36MB) para combater os riscos relacionados a ataques de spam e negação de serviço distribuído (DDoS).

Ataques DDoS podiam ser realizados através de transações com quantidades muito pequenas de Bitcoins, conhecidas como transações de poeira. Nos primórdios do Bitcoin esse era um risco real, pois as taxas de transações eram praticamente zero, atualmente com as taxas mais altas devido a valorização do Bitcoin esse tipo de ataque não é economicamente viável.

Então, basicamente, o limite de 1MB significa que a quantidade de transações que podem caber em um único bloco de dados a ser colocado na cadeia é limitada. No entanto, mais e mais pessoas estão acostumadas com a ideia de que o tamanho do bloco deve ser aumentado para acomodar o aumento da popularidade do Bitcoin como método de pagamento. Este lado do debate argumenta que a moeda precisa aumentar a quantidade de transações por segundo para promover a adoção em massa e impulsionar o Bitcoin para um nível que rivaliza com sistemas de pagamento como o Visa ou o PayPal.

Por outro lado, outros argumentam que, como o aumento do tamanho do bloco significaria a necessidade de aumentar o poder de processamento, armazenamento e largura de banda, a mineração se tornaria mais centralizada pois uma parcela cada vez menor de nós teria condições de possuir tais recursos.
Para que os tamanhos dos blocos Bitcoin aumentem é necessário um hard fork no sistema, então deve haver um consenso de 100% entre a comunidade mineradora sobre o tamanho, caso contrário, veríamos uma bifurcação em dois blockchain, um da versão mais antiga e o outro da mais nova. É exatamente assim que a criptomoeda Bitcoin Cash foi criada.

O Bitcoin Cash se originou de um hard fork do Bitcoin em 1º de agosto de 2017 e gerou um blockchain alternativo ao Bitcoin pois a maior parte dos mineradores não aceitou essa mudança. O Bitcoin Cash implementou um limite de 8MB, o que significa tempos de transação mais rápidos, porém como a maioria dos mineradores não aceitaram esse fork, a rede possui um menor poder computacional em relação ao Bitcoin e consequentemente é mais vulnerável a ataques de gasto duplo.

4.2 Lightning network

A Lightning Network, também conhecida como uma solução de segunda camada (pois não altera o protocolo do Bitcoin) é um rede construída sobre Bitcoin que permite às pessoas instantaneamente enviar / receber pagamentos e reduzir as taxas de transação, mantendo-as fora da rede principal. Esse sistema utiliza o Script do Bitcoin, uma mini linguagem de programação baseada em pilha, que é bem limitada em relação aos contratos inteligentes do Ethereum mas que é suficiente para criar esse sistema.

A Lightning Network é um sistema de contratos inteligentes construído sobre a camada de rede do Bitcoin que permite pagamentos rápidos e baratos diretamente entre duas partes através de canais de pagamento. O sistema funciona da seguinte forma:

- Uma carteira de multi-assinatura que contém uma certa quantidade de bitcoin (fornecida por pelo menos uma das duas partes) é criada
- O endereço da carteira é armazenado no blockchain público do Bitcoin, incluindo um contrato inteligente que prova quanto deste depósito de bitcoin pertence a quem
- Depois que esse canal de pagamento é configurado uma vez, é possível que essas duas partes realizem uma quantidade ilimitada de transações sem nunca tocar nas informações armazenadas no blockchain
- Em cada transação, ambas as partes assinam um saldo atualizado para sempre refletir quanto do bitcoin cada parte tem
- O saldo atualizado não é transmitido para o blockchain, mas ambas as partes mantêm uma prova criptográfica desse saldo

- Quando o canal de pagamento for fechado pelas partes ou houver uma disputa, as partes transmitem a prova do saldo mais recente para receber sua parte dos fundos da carteira multi-assinatura

Tudo isso parece complicado para o usuário final, mas normalmente todos os itens acima acontecerão automaticamente em segundo plano.

O uso de canais de pagamento da Lightning Network permite que os usuários efetuem transações diretamente entre si, em vez de transmitirem seus negócios para o blockchain. Se houver algum tipo de disputa em relação aos saldos na Lightening Network, o saldo mais recente fornecido por qualquer uma das duas partes decidirá como os fundos na carteira multi-assinatura serão divididos.

Não é necessário que um usuário crie um canal de pagamento toda vez que deseja realizar transação com um outro usuário, apenas que exista um caminho de canais de um usuário para o outro. Por exemplo, suponha que Alice deseja enviar 10 satoshis para David e possui um canal aberto com Bob mas não com David, e Bob possui um canal aberto com David. Alice pode então usar o canal aberto com Bob para enviar os 10 satoshi para David.

Assim é possível encontrar caminhos para realizar transações na rede de forma similar ao de rotas de pacotes na internet. Os nós no caminho não precisam ser confiáveis, pois o pagamento é garantido utilizando um script que reforça atomicidade (ou pagamento é realizado ou falha).

Atualmente a Lightening Network está em fase de testes no Bitcoin e mostra resultados promissores. Também está sendo desenvolvida uma versão desses canais de pagamento para o Ethereum, conhecida com Raiden Network.

Abaixo são citados os principais prós e contras da Lightening Network

4.2.1 Prós

- Pequenos pagamentos são possíveis: como as taxas são proporcionais ao valor do pagamento, você pode pagar uma fração de um centavo; a contabilidade é feita até mesmo em milésimos de um satoshi.
- Os pagamentos são liquidados instantaneamente: o dinheiro é enviado no tempo necessário para atravessar a rede até o seu destino e voltar, normalmente uma fração de segundo.
- Privacidade melhorada: Nem toda transação é armazenada no blockchain público, apenas uma vez quando o canal de pagamento é eventualmente fechado e o saldo é pago para ambas as partes.

4.2.2 Contras

- Falhas de nós: se um dos nós não responder, os usuários podem ter que esperar por horas para fechar um canal de pagamento e reenviar os fundos por meio de uma rota alternativa.

- Nenhum pagamento off-line: os usuários não podem pagar alguém que não esteja on-line
- Não é ideal para pagamentos de valores altos: embora possa existir uma rota através de vários canais de pagamento, os fundos nas carteiras de vários nós podem não ser suficientes para transferir grandes fundos
- Centralização: A Lightning Network pode incentivar a centralização, criando hubs de pagamento

4.3 Sharding

O Sharding é, na verdade, muito mais antigo que a tecnologia blockchain e foi implementado em vários sistemas de banco de dados. Essencialmente, o particionamento é um método específico para particionar horizontalmente os dados em um banco de dados. Mais geralmente, o banco de dados é dividido em pequenas partições chamadas “shards”, que, quando agregadas, formam o banco de dados original.

Os nós na rede das criptomoedas não têm privilégios especiais e todos os nós da rede armazenam e processam todas as transações. Como resultado, em uma rede do tamanho do Ethereum, problemas como altos custos de gás e tempos de confirmação de transação mais longos se tornam problemas visíveis quando a rede é sobrecarregada. A rede é tão rápida quanto os nós individuais, em vez da soma de suas partes.

O Sharding ajuda a aliviar esses problemas fornecendo uma solução interessante, mas complexa. O conceito envolve o agrupamento de subconjuntos de nós em shards que, por sua vez, processam transações específicas para esse shard. Assim o sistema processa transações em paralelo, permitindo uma escalabilidade horizontal conforme a quantidade de nós aumenta.

Os desenvolvedores do Ethereum planejam implementar o sharding após uma atualização para método de consenso Proof of Stake (Prova de participação), e o método de sharding planejado é o “sharding de estado” pois além de particionar a validação de transações, o estado da rede também é particionado.

4.3.1 Proof of Stake

O Proof of Stake é um algoritmo de consenso alternativo ao Proof of Work. No algoritmo em vez de resolver um problema matemático, cada minerador precisa colocar em aposta uma certa quantidade de criptomoeda, o sistema então seleciona um minerador aleatoriamente para gerar um novo bloco, que então recebe sua recompensa. Nesse modelo, a probabilidade de um certo minerador ser selecionado é diretamente proporcional ao tanto que ele aposta, e se algum desses mineradores tentar burlar o sistema, ele é penalizado com a perda total de seus fundos apostados.

4.3.2 Implementação no Ethereum

O estado atual do blockchain do Ethereum é conhecido como o “estado global” e é o que todos podem ver quando observam o blockchain em uma instância específica. A parte complicada na implementação do particionamento no Ethereum é que ao dividir os nós em subconjuntos menores, esses subconjuntos precisam ser capazes de processar conjuntos específicos de transações enquanto atualizam simultaneamente o estado global da rede, tudo isso garantindo sua validade.

O Sharding no Ethereum será implementado em um lançamento em duas fases. A primeira fase será a camada de dados que consiste no consenso de quais dados estão nos shards. A fase dois é a camada de estado. Um resumo geral de como isso pode funcionar está a seguir.

Cada shard é atribuído a um grupo específico de transações que é determinado pelo agrupamento de contas específicas (incluindo contratos inteligentes) em um shard. Cada grupo de transações tem um cabeçalho e um corpo que consistem no seguinte.

Cabeçalho:

- O ID do shard do grupo de transações
- Atribuição de validadores através de amostragem aleatória (verifica as transações no shard)
- Raiz do estado (raiz de merkle do estado do shard antes e depois das transações adicionadas)

Corpo:

- Todas as transações que pertencem ao grupo de transações que fazem parte do shard específico.

As transações são específicas para cada shard e ocorrem entre contas nativas desse shard. Quando as transações são verificadas, o estado global da rede muda e os saldos das contas, o armazenamento, etc. são atualizados. Para que o grupo de transações seja verificado como válido, a raiz do pré-estado do grupo de transações deve corresponder à raiz do estado do shard no estado global. Se eles corresponderem, o grupo de transações será validado e o estado global será atualizado por meio da raiz de estado e do ID do shard.

Em vez de apenas conter uma raiz do estado, cada bloco do blockchain do Ethereum agora contém uma raiz do estado e a raiz do grupo de transações. A raiz do grupo de transações é a raiz do merkle de todos os grupos de transações dos shards específicos para esse bloco de transações. Basicamente, existe uma raiz de merkle de todos os diferentes shards que contêm os grupos de transações atualizados e verificados. Essa raiz é armazenada no blockchain juntamente com a raiz do estado atualizada.

O consenso dentro de um shard é alcançado por meio de um consenso de Prova de Estaca de nós selecionados aleatoriamente que são aplicados a um shard para uma rodada de consenso específica. Isso não apenas fornece a finalidade do consenso, que é necessário dentro dos fragmentos, mas também fornece uma

defesa particular para um ataque que uma blockchain de Prova de Trabalho seria suscetível nesse caso.

4.3.3 Comunicação entre shards

Um protocolo de comunicação entre shards é necessário pois uma conta de um determinado shard pode interagir com outra.

A comunicação entre shards é obtida através da aplicação do conceito de recibos de transações. O recibo de uma transação é armazenado em uma raiz de merkle na rede que pode ser facilmente verificada, mas que não faz parte da raiz do estado. O shard que recebe uma transação de outro shard verifica a raiz do merkle para garantir que o recebimento não tenha sido gasto. Essencialmente, os recibos são armazenados em uma memória compartilhada que pode ser verificada por outros shards, mas não alterada. Portanto, por meio de um armazenamento distribuído de recibos, os shards podem se comunicar uns com os outros.

4.3.4 Pontos positivos e negativos

Os pontos positivos dessa proposta de solução é que permite uma escalabilidade horizontal da rede conforme a quantidade de nós aumenta e também a rede se mantém descentralizada.

O ponto negativo é que a rede se torna um pouco mais vulnerável do que o sistema atual. De acordo com os pesquisadores do Ethereum probabilisticamente é necessário apenas 33% de controle da rede em contraste com os 51% do sistema atual para realizar um ataque com sucesso em alguma shard, apesar de os validadores de cada shard serem selecionados aleatoriamente.

5 CONCLUSÃO

As criptomoedas representam uma grande inovação não só do ponto de vista financeiro, mas também do ponto de vista tecnológico. Diversas foram as tentativas anteriores ao Bitcoin de criar uma forma de moeda digital descentralizada e segura, mas nenhuma delas obteve sucesso. Tanto o Bitcoin como o Ethereum são considerados seguros, pois até o atual momento (25/10/2018) não houve nenhuma transação alterada no blockchain após a quantidade de confirmações recomendada.

O fato das criptomoedas serem baseadas em provas criptográficas em vez de confiança permite que qualquer pessoa com acesso à internet possa criar uma conta, realizar pagamentos e recebimento de dinheiro do mundo todo, sem restrições e sem a necessidade de fornecer dados pessoais. Assim, pode-se dizer que as criptomoedas facilitam a transferência de valor entre as pessoas.

Existem inúmeras possibilidades à vista além de um sistema de pagamento online peer to peer descentralizado, o que por si só já é impressionante. Os contratos inteligentes possuem potencial para desburocratizar diversas indústrias e sistemas que necessitam de confiança entre as partes, aumentando a eficiência através da automatização que contratos normais não possuem.

Naturalmente, nem tudo é perfeito, como toda a tecnologia recente, as criptomoedas baseadas em blockchain atualmente tem suas limitações, sendo que a maior limitação atualmente é a escalabilidade.

Porém, como vimos no capítulo anterior, existem diversas propostas de soluções para a escalabilidade, as soluções abordadas no trabalho são as principais atualmente, porém existem muitas outras que estão sendo desenvolvidas. Dessa forma, é provável que essa limitação seja contornada nos próximos anos, tornando possível a adoção dessas criptomoedas pela população em geral.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ANTONOPOULOS A. M. **Mastering bitcoin: Unlocking digital cryptocurrencies**. 2 ed. O'Reilly media, 2017.

A Next-Generation Smart Contract and Decentralized Application Platform. Disponível em: <https://github.com/ethereum/wiki/wiki/White-Paper>. Acesso em: 01 out. 2018.

Bitcoin Avg. Transaction Fee historical chart. Disponível em: <https://bitinfocharts.com/comparison/bitcoin-transactionfees.html>. Acesso em: 06 maio 2018

BARRERA, A. **A Guide to Bitcoin (Part I): A look under the hood.** Disponível em: <http://tech.eu/features/808/bitcoin-part-one>. Acesso em: 10 ago. 2018.

Bitcoin Developer Guide. Disponível em: <https://bitcoin.org/en/developer-guide>. Acesso em 10/08/2018

Bitcoin Wiki. Disponível em: <https://en.bitcoin.it/wiki>. Acesso em: 10 ago. 2018.

ESTEVÃO, P. **The Bitcoin Transaction Life Cycle**. Disponível em: <https://imgur.com/a/BCvZr> Acesso em: 25 ago. 2018.

NAKAMOTO, S. **Bitcoin: A Peer-to-Peer Electronic Cash System**. 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 25 ago. 2018.

NARAYANAN, A. *et al*. **Bitcoin and Cryptocurrency Technologies**. 1 ed. Princeton University Press, 2016.

POON, J.; DRYJA, T. **The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments.** Disponível em: <https://lightning.network/lightning-network-paper.pdf>. Acesso em: 01 nov. 2018.

VISA INC. **Visa inc. at a glance**. 2015. Disponível em: <https://usa.visa.com/dam/vcom/download/corporate/media/visa-fact-sheet-jun2015.pdf>. Acesso em: 05 maio 2018

Wood, G. **Ethereum: A secure decentralised generalised transaction ledger.** Disponível em: <https://gavwood.com/paper.pdf>. Acesso em: 15 set. 2018.