Capítulo 3

Análise de mecanismos para consenso distribuído aplicados a Blockchain

Charles C. Miers (UDESC), Guilherme P. Koslovski (UDESC),
Maurício A. Pillon (UDESC), Marcos A. Simplício Jr. (USP),
Tereza C. M. B. Carvalho (USP), Bruno B. Rodrigues (UZH),
João H. F. Battisti (UDESC)

Abstract

The Blockchain concept has recently emerged as an alternative approach for e-commerce payment, based on decentralized systems that do not rely on trusted institutions. Actually, since Blockchain was first proposed for use in cryptocurrencies, several solutions have appeared that employ this technology in a variety of domains, such as creating distributed registries of smart contracts. Whichever the target application domain, though, solutions implementing Blockchain use a set of well-known technologies, such as encryption, Merkle Trees, P2P networks, and consensus mechanisms. The latter are of particular research interest today, since most consensus mechanisms used in the early cryptocurrencies (e.g., Bitcoin) involve considerable computational power to ensure system consistency. Aiming to evaluate the state of the art on the area, this manuscript reviews Blockchain concepts and classifications, focusing specifically on the underlying consensus mechanisms and security aspects of the resulting solutions. We also describe a brief experiment using Ethereum and MultiChain, aiming to analyze security aspects of the Practical Byzantine Fault Tolerance (PBFT) and Proof-of-Work (PoW) consensus approaches.

Resumo

O conceito de Blockchain tem emergido como um meio alternativo para pagamentos em cenários de comércio eletrônico, criando um sistema descentralizado no qual não existe dependência de instituições confiáveis. Além disso, desde que foi originalmente proposto para uso em criptomoedas, diversas soluções surgiram que visam empregar a tecnologia de Blockchain em uma variedade de domínios de aplicação, como a criação de um cartório distribuído para armazenar contratos inteligentes. A despeito do domínio de aplicação específico, porém, soluções baseadas em Blockchain utilizam um conjunto de

diversas tecnologias bem conhecidas, como esquemas criptográficos, árvores de Merkle, redes P2P e mecanismos de consenso. Estes últimos são de particular interesse de pesquisa atualmente, em especial porque a maioria dos mecanismos de consenso usados nas primeiras criptomoedas (e.g., Bitcoin) exigem um poder computacional considerável para garantir a consistência do sistema. Com o objetivo de avaliar o estado da arte das pesquisas nesta área, o presente manuscrito apresenta os principais conceitos e classificações de Blockchain, dando ênfase nos mecanismo de consenso subjacentes e nos aspectos de segurança das soluções resultantes. Também é apresentado um breve experimento usando as soluções Ethereum e MultiChain para ilustrar os aspectos de segurança relativos aos consensos do tipo Tolerância Prática a Falhas Bizantinas (Practical Byzantine Fault Tolerance – PBFT) e Prova-de-Trabalho (Proof-of-Work – PoW).

3.1. Introdução

O grande volume de transações eletrônicas realizadas atualmente realça a sua importância nas relações comerciais modernas. Ao mesmo tempo, essa relevância lança desafios diversos, tais como a necessidade de desenvolvimento de tecnologias que forneçam maior eficiência e segurança nas transações realizadas. Neste sentido, o uso de soluções que sigam um paradigma centralizado (e.g., banco de dados tradicionais) muitas vezes tem sua eficiência e segurança questionadas, seja por gerar pontos centrais de falha ou por questões de escalabilidade e confiabilidade. Em parte devido a essas preocupações, surgiu recentemente um movimento cujo objetivo é elaborar novos conceitos e soluções baseados em abordagens não-centralizadas. Cabe notar que os bancos e instituições financeiras tradicionais, em sua grande maioria, são exemplos de centralização quando refere-se ao modo como as transações financeiras são realizadas: estas concentram a confiança durante as transações e centraliza o banco de dados com informações financeiras, criando assim uma relação de dependência por parte seus usuários [1, 2].

Um banco de dados tradicional geralmente usa uma arquitetura cliente-servidor, na qual as entradas feitas pelos clientes são armazenadas em um servidor centralizado e, eventualmente, replicadas. Nesse caso, o controle do banco de dados é mantido por uma autoridade central no lado do servidor, que regula o acesso e decide quais entidades têm permissão de leitura ou gravação. Em contraste, um Blockchain usa um modelo totalmente descentralizado, no qual cada participante (não necessariamente confiável) mantém, calcula e atualiza novas entradas que são replicadas em todos os nós do sistema. Os registros das transações formam então uma espécie de *livro razão distribuído* (usando o termo comumente empregado na área de contabilidade), que pode ser acessado por qualquer usuário para verificar a validade das transações realizadas, segundo regras definidas pela aplicação. É exatamente essa característica do Blockchain que é explorada no Bitcoin, a primeira criptomoeda criada com base nessa tecnologia e a responsável por trazer maior notoriedade ao conceito de Blockchain. Tal notoriedade deve-se ao fato do Bitcoin criar um cenário financeiro com grande autonomia, no qual transações comerciais eletrônicas não dependem do aval de instituições financeiras governamentais ou privadas reconhecidas, e ainda assim podem ser feitas com um bom grau de segurança [3]. Em particular, o Blockchain no Bitcoin previne tentativas de *double-spending* (i.e., "gasto-duplo"), em que um usuário desonesto tenta repassar um mesmo conjunto de moedas para mais de um usuário.

Embora o Blockchain tenha sido criado para permitir transações financeiras, e seja comumente associado a esse domínio, outras áreas de aplicação e uso de Blockchain estão surgindo com o crescimento da atenção do público em torno da tecnologia [4]. De fato, as tecnologias que possibilitam a implementação Blockchain têm sido objeto de um número crescente de pesquisas científicas, gerando interesse significativo por diversos setores da indústria, governamental e pesquisadores devido às suas características de transparência, confiança e segurança [5]. Como resultado, enquanto a primeira implementação popular do Blockchain (o Bitcoin) foi introduzida em 2008 [6], desde então vários sistemas Blockchain, como Ethereum [7], Hyperledger [8] e Multichain [9] emergiram com propostas fora do setor financeiro [3].

Embora os detalhes da implementação do Blockchain possam variar dependendo da aplicação alvo, a tecnologia em si se baseia em dois elementos subjacentes principais:

- Uma estrutura de dados com verificação de integridade: como o próprio nome indica, o Blockchain utiliza uma estrutura de dados baseada em encadeamento de blocos, em que cada bloco (um conjunto de registros quaisquer, acompanhados de metadados) carrega o valor de *hash* do bloco anterior. Ao assinar digitalmente os registros armazenados nos blocos individuais, o Blockchain permite a detecção de alterações locais e, indiretamente, também permite verificar a integridade dos blocos anteriores (cujos *hashes* seriam alterados no caso de modificações indevidas).
- Um mecanismo de consenso distribuído: o Blockchain utiliza técnicas para garantir a consistência do livro-razão, i.e., que todos os usuários eventualmente terão a mesma visão sobre a ordem em que os blocos foram inseridos na cadeia. Assim, garante-se que a característica distribuída do banco de dados não leve a decisões inconsistentes por parte dos usuários. Em particular, esse mecanismo é importante para evitar que uma mesma moeda seja gasta múltiplas vezes: contanto que todos os usuários concordem sobre qual foi a primeira transação de repasse dessa moeda, apenas essa primeira transação será considerada válida. Ao realizar esse procedimento, o Blockchain pode ser visto como uma Autoridade de Carimbo de Tempo (*Timestamp Authority* – TA) distribuída, que define a relação temporal entre bloco inserido no sistema. É importante notar, entretanto, que essa relação temporal é: (1) apenas relativa, i.e., não se busca determinar os instantes de tempo exatos em que uma transação ocorreu; e (2) não necessariamente corresponde aos instantes de tempo reais, de modo que uma transação que ocorra no instante de tempo t pode aparecer no Blockchain depois de uma transação que ocorra posteriormente, em um instante $t + \varepsilon$. Portanto, o Blockchain cria uma espécie de "linha de tempo alternativa", que não necessariamente corresponde à ordem temporal real das transações.

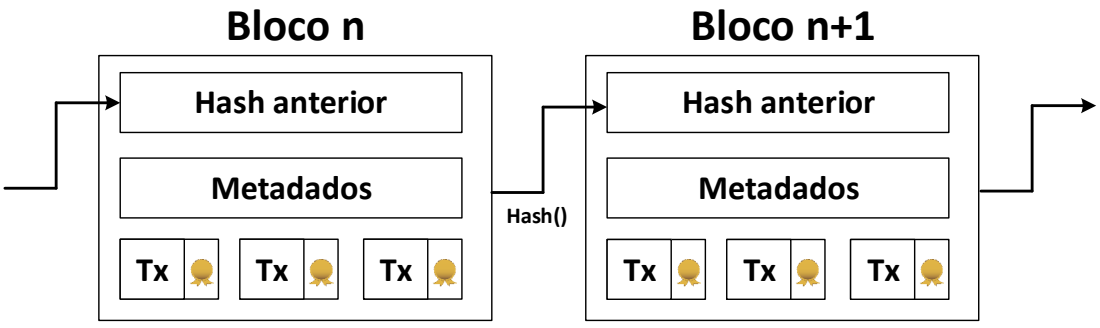
Cada um desses elementos básicos é discutido mais profundamente nas próximas seções. Antes disso, entretanto, é importante ressaltar que o Blockchain *per se* não garante a validade dos dados subjacentes nele introduzidos, mas apenas que o conteúdo e a ordem dos blocos contendo esses dados não podem ser alterados de forma imperceptível. Assim, a verificação da validade dos dados fica à cargo da camada de aplicação, que deve definir as regras para que os dados inseridos nos blocos sejam considerados "válidos". No caso do Bitcoin, a regra básica é que (1) as moedas sendo transferidas ainda estejam na

posse do usuário origem da transferência (i.e., não exista uma transação anterior em que a moeda tenha sido gasta), e (2) a transferência seja digitalmente assinada pela origem, garantindo sua anuência com relação à transferência. Para outras aplicações, as regras em questão devem ser definidas conforme a necessidade. Por exemplo, em aplicações cujo objetivo é permitir a troca de ativos genéricos entre usuários (i.e., não apenas ativos financeiros virtuais), dispensando entidades confiáveis e processos cartoriais, é comum o uso de regras similares às do Bitcoin para verificação da posse do ativo e da assinatura do seu dono na transação. Entretanto, no caso de ativos do mundo real, normalmente é também necessário "virtualizá-lo", ou seja, identificá-lo de forma unívoca e verificável tanto no mundo digital e real. Isso normalmente passa pela criação de um *identificador virtual* que garanta a própria existência do ativo e sua descrição; alguns exemplos de ativos cuja virtualização é possível incluem imóveis, que podem ser identificados pelo seu número de matrícula, e veículos, que podem ser identificados pelo seu número de chassis. Cabe notar que, embora esse processo de identificação possa envolver um entidade confiável (e.g., um cartório), as transações posteriores de transferência desse ativo podem ser feitas sem a interveniência dessa entidade, bastando incluir o identificador do ativo entre os dados da transação. Transações fraudulentas ainda podem ocorrer, porém o histórico de eventos permite identificar eventuais transgressões.

3.1.1. A estrutura de dados do Blockchain

A estrutura de dados subjacente a esquemas de Blockchain consiste em uma sequência de blocos, como ilustrado na Figura 3.1. Nesta sequência, fica armazenada uma lista completa de registros, criando um conjunto de dados que pode ser consultado por qualquer usuário do sistema [6]. A semântica de cada registro é definida pela aplicação em si, dependendo das necessidades do cenário alvo, cabendo ao Blockchain apenas armazenar esses registros com garantias de integridade dos blocos e de sua ordem relativa no sistema. No entanto, para permitir uma discussão mais focada e sem perda de generalidade, ao longo deste documento assume-se que os registros referem-se a transações de ativos entre diferentes partes. A escolha desse cenário deve-se ao fato de este ser um cenário típico no qual é necessária a ordenação de registros (no caso, para determinar o proprietário atual), que é a finalidade básica de um Blockchain.

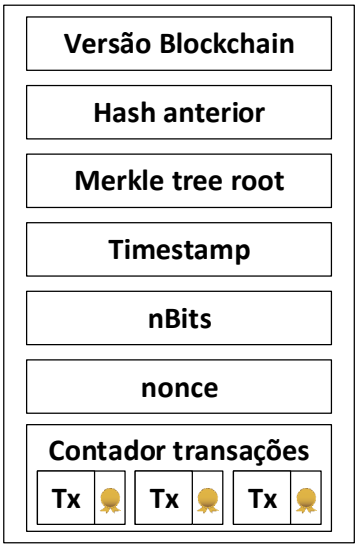
Figura 3.1: Blockchain: exemplo de sequência de blocos.



Os principais elementos de cada bloco em um Blockchain, que também são mostrados na Figura 3.1, são: (1) o valor do *hash* do conteúdo completo do bloco anterior, criando uma ordenação entre eles e permitindo a verificação da integridade de cada bloco e também de sua ordem relativa; e os dados da aplicação em si, representados na Figura 3.1 como uma lista de transações Tx, e que comumente são assinados digitalmente pelo usuário responsável por sua geração, ou mesmo por todas as partes envolvidas na transação. Como metadados adicionais, o bloco completo costuma incluir alguma forma de identificação da(s) entidade(s) que fizeram a seleção e verificação dos dados armazenados no bloco, entidades estas conhecidas como *validadores* ou *mineradores*[1]. Essa identificação costuma se dar na forma da chave pública do(s) minerador(es), que pode ser acompanhada ou não da assinatura digital do bloco completo; no Bitcoin, por exemplo, a primeira transação de qualquer bloco, chamada de *transação base* (do inglês *coinbase transaction*) consiste basicamente na chave pública do minerador, a qual não acompanha sua assinatura digital. Outros metadados vão depender da aplicação específica do Blockchain. Por exemplo, algumas soluções incluem um carimbo de tempo (*timestamp*) com a data e a hora da criação do bloco; entretanto, essa informação só deve ser considerada para fins de referência, pois o baixo grau de sincronismo normalmente apresentado por uma rede Blockchain impediria o registro de um instante exato e confiável de tempo.

A Figura 3.2 exemplifica a composição de um bloco genérico no Bitcoin. Em suma, os seguintes campos estão presentes nos blocos em adição às transações escolhidas pelos mineradores [10]:

Figura 3.2: Estrutura básico de um bloco no Bitcoin.



- Versão Blockchain: Consiste na versão da estrutura do bloco.
- *Hash Anterior*: O valor do *hash* do bloco anterior, calculado usando uma função de *hash* segura (no caso do Bitcoin, SHA-256 [11]).

[1]O termo "minerador" deriva do processo conhecido no Bitcoin como *mineração*, que consiste em realizar o esforço computacional necessário para verificar os dados a serem colocados no bloco, conforme discutido mais profundamente na Subseção 3.1.2

- *Merkle Tree Root*: O valor de *hash* correspondente à raiz de uma árvore de Merkle [12] construída a partir de todas as transações incluídas neste bloco.

- *Timestamp*: Data e hora de criação do bloco.

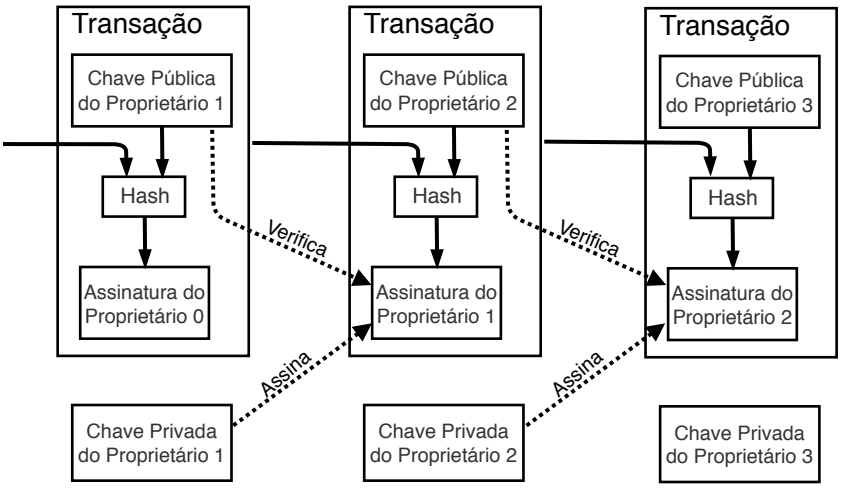
- *nBits*: Valor que representa a dificuldade computacional para mineração desse bloco. Este campo está diretamente relacionado com o mecanismo de consenso utilizado pelo Bitcoin, conhecido como *prova de trabalho* (*Proof of Work* (PoW)), conforme discutido mais adiante na Subseção 3.1.2.

- *Nonce*: Valor arbitrário adicionado ao bloco para dar variabilidade ao valor do *hash* do bloco. Como o campo "nBits", este campo também está relacionado ao mecanismo de consenso via PoW.

Quando uma nova rede de Blockchain é desenvolvida, um bloco conhecido como *Gênesis* ou Bloco Zero é criado. O Bloco Zero é um bloco diferente dos demais, pois, pelo fato de ser o primeiro bloco da cadeia, ele não possui *hash* anterior. Além disso, o Bloco Zero pode conter informações diferentes de acordo com a implementação do Blockchain (e.g., dados de configuração).

Com essa estrutura baseada em encadeamento de *hashes*, o Blockchain garante a integridade e da sua ordem relativa dos blocos. Aliado à assinatura digital dos registros armazenados, pode-se também verificar a autenticidade dos dados de forma distribuída. Em particular, quando esses dados assumem a forma de transações de ativos entre usuários, pode-se identificar o atual proprietário de um ativo qualquer simplesmente analisando a cadeia de transações válidas realizadas. Isso é ilustrado na Figura 3.3, que mostra como diferentes transações alteram o proprietário do objeto da transação (e.g., uma criptomoeda): ao assinar uma transação, o dono do ativo indica sua anuência em transmití-lo ao novo proprietário; o último usuário identificado como receptor do ativo é então seu atual proprietário.

Figura 3.3: Exemplo de aplicação do Blockchain: ordem relativa das transações revela o atual proprietário do ativo que é objeto dessas transações.



3.1.2. Mecanismo de Consenso

Todos os nós no Blockchain detêm de forma independente suas próprias cópias dos blocos que o compõem, e cada registro nele contido é verificado individualmente conforme regras definidas pela camada de aplicação [10]. Após a verificação de um bloco, o nó do sistema envia uma cópia desses blocos para outros usuários da rede, mecanismo que se repete até que todos os nós recebam a nova informação.

Como esse processo é feito de forma assíncrona e sem coordenação central, cada nó pode ter uma visão diferente do estado do Blockchain em um dado instante. Entretanto, para garantir a convergência dessas visões em um espaço de tempo não muito longo, o Blockchain utiliza um *mecanismo de consenso*. Isso é essencial para criar um sistema consistente, no qual todos os nós concordem com a ordem dos blocos e sobre os seus conteúdos. Em uma aplicação voltada a troca de ativos, por exemplo, é isso que garante que todos os usuários têm a mesma visão sobre quem é o proprietário de um determinado ativo, evitando tentativas de gasto duplo. Nesse cenário, o mecanismo de consenso permite construir um ambiente bastante resiliente a tentativas de violação, no qual as transações envolvendo qualquer ativo digital são verificadas por uma gama de participantes não mutuamente confiáveis [1].

É interessante notar que a questão do consenso não surgiu com o Blockchain. Na realidade, trata-se de um problema computacional bastante antigo na área de computação distribuída, que deve considerar garantias de acordos de maneiras democráticas, acordos estes que garantem a confiabilidade do que é acordado e distribuído [13]. Em qualquer cenário, incluindo no Blockchain, o consenso tem como função a aplicação/verificação de um conjunto regras estabelecidas por um conjunto de participantes atuando de forma organizada. Assim, a especificação de um esquema de consenso envolve, além da organização dos nós e da definição de suas regras de atuação, a própria infraestrutura sobre a qual o Blockchain opera, (e.g., a forma como as mensagens são trocadas, a organização da rede e os algoritmos empregados). Alguns dos principais subcomponentes que formam mecanismos de consenso são, portanto [2]:

- Topologia de rede de consenso. Está relacionada ao nível de descentralização no processo de validação e também a fatores como o mecanismo de recompensa utilizado. As principais topologias adotadas no Blockchain são similares às adotadas em outras redes P2P, a saber: descentralizada, na qual todos os nós têm as mesmas responsabilidades; parcialmente centralizada, em que assume-se a aparição dinâmica de "super-nós", que assumem responsabilidades adicionais na rede; hierárquica, na qual os nós se organizam de forma bastante rígida, seguindo uma relação de responsabilidades bem definida; e centralizada, que envolve alguma entidade (parcialmente) confiável, como um servidor.

- Comunicação e troca de mensagens. Os Blockchains também são sistemas de armazenamento redundantes e descentralizados. Esta redundância torna difícil sequestrar/roubar as informações armazenadas neles. Como essa informação trafega através de redes e normalmente não há uma autoridade de coordenação central, cada nó deve transmitir as informações que possui (e.g., novos blocos ou mesmo o Blockchain completo) para outros nós que sabe-se estarem participando do sistema.

Para isso, os nós possuem uma lista (não necessariamente completa) dos demais nós da rede. Sempre que um novo bloco é adicionado à versão local do Blockchain de um nó, este nó passa o bloco para outros em sua lista de nós por meio de protocolos de propagação de mensagens via *broadcast*, como é o caso de mecanismos de fofoca (*gossip*) [14]. Já em redes nas quais os nós são conhecidos, pode-se estabelecer um conjunto de nós que ficam responsáveis exatamente pelo repasse de mensagens, como acontece em soluções como Ripple [15]. Há então dois tipos de trocas de mensagens [2]:

  - Local: A troca de mensagens ocorre primeiro entre nós vizinhos, através de um processo de validação local, e posteriormente se propaga pela rede até que o consenso global seja alcançado. Esse tipo de mecanismo também é chamado de "consenso federado". Um exemplo de sistema que usa esse processo é o Ripple, no qual os nós podem compartilhar registros de transações entre si e chegar a um consenso sem conhecer diretamente todos os nós da rede.
  - Global: Esse é o tipo mais comum na maioria das implementações de Blockchain (e.g., Bitcoin, Ethereum, etc.). Neste caso, a comunicação ocorre em uma lista de nós selecionados, conhecidos na rede Bitcoin como nós de *fallback*, os quais são responsáveis por manter uma lista de todos os nós na rede. Na conexão de um novo nó, os nós de *fallback* enviam uma lista de nós escolhidos aleatoriamente para o participante. A topologia de rede resultante carece de um conceito de proximidade ou vizinhança local.

- Acordo para o consenso e resolução de conflitos. Define um conjunto de regras sob as quais os registros (como conjuntos de transações ou qualquer outra parte atômica) são atualizados independentemente pelos nós de um sistema distribuído. Isso é importante para entender como um sistema distribuído é capaz de lidar com as chamadas falhas bizantinas, i.e., falhas que podem apresentar-se de forma diferente para observadores distintos, devido à ausência de uma visão global da rede, criando então inconsistências no sistema. Dependendo do cenário, esse tipo de falha por ser tratado por meio de comunicações síncronas ou assíncronas, aliadas a um mecanismo de votação para decidir conflitos. Uma vez estabelecido um consenso na rede, ele pode ser considerado [2]: determinístico, i.e., a informação armazenada em um Blockchain após o consenso não pode ser alterada posteriormente; ou não-determinístico, nos quais o estabelecimento de um consenso em certo momento não impede que o estado do Blockchain seja alterado por um consenso posterior (embora possa reduzir a probabilidade de tais modificações). Para um cenário altamente distribuído, como é o caso da maioria dos sistemas baseados em Blockchain, o consenso determinístico é bastante difícil de alcançar. Por exemplo, no Bitcoin os novos blocos se difundem por meio de mensagens assíncronas e os nós têm por regra armazenar localmente a versão do Blockchain que apresentar o maior número de blocos (a chamada "regra da cadeia mais longa"); nesse caso, mesmo que seja alcançado um consenso global sobre o estado do Blockchain, nada impede que um conjunto de novos nós com poder computacional suficiente entre no sistema e anule o consenso anterior oferecendo versões mais longas da cadeia de blocos.

- Algoritmo para inserção de blocos. Um Blockchain, como um tipo especial de sistema distribuído, é considerado tolerante a falhas devido à replicação de seu estado por todos os nós. Assim, mesmo que alguns nós deixem a rede (e.g., devido a um desligamento acidental ou proposital), a aplicação baseada em Blockchain tem a capacidade de continuar funcionando, i.e., é capaz de manter o mesmo grau de confiabilidade e validade das informações armazenadas nos seus registros que apresentava antes da partida daquele nó. De fato, Blockchains representam uma solução descentralizada para o armazenamento de informações nos quais há elevado grau de redundância, já que cada nó do sistema possui uma réplica do livro razão. Para que haja consistência entre os dados armazenados pelos diferentes nós, é essencial que sejam definidas regras para a atualização desses dados de forma distribuída. Caso contrário, se qualquer nó puder a qualquer momento atualizar sua versão local e disseminá-la para outros nós, dificilmente haveria uma convergência entre as diversas versões do Blockchain espalhadas pela rede. Nos últimos anos, a evolução das tecnologias de Blockchain foi acompanhada pelo desenvolvimento de diferentes mecanismos de regulação da inserção de blocos no Blockchain que ajudam a manter a consistência das informações contidas no livro razão. Alguns dos principais são:

  - *Proof of Work* (PoW): Introduzido no Bitcoin [6], foi o primeiro e ainda é provavelmente o mais conhecido mecanismo de consenso distribuído no cenário do Blockchain. No PoW todos computadores da rede são encarregados de manter a segurança do Blockchain, e comumente o fazem em troca de gratificações. Tecnicamente, a tarefa desempenhada pelos mineradores consiste em encontrar uma entrada cujo valor de *hash* seja menor que um determinado alvo. Essa é uma tarefa computacional extremamente repetitiva e que envolve um alto custo computacional, de modo que comumente é realizada por meio de hardware especializado. Apesar desse mecanismo de consenso fazer com que seja custoso adicionar novos blocos no Blockchain, é simples a verificação dos blocos inseridos (basta o cálculo de uma única função de *hash*), e o esquema é bastante eficiente para combater mineradores maliciosos.
  - *Practical Byzantine Fault Tolerance* (PBFT): Trata-se de um algoritmo de replicação originalmente criado para permitir a qualquer sistema tolerar falhas bizantinas. Os nós se organizam para operar em rodadas, de modo que em cada rodada um nó primário é selecionado de acordo com certas regras. O nó primário fica então responsável por inserir o próximo bloco na cadeia. O processo é dividido em três fases: Pré-Preparado, Preparado e Comprometido. Para passar de uma fase a outra, um nó precisa receber o voto de 2/3 de todos os nós. Para que isso seja possível, portanto, exige-se que o número total de nós seja conhecidos pela rede. Não há mecanismos custosos computacionalmente nesse caso, bastando a consulta entre nós para se chegar a um consenso [16].
  - *Proof of Authority* (PoA): Os participantes não são solicitados a resolver problemas matemáticos arbitrariamente difíceis, mas devem usar um conjunto de autoridades configuradas para colaborar sem confiança. Como essas autoridades são nós com permissão especial para criar novos blocos e proteger o

Blockchain, normalmente os mecanismos PoA não são utilizados em redes públicas, nas quais tal confiança não existe. Por outro lado, esse mecanismo se encaixa bem em redes privadas do Modelo Consórcio, nas quais algumas entidades reais pré-selecionadas atuam como autoridades. Para prevenir a personificação desses nós especiais por outros nós, as autoridades recebem certificados digitais válidos para assinar os novos blocos. Assim, cada bloco (ou cabeçalho) que um cliente vê pode ser comparado com uma lista de signatários confiáveis [2].

- *Proof of Burn* (PoB): Os mineradores devem provar que "queimaram" alguns ativos digitais (e.g., criptomoedas). Eles fazem isso enviando os ativos em questão para um endereço especial do Blockchain, ao qual não é associada a capacidade de repassar esses ativos. Embora o PoB tenha sido criado com o objetivo de minimizar o desperdício de recursos gerados pelo PoW, atualmente todos os mecanismos do PoB funcionam exatamente pela queima de moedas digitais mineiradas pelo PoW. Portanto, o mecanismo acaba sendo caro, tendo em vista que as moedas digitais usadas como combustível para "queima" em um sistema PoB não podem ser recuperadas [17].
- *Proof of Capacity* (PoC): também conhecida como *Proof of Space*, *PoSpace* ou *PoStorage*, esse mecanismo é baseado no conceito de espaço como recurso (*Storage as a Resource* – SaaR). O foco desse mecanismo não está nos ciclos da CPU, mas na quantidade de armazenamento real não volátil (e.g., disco rígido e SSD) que o minerador deve empregar para ganhar o direito de inserir um novo bloco no Blockchain. Em outras palavras, os nós devem alocar um volume significativo de espaço em memória secundária para a mineração, em vez de usar a CPU como no PoW. Mineradores que dedicam mais espaço de armazenamento têm uma expectativa proporcionalmente maior de minerar com sucesso um bloco e receber a recompensa correspondente [18, 19].
- *Proof of Stake* (PoS): É uma abordagem alternativa ao PoW que busca reduzir o custo de energia do processo de mineração, bem como a resultante dependência de hardware especializado para fazê-lo. Basicamente, esse tipo de mecanismo dá preferência a mineradores que tenham maior participação na rede e, portanto, tenham mais a perder no caso de fraudes. Por exemplo, em um sistema de criptomoedas, seria dada preferência aos mineradores que detenham mais moedas no momento da inserção do bloco [19].
- *Proof of Importance* (PoI): Implementado pela companhia NEM, este tipo de mecanismo leva em consideração a importância dada um minerador na rede, com base no número de moedas que ele possui, no número de transações por ele realizadas, e na sua reputação dentro da rede [19]. Esse tipo de mecanismo pode, portanto, ser visto como uma extensão do conceito de PoS.
- *Delegated Proof of Stake* (DPoS): Este mecanismo basicamente elege uma lista de nós que terão a oportunidade de participar do bloco de novas transações e adicioná-los ao Blockchain. A prova de participação delegada tem a função de incluir todos os detentores de moedas, mesmo não recompensando da mesma maneira que a PoS. A ideia central é dar maior responsabilidade e importância para os titulares da rede [19].

- *Leased Proof of Stake* (LPoS): Uma variante do PoS que fornece uma possibilidade de mineradores pequenos gerarem lucros. Basicamente, este mecanismo permite que os nós com maior participação na rede (e, portanto, maior chance de validarem blocos) sejam alugados por outro nós [19].
- *Ripple*: O Ripple utiliza um protocolo próprio, denominado XRP, para que opere em sub-redes confiáveis coletivamente dentro da rede maior. Na rede, os nós são divididos em dois tipos: Servidor, que participa do processo de consenso; e cliente, que apenas transfere. Na rede Ripple, os servidores de validação trocam informação sobre os registros que precisam ser incluídos no Blockchain; o sub-conjunto de transações inserido no próximo bloco corresponde àquelas que receberam o voto positivo de uma quantidade mínima (e ajustável) de validadores. Para facilitar o consenso, cada validador pode definir um conjunto de outros validadores nos quais ele confia, de modo que seus votos dão preferência às transações propostas por nós em sua rede de confiança [15].
- *Tendermint*: Semelhante ao PBFT, esse mecanismo segue o mesmo padrão de segurança. A diferença é que no Tendermint exige que os validadores bloqueiem suas moedas enquanto participam do processo de validação, e tentativas de fraudar o sistema (e.g., criar situações de gasto duplo) são punidas pelos nós da rede (e.g., com a perda de moedas) [20].

As características de cada mecanismo de consenso devem ser considerados em conjunto ao projetar um processo ativo de validação de consenso de rede, porque não apenas sua configuração individual, mas também sua combinação determinam quando e como o consenso no Blockchain é alcançado e o livro razão é atualizado. A Tabela 3.1 mostra uma comparação entre alguns dos diferentes mecanismos de consenso, classificados em três funcionalidades: Gerenciamento do nó, Economia de Energia e Controle do Nó.

Tabela 3.1: Tabela de Comparação entre mecanismos de consenso Blockchain.

Funcionalidades	**Gerenciamento do Nó**	**Economia de Energia**	**Controle do Nó**
PoW	Aberto	Não	25% do poder *hash*
PoS	Aberto	Parcial	51% *stake*
DPoS	Aberto	Parcial	51% mineradores
LPoS	Aberto	Parcial	51% *stake*
PBFT	Permissão	Sim	33.3% de falhas
PoI	Aberto	Sim	51% mineradores
PoA	Permissão	Parcial	51% mineradores
PoB	Aberto	Parcial	51% mineradores
PoC	Aberto	Sim	51% *Stake*
Ripple	Aberto	Sim	20% falhas UNL
Tendermint	Permissão	Sim	33.3% poder de voto

O *Gerenciamento do nó* refere-se a quem pode participar do mecanismos de consenso, podendo ser aberto a qualquer pessoa ou necessitar de autorização para fazer parte

do Blockchain. A *Economia de energia* informa se a concepção do mecanismo de consenso tem em suas finalidades economizar energia, aspecto que tem ganhado importância em particular depois que o custo de mineração via PoW do Bitcoin começou a elevar-se demasiadamente. Por fim, o *Controle do nó* refere-se às condições para que o processo de consenso possa ser comprometido de algum modo indevido (e.g., manipulação, mineração egoísta, etc.).

Diferentes mecanismos de consenso são apresentados na Tabela 3.1 e para cada um destes há diferentes contextos de aplicação, benefícios e potenciais dificuldades. Para cada versão de Blockchain existem diferentes métodos de aplicação destes mecanismos. Por exemplo, embora ambas sejam plataformas de gestão de contratos inteligentes, a *Bitshares* adota DPoS, enquanto o *Ethereum* utiliza PoS. De forma geral, os mecanismos PoW e PoS são mais os mais utilizados para Blockchain do tipo Público, enquanto Blockchains de Consórcio e Privado têm preferência pelos mecanismos PBFT e DPoS, que consomem menos energia, ou então PoW devido ao grau de segurança resultante.

3.1.3. Taxonomia de Modelos Blockchain

Atualmente, os sistemas que implementam o conceito de Blockchain podem ser classificados em três modelos principais quanto ao seu acesso [5, 21]:

- Blockchain Público: é o modelo mais conhecido. Nesse tipo de Blockchain, qualquer pessoa pode ler, enviar, ou validar transações, além e também poder participar do processo de consenso distribuído. Esses Blockchains são considerados totalmente descentralizados, pois a influência que uma pessoa tem no processo de consenso depende apenas de seus méritos em comparação com outros nós na rede (e.g., no caso de Blockchains baseadas em PoW, sua capacidade de mineração é proporcional ao seu poder computacional). Assim, pode-se dizer que o processo de consenso em Blockchains públicas é bastante democrático.

- Blockchain Consórcio: composto por duas ou mais instituições. Este modelo é basicamente controlado e modelado por instituições parceiras, que podem alterar as regras de acesso ao Blockchain conforme seus interesses e necessidades. Este modelo é considerado parcialmente descentralizado, pois não é controlado por apenas uma parte, mas sim por um conjunto pré-definido de instituições.

- Blockchain Privado: é um modelo mais conservador, no qual apenas uma instituição detém o controle sobre as operações. Seu modelo de funcionamento pode mudar de acordo com a necessidade e interesse desta instituição, de forma monocrática. Esse tipo de Blockchain é considerado centralizado e de forma geral é utilizado para auditoria, gerenciamento de banco de dados e outras informações que necessitam de um nível diferenciado de controle sobre os dados armazenados.

A Tabela 3.2 lista alguns critérios de comparação entre os modelos de Blockchain: Público, Consórcio e Privado. Estes modelos são avaliados de diferentes formas, baseados nos conceitos da tecnologia Blockchain [22]:

- Consenso distribuído: Se todos os nós podem participar do processo de consenso, ou apenas nós pré-determinados.

- Permissão de Verificação: Se as transações podem ser verificadas de modo público ou restritamente.
- Imutabilidade: O Blockchain, por princípio, cria uma estrutura de dados imutável após um consenso amplo ser atingido. Contudo, algumas implementações podem possibilitar alterações nos dados armazenados, que geralmente ocorrem somente por processos de consenso distribuído adicionais ou por meio da atuação de uma entidade confiável no sistema.
- Centralização: refere-se ao grau de controle de alguma instituição sobre o Blockchain.
- Processo de consenso: Se qualquer entidade pode participar do processo de consenso ou apenas entidades certificadas ou pré-selecionadas.

Tabela 3.2: Tabela de Comparação modelos de acesso do Blockchain.

	Blockchain Público	**Blockchain Consórcio**	**Blockchain Privado**
Consenso Distribuído	Todos Mineradores	Mineradores Selecionados	Mineradores Selecionados
Permissão de Verificação	Pública	Restrita	Restrita
Imutabilidade	Sim	Adulterável	Adulterável
Centralização	Descentralizado	Parcial	Centralizado
Processo de Consenso	Todos Mineradores	Mineradores Selecionados	Mineradores Selecionados

3.1.4. Versões Blockchain

Embora os mecanismos subjacentes ao Blockchain existam há muito tempo na literatura, pode-se afirmar que a tecnologia de Blockchain como utilizada atualmente é razoavelmente recente (surgiu em 2008 com o Bitcoin). Esse curto período de existência não impediu, entretanto, que os conceitos por trás da tecnologia evoluíssem. De fato, alguns autores identificam diferentes eras ou estágios de evolução do Blockchain, que, embora nem sempre tenham sido formalmente definidos, apresentam algumas características marcantes que os diferenciam [1]. Essas características são discutidas nas próximas seções.

3.1.4.1. Blockchain 1.0

A primeira versão do Blockchain muitas vezes é confundida com a própria aplicação que faz uso dessa tecnologia, o Bitcoin. O Bitcoin é uma solução para um problema antigo com dinheiro digital, que é o gasto duplo. Até o surgimento do Bitcoin, soluções envolvendo dinheiro digital tratavam esse tipo de dado como um ativo digital comum, que poderia ser copiado diversas vezes como qualquer arquivo ou mensagem. Portanto, sem as devidas precauções, transações sobre um mesmo ativo financeiro poderiam ser autorizadas sem a devida verificação prévia de sua existência ou de seu proprietário. A solução praticada para resolver esse problema consistia basicamente na delegação de confiança a uma terceira parte (e.g., bancos, Paypal, etc.), que ofertava o serviço de controle do ativo financeiro. Esse terceiro confiável seria responsável por garantir as identidades das

partes envolvidas na transação e o cumprimento das regras nela envolvidas (e.g., o ativo seria transferido após a conclusão da transação), além de prevenir a ocorrência gastos duplos [1].

Nesse cenário, o Blockchain 1.0 moldou-se como uma moeda para a Internet, criando o sistema de pagamento digital que é a primeira aplicação do Blockchain. A principal funcionalidade das moedas baseadas em Blockchain é de que qualquer transação pode ser realizada sem a necessidade da confiança entre as partes envolvidas na transação, de modo completamente descentralizado, distribuído e global. Esta habilidade cria possibilidades muito além de moedas e pagamentos, e que são aproveitadas no Blockchain 2.0.

3.1.4.2. Blockchain 2.0

Após o uso inicial como solução para dinheiro digital, o Blockchain foi visto como uma tecnologia que pode ser explorada em outros setores da tecnologia. Durante conversas entre os desenvolvedores originais do Bitcoin, foi indicado que o Blockchain suportaria uma variedade de tipos de transações possíveis, não apenas criptomoedas. Estas transações podem ser, por exemplo, contratos alfandegários, arbitragens de terceiros, transações de custódia e outros tipos de modelos.

É interessante notar que essa expansão no uso da tecnologia Blockchain foi motivada especialmente pela elevada escala assumida pelo Bitcoin, e pela robustez observada no seu mecanismo de consenso a despeito dos custos energéticos envolvidos [1]. Outro fator importante foi o modelo baseado em incentivos para os usuários que contribuem com a rede (os mineradores), algo essencial em qualquer tecnologia P2P. Por outro lado, conforme previamente mencionado, muitas aplicações são baseadas em uma concepção equivocada sobre (1) quais serviços de fato são fornecidos por um Blockchain, (2) quais serviços poderiam ser obtidos usando simplesmente os mecanismos subjacentes a ele (e.g., assinaturas digitais), e (3) quais serviços não são fornecidos. Por exemplo, não é incomum que aplicações nas quais é necessário verificar apenas a *existência de registros*, mas não importa sua ordem relativa, utilize um Blockchain quando a simples assinatura digital desses registros bastaria. Em outros casos, assume-se que o mecanismo de consenso utilizado no Blockchain é suficiente para decidir sobre a validade dos dados registrados, quando na realidade o consenso apenas decide sobre a *ordem relativa* dos registros, deixando a definição das regras de validação dos dados a cargo da camada de aplicação construída sobre o Blockchain.

De qualquer forma, o Blockchain 2.0 foi elaborado tomando proveito do conceito de transação descentralizadas subjacentes à tecnologia, podendo ser usado para registrar, confirmar e transferir todos os tipos de contratos e patrimônio. Sem entrar no mérito da real aplicabilidade de um sistema de Blockchain a cada aplicação específica, alguns exemplos de contratos e patrimônios suportados pelo Blockchain 2.0 são [1]:

- Geral: Garantia de transações, contratos coligados, arbitragem por terceiros, aplicação de multas.
- Transações Financeiras: estoque, captação própria, financiamento colaborativo,

cartas de confiança, fundos mútuos, pensões, derivados, anuidades.

- Registros públicos: escritura de terra e patrimônio, registro de veículos, licença de comércio, certidão de casamento, certidão de óbito.
- Identificações: carteira de motorista, carteira de identidade, passaporte, título eleitoral.
- Registros privados: empréstimos, contratos, apostas, assinaturas, garantias, testamento.
- Atestados: casa, quartos de hotel, carros alugados, acesso automotivo.
- Chaves de ativos físicos: casa, quartos de hotel, carros alugados, acesso automotivo.
- Ativos intangíveis: patente, marca comercial, direito autoral, nomes de domínio.

3.1.4.3. Blockchain 3.0 e além

Enquanto o uso inicial do Blockchain se concentrava no registro de ativos financeiros (criptomoedas) e soluções seguintes foram voltadas ao registro de ativos diversos, as propostas atuais costumam ter como foco as aplicações que podem ser construídas com base nesses registros. A maioria dessas aplicações é voltada à construção de sistemas altamente descentralizados, fornecendo um bom grau de verificabilidade mesmo na ausência de entidades confiáveis (ou, mais precisamente, buscando substituir tais entidades).

Um exemplo nesse sentido é o *Namecoin: Domain Name Systen (DNS) Descentralizado* [23]. Lançado entre 2011, o Namecoin usa um sistema de monetização de trabalho computacional similar ao do Bitcoin, mas em vez de criar uma moeda "genérica", seu objetivo primário é servir como um repositório para permitir a verificação de registros DNS. A aplicação é um uso não-usual da tecnologia Blockchain para uma aplicação de alcance e interesse globais, mas que diferentemente da Internet tradicional não pode ser controlada por qualquer corporação ou governo. Assim, de modo similar a outros sistemas de armazenamento de dados P2P (e.g., Freenet [24]) o *Namecoin* visa criar um sistema resistente à censura ou repressão, no qual usuários consigam publicar informações livremente na Internet. Essa é uma das motivações pelas quais o *InterPlanetary File System* (IPFS) [25], um sistema de armazenamento descentralizado de conteúdo Web sem a necessidade de um servidor de hospedagem, que propõe o uso de do *Namecoin* como uma das formas de aumentar seu grau de descentralização.

Outro exemplo interessante é o BitID [26], que fornece um serviço de verificação de identidade de indivíduos com base no seu identificador no Blockchain do Bitcoin.

Embora emblemáticos, esses exemplos estão longe de dar a dimensão completa do que é atualmente classificado como Blockchain 3.0. Afinal, esta fase é marcada por criar um ambiente bastante rico de aplicações distribuídas, para propósitos como proteção de propriedade intelectual, rastreamento de cadeia de suprimentos/*supply chain*, pagamentos internacionais, internet das coisas, e privacidade de pacientes no tratamento médico [5].

Ironicamente, embora várias propostas nessas áreas se definam como parte do "Blockchain 3.0", muitas delas o fazem apenas por serem baseadas em redes *Peer-to-Peer* (P2P), embora não envolvam de fato princípios fundamentais que caracterizam o Blockchain original, como encadeamento de blocos ou mecanismos de consenso. Independentemente de eventuais discussões sobre uso indevido do termo "Blockchain" nesses casos, o fato é que tais esforços têm levado a uma grande expansão no número e na variedade de aplicações descentralizadas na Internet. Exatamente por isso, um requisito importante para a evolução das tecnologias do Blockchain 3.0 é a integração efetiva não só entre diversas plataformas, mas também na indústria com ferramentas e tecnologias existentes. Por exemplo, propostas como Polkadot [27] e Aelf [28] têm como objetivo é realizar a integração entre diversas Blockchains de fins específicos. Os desafios envolvidos envolvem questões de padronização entre interfaces de comunicação e estruturas de dados, fatores determinantes para a interoperabilidade entre soluções.

3.1.4.4. Comparação entre as versões do Blockchain

Com a apresentação das três versões do Blockchain, se faz necessário um comparativo destas versões da tecnologia. A Tabela 3.3 realiza esta comparação apresentando: funcionalidades, abrangência de consenso distribuído e suas principais aplicações.

Tabela 3.3: Comparativo básico entre as versões do Blockchain.

	Blockchain 1.0	**Blockchain 2.0**	**Blockchain 3.0**
Funcionalidades	Criptomoedas	*Smart Contracts*	Aplicativos Descentralizados
PoW	Sim	Sim	Sim
DPoS	Não	Sim	Sim
PoS	Sim	Sim	Sim
PBFT	Não	Sim	Sim
LPoS	Não	Sim	Sim
PoI	Não	Sim	Sim
PoA	Não	Sim	Sim
PoB	Sim	Sim	Sim
PoC	Não	Sim	Sim
Ripple	Não	Sim	Sim
Tendermint	Não	Sim	Sim
Aplicações	Bitcoin	Ethereum,MasterCoin, *Open assets*, *Colored Coins*	Computação Descentralizada, Armazenamento Descentralizado

Observa-se a partir da Tabela 3.3 a evolução de tecnologias baseadas em Blockchain através de suas diferentes versões. Inicialmente, partiu-se de uma solução para o problema de centralização das instituições financeiras, evoluindo para o desenvolvimento e utilização de contratos inteligentes, e por fim para aplicações de computação e armazenamento descentralizados, que envolve diferentes setores da indústria.

Outro ponto visível, é que foi percebido o elevado gasto energético para a mineração dos blocos de Blockchain, além de sua lentidão e vazão, a partir deste pretexto foram desenvolvidos outros mecanismos além do PoW [29]. O mecanismo de consenso DPoS basicamente aproveita o poder da votação de aprovação de partes interessadas delegadas para resolver problemas de consenso e validar o Blockchain em um modelo com designs semelhantes aos sistemas democráticos, consequentemente utiliza um menor gasto energético que o o modelo PoW e os usuários votam proporcionalmente ao que produzem [30].

Por fim, percebe-se que dentre todos mecanismos que são propostos cada qual tem seu benefício em comparação um com os outros, mas o PoW é o mecanismo que, apesar de ter um custo elevado, é o que mais se encaixa no quesito de segurança e procedência na aplicação em conjunto com outras tecnologias.

3.2. Segurança e Aplicações de Blockchain

Não há dúvidas quanto a popularidade da tecnologia Blockchain, mais do que isso o Blockchain tem feito um impacto duradouro no mundo [31]. Como exemplo de seu desenvolvimento, as características do Blockchain tornam sua utilização uma ideia atrativa em muitas áreas de negócios como setores financeiros, governamentais, industriais, farmacêuticos, saúde e segurança cibernética.

Com o crescimento do escopo de aplicação da tecnologia Blockchain e a acessibilidade aos modelos do Blockchain, novas plataformas baseadas na tecnologiaBlockchain foram desenvolvidas. Estas plataformas fornecem suporte para diversas aplicações, mecanismos de consenso, modelos de Blockchain e outras características.

A partir da versão 3.0 do Blockchain, a tecnologia expandiu-se para aplicações e armazenamento descentralizados, a qual beneficia os modelos de Blockchain de consórcio e privado. Sistemas de Blockchain como Linux Foundation's Hyperledger [8] e Multichain [9] foram desenvolvidos propriamente para estes dois modelos, mas sistemas como Credits e Ethereum, possuem versões aplicáveis nos três modelos Blockchain.

Como apresentado (Subseção 3.1.3) o modelo mais utilizado de acesso é o Público, sendo este o que possui maior escopo de plataformas Blockchain desenvolvidas, como Bitcoin [32], Ripple [15], Ethereum [7], Credits [33], entre outros. Para este minicurso são descritas três plataformas que possuem em seus modelos o Blockchain Privado/Consórcio. As plataformas abordadas são: Ethereum, Hyperledger e MultiChain.

3.2.1. Ethereum

O Ethereum [34] oferece um protocolo alternativo para criação de aplicativos descentralizados, fornecendo um conjunto diferente de compensações/recompensas que sejam úteis para uma classe considerável de aplicativos descentralizados. O foco da tecnologia leva em consideração o suporte para o desenvolvimento da aplicação, assim como segurança e a capacidade de diferentes aplicações interagirem de forma muito eficientes.

O Ethereum é um Blockchain de propósito geral que suporta a execução de contratos inteligentes por meio de uma máquina virtual chamada *Ethereum Virtual Machine* (EVM). A EVM é uma camada de abstração, executando no *host* que conecta o cliente com a rede e oferece o ambiente que executa as instruções dos contratos inteligentes de terceiros em uma rede global de computadores. Uma vez criados, os contratos são compilados na EVM que gera o bytecode para o hardware do hospedeiro. Neste modelo, cada operação realizada por um contrato está associado com um custo de execução, fazendo com que transações que modifiquem o estado de um contrato exijam um combustível (chamado de *gas*) ou incentivo financeiro associado com a transação.

A tecnologia empregada no Ethereum possibilita a fácil escrita de contratos inteligentes e construção de aplicativos descentralizados, nos quais possam os usuários possam

criar as suas próprias regras arbitrárias [35]. A partir do desenvolvimento da plataforma Ethereum foi possível, de forma real, verificar as diversas possibilidades de uso da tecnologia Blockchain que até aquele ponto havia apenas sido utilizada com finalidades do setor financeiro.

3.2.1.1. Contas Ethereum

O Ethereum é composto por objetos(contas), que contém um endereço de vinte bytes e transições de estado, sendo transferências diretas de valor e informação entre objetos. Um objeto Ethereum possui quatro campos:

- *Nonce* é um contador usado para garantir que cada uma destas transações possam ser processadas apenas uma vez;
- Balanço da conta.
- Código do contrato, se existir.
- Armazenamento da conta.

O Ethereum possui sua própria forma para financiar a mineração e monetizar a mesma, o seu "combustível", como é definido pela plataforma, é conhecido como *Ether*. De forma geral o Ethereum possui dois tipos de conta:

- Propriedade externa que é controlada por chaves privadas.
- Contas de contrato que são controladas por seu código de contrato.

As contas do Blockchain realizam o processo de comunicação, troca de mensagens e de transações. Estas contas do Ethereum podem ser somente usuários ou também mineradores que obtém seus ganhos a partir da moeda.

3.2.1.2. Mensagens e Transações

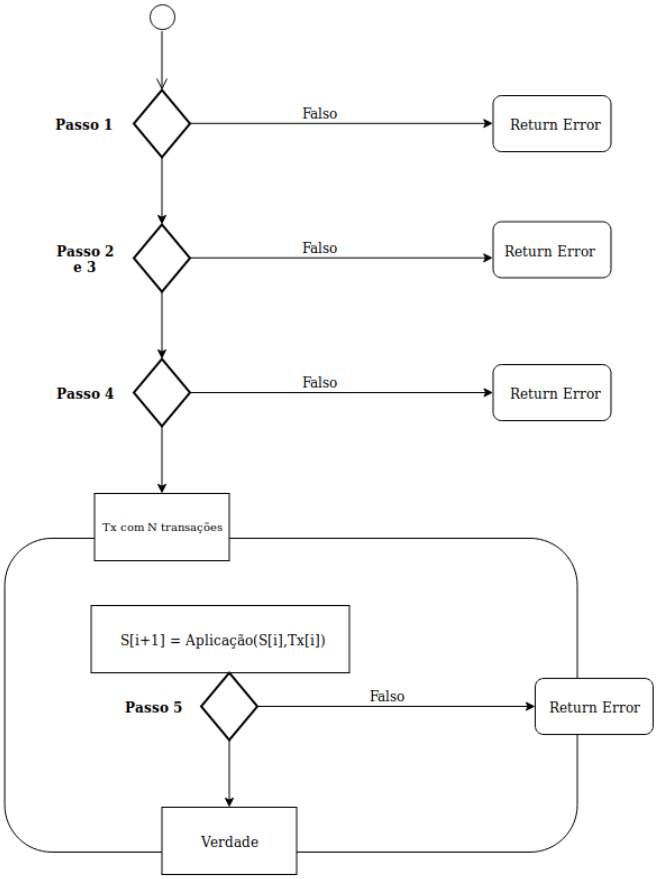
As transações são pacotes de dados assinados que armazenam uma mensagem a ser enviada de uma conta de propriedade externa. As transações possuem: Destinatário, assinatura de identificação do remetente, quantidade de *Ether* para o destinatário, campo de dados opcional, o *STARTGAS* e um *GASPRICE*, o último representa a taxa que o remetente paga por etapas computacionais [35].

Os campos *STARGAS* e *GASPRICE* são cruciais para o modelo de negação de serviço do Ethereum. O uso destes campos tem como finalidade evitar a ocorrência de *loops* infinitos, assim como desperdícios computacionais que são desnecessários [35].

3.2.1.3. Blockchain e Mineração

O Ethereum é muito similar ao Bitcoin. Os blocos de Ethereum armazenam a lista de transações e o estado mais recente, estes mesmos blocos também armazenam o número do bloco e a dificuldade do mesmo. O fluxo básico de validação de bloco segue os seguintes passos listados na Figura 3.4 [35].

Figura 3.4: Fluxo de transição de estados no Ethereum.



Adaptado de [35]

A partir da Figura 3.4 é possível observar o fluxo de transição da validação de um bloco no Ethereum. Nesta mesma figura, percebe-se que há a necessidade de verificar a validação de todos os passos para seguir até o processo que valida o bloco.

Todos os mineradores que participam de forma ativa como mineradores, necessitam a validação de todos estes estados. A partir do processo de validação destes passos, o bloco é integrado a rede e passa a ser validado por outros nós que verificam a autenticidade do bloco e de seu minerador, permitindo que a rede seja mais segura e confiável.

3.2.2. Hyperledger

O projeto Hyperledger é um projeto de código aberto Blockchain e ferramentas relacionadas, teve seu desenvolvimento iniciado em 2015 pela Linux Foundation [8]. O Hyperled-

ger tem como visão que a tecnologia Blockchain tem potencial de impactar quase todas as áreas com atividades. Tendo em vista essa visão, acreditam que no futuro em vez de grandes Blockchains funcionando entre empresas, haverá muitos Blockchains interconectados entre si, cada um destes ajustado e adaptado para determinado propósito.

Este projeto tem como característica abranger uma considerável parte dos espectros de casos de uso, lidando com essa diversidade. Os requisitos básicos que todos os projeto da Hyperledger devem ter são:

- Transações Privadas e Contratos Confidenciais: A estrutura do Hyperledger atende aos requisitos de confidencialidade básica para algoritmos sofisticados e complexos de privacidade.
- Identidade e Auditoria: Diferentemente do Blockchain em si, o Hyperledger oferece aos usuários a capacidade de mascarar sua identidade em determinadas situações e prová-la somente quando necessário.
- Interoperabilidade: Relacionamentos com diferentes redes Blockchain.
- Modular: Estrutura extensíveis e modulares com blocos de construção comuns que podem ser reutilizados. O *WG* define módulos funcionais e interfaces para problemas como comunicação, consenso, criptografia, identidade, armazenamento do livro-razão, contratos inteligentes e política.

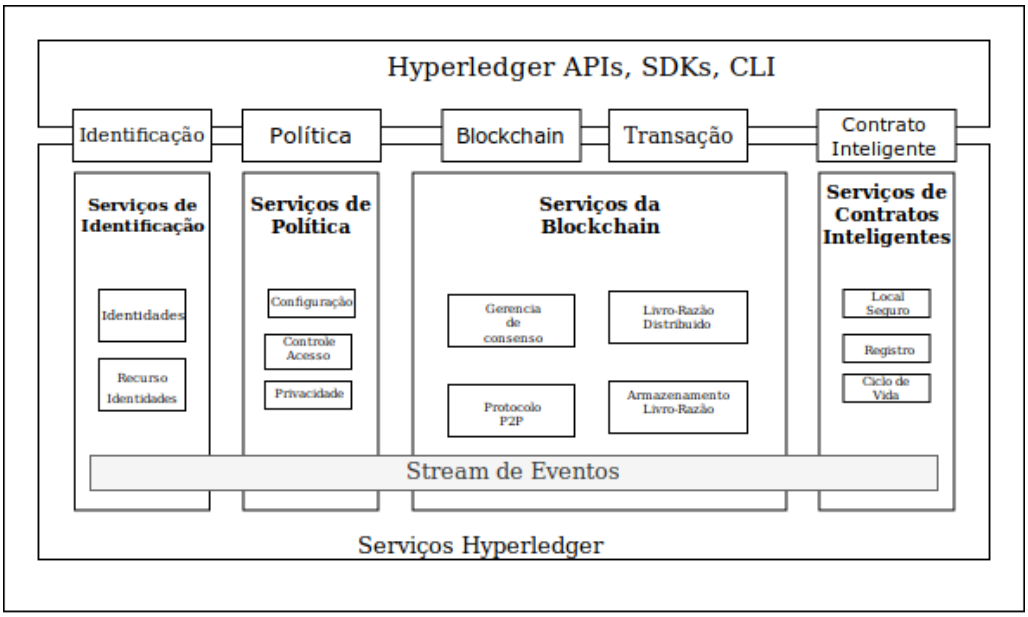
As características da tecnologia Hyperledger realizam a padronização da forma de implementação da tecnologia Blockchain, essa padronização possibilita o amplo desenvolvimento de aplicações com sua tecnologia. A arquitetura do Hyperledger garante a eficiência aos desenvolvedores, a prova deste detalhe é a quantidade de *frameworks* que foram desenvolvidos por empresas que estão disponíveis para uso de modo aberto.

3.2.2.1. Arquitetura Hyperledger

A referência de arquitetura do Hyperledger é dividida em quatro categorias que são ilustradas na Figura 3.5:

- Serviços de Identificação: Este serviço gerencia a identificação de todos participantes e componentes do sistema.
- Serviços de Políticas: Este serviço inclui permissões de acesso e autorização, incluindo políticas de confidencialidade e de consenso da implementação.
- Serviços Blockchain: Este serviço consiste no P2P, distribuição/armazenamento do livro razão e controle do algoritmo de consenso.
- Serviços de Contrato Inteligente: Este serviço inclui ambiente de tempo de execução seguro, registro de contrato inteligente e gerenciamento do ciclo de vida.

Figura 3.5: Arquitetura Hyperledger.



Adaptado de [8]

A arquitetura da especificação do protocolo Hyperledger (Figura 3.5) tem como benefício o suporte de modularidade, interoperabilidade *plug-and-play* e permite o uso de suporte de tecnologias como contêineres para suportar contratos inteligentes. Cada *framework*, desenvolvido na plataforma, possui suas finalidades e sua arquitetura é alterada de acordo com a necessidade da tecnologia.

3.2.2.2. *Framework* e Ferramentas Hyperledger

A plataforma Hyperledger inclui e promove uma variedade de tecnologias de Blockchain de negócios, é integralmente voltada para a versão Blockchain 3.0. Esta estratégia é voltada ao incentivo e reutilização de blocos de construção comuns, permitindo uma rápida inovação de componentes e promove a interoperabilidade entre os projetos.

Os principais frameworks do Hyperledger são:

- *Fabric*: Uma plataforma para criar soluções de contabilidade distribuída com uma arquitetura modular que oferece um alto grau de confidencialidade, flexibilidade, resiliência e escalabilidade.
- *Burrow*: Um cliente Blockchain modular com um intérprete de contrato inteligente com permissão desenvolvido em parte para as especificações da *máquina virtual* (VM) Ethereum.
- *Indy*: Um *ledger* distribuído que fornece ferramentas, bibliotecas e componentes reutilizáveis criados para identidade descentralizada.
- *Iroha*: Um *framework* Blockchain projetado para ser simples e fácil de incorporar em projetos de infraestrutura corporativa.

- *Sawtooth*: Uma plataforma modular para criar, implantar e executar registros (*ledger*) distribuídos.

O Hyperledger possui ferramentas e bibliotecas de utilitários para garantir esta variedade de tecnologias. As principais ferramentas disponíveis são:

- *Caliper*: Uma ferramenta de *benchmark* Blockchain que mede o desempenho de qualquer Blockchain usando um conjunto de casos de uso predefinidos.
- *Cello*: Um conjunto de ferramentas para levar o modelo de implantação sob demanda ao ecossistema Blockchain com maneiras automatizadas de provisionar e gerenciar operações Blockchain que reduzem o esforço.
- *Composer*: Um conjunto de ferramentas e estrutura de desenvolvimento aberto para facilitar o desenvolvimento de aplicativos Blockchain.
- *Explorer*: Um painel para visualizar informações na rede, incluindo blocos, *logs* de nós, estatísticas, contratos inteligentes e transações.
- *Quilt*: Um conjunto de ferramentas que oferecem interoperabilidade implementando o *Interledger Protocol* (ILP), que é basicamente um protocolo de pagamentos projetado para transferir valor em regitros/*ledgers* distribuídos e não distribuídos.

Os *frameworks* e ferramentas disponíveis da plataforma, facilitam o embasamento e utilização da tecnologia Blockchain por diferentes modelos de indústria. Desta forma, revela-se a versatilidade do uso da tecnologia Blockchain e também das possibilidades de desenvolvimento a partir do Hyperledger.

3.2.3. MultiChain

O MultiChain [9] é uma plataforma desenvolvida em 2014 para a criação e implementação de Blockchains privados, dentro ou entre organizações. A tecnologia suporta servidores MS-Windows, GNU/Linux e Apple MacOS, sendo que o MultiChain fornece uma *Application Programming Interface* (API) simples e uma interface de linha de comando.

O MultiChain resolve problemas relacionados à mineração, privacidade e abertura por meio do gerenciamento integrado de permissões de usuários. Com o uso do Blockchain privado, problemas relacionados a escala são facilmente reduzidos, além de que o Blockchain conterá apenas transações que sejam de interesse para este grupo. Três objetivos principais do MultiChain [9]:

- Assegurar que a atividade do Blockchain só seja visível para os participantes escolhidos.
- Introduzir controles sobre quais transações são permitidas.
- Permitir que a mineração ocorra com segurança sem PoW e sem custos adicionais.

Estes três objetivos principais do MultiChain são responsáveis pelo diferencial da plataforma. Um aspecto que merece destaque é que o MuiltiChain possui um administrador, sendo este responsável por toda a parte organizacional desta rede Blockchain que usa o Modelo Privado. O administrador é responsável pela criação do Bloco Gênesis, definição dos participantes da rede e também das operações que podem ser realizadas por estes. Basicamente, a rede MultiChain é constituída por um administrador, mineradores e participantes que interagem com a rede.

3.2.3.1. Mineração MultiChain

A mineração de dados no MultiChain, por ser uma plataforma de Blockchain privada resolve o problema que os participantes podem monopolizar o processo de mineração. A solução apontada pela plataforma está na restrição de número de blocos que podem ser criados pelo mesmo minerador dentro de uma janela de tempo. Este modelo de verificação impõe um tipo de *round-robin*, em que todos os mineradores autorizados devem criar blocos no formato de prioridade/rotação para gerar um Blockchain válido. A validação de um bloco MultiChain é verificada da seguinte forma [9]:

1. Aplicar todas as mudanças de permissões definidas pelas transações no bloco em ordem.
2. Contar o número de mineradores permitidos definidos após a aplicação dessa alteração.
3. Multiplicar os mineradores pela diversidade de mineração, arredondando para obter o *spacing*.
4. Se o minerador deste bloco extraiu um dos blocos anteriores de *spacing*-1, o bloco é invalidado.

A mineração, por ser privada, é restrita para certas entidades, levando ao questionamento dos seus benefícios ao utilizar este Blockchain sobre um banco de dados centralizado que aceita transações de entrada, resolve disputas e consultas relacionadas ao estado do banco de dados. Segundo a equipe do MultiChain, a utilização da plataforma se faz viável pelos seguintes ganhos [36]:

- Cada um dos participantes mantém o controle total sobre seus ativos por meio de sua chave privada.
- O controle dos registrso é distribuído por várias entidades, de modo que nenhum indivíduo ou grupo possa decidir quais transações são válidas ou confirmadas.
- Maior robustez, uma vez que problemas em um servidor não afetará o processo contínuo de transações pela rede como um todo.

Basicamente, cada membro que recebe autorização para participar da rede do MultiChain é responsável por suas ações e controles dentro da rede. Há uma comunicação entre todos os nós, que a partir do momento no qual um bloco ou uma transação é validado,

estes nós passam a realizar o processo de confirmação de transação e do bloco, garantindo a segurabilidade de todo o sistema.

3.2.4. Comparativo

Abordadas as três plataformas apresentadas neste minicurso: Ethereum, Hyperledger e MultiChain, torna-se relevante a realização de um comparativo que contenha as principais e mais atrativas configurações de cada uma dessas plataformas. Como características para comparação, apresenta-se na Tabela 3.4 as características:

- Modelos Blockchain: com base nas sua características, classificar o Blockchain nos modelos conhecidos.
- Versão Blockchain: As versões Blockchain que são aplicadas na plataforma.
- Linguagem de programação: Quais as principais linguagens de programação que realizam interações com a plataforma.
- Mecanismos de Consenso: Os mecanismos de consenso que são aplicados e utilizados pela plataforma.
- Licença: Se a aplicação possui licença aberta ou proprietária.

Tabela 3.4: Características das plataformas que utilizam a tecnologia Blockchain.

Plataformas	Modelo	Versão	Linguagem	Consenso	Licença
Ethereum	Pública/Privada	Blockchain 1.0/2.0/3.0	C++, Go, JavaScript,Python	PoW e PoS	Aberto
Hyperledger	Privada	Blockchain 3.0	C++, Java, Python, Ruby	PBFT e outros[2]	Aberto
Multichain	Privada	Blockchain 2.0/3.0	C#, JavaScript, PHP, Python, Ruby	PBFT	Aberto

A partir da Tabela 3.4 é possível observar que a plataforma Hyperledger possui o maior escopo quando se relaciona a aplicações descentralizadas. No entanto, possui maior exigência de configuração e adequação para o seu próprio problema. Já as plataformas Ethereum e MultiChain possuem seu principal escopo voltado para contratos inteligentes a partir da aplicação de transações, tornando-as mais adequadas para agilidade no desenvolvimento de aplicações e mais simples de serem configuradas. Contudo, torna-se necessário o conhecimento das vulnerabilidades que atingem a tecnologia Blockchain, pois estas podem influenciar o bom funcionamento destas redes desenvolvidas com o uso da tecnologia.

3.2.5. Ataques e vulnerabilidades envolvidos com Blockchain

Com a necessidade de investigar as principais vulnerabilidades conhecidas na tecnologia, é realizada uma revisão bibliográfica que identifica os principais ataques e vulnerabilidades em relação ao uso de Blockchain. A Tabela 3.5 é dividida pelos seguintes campos: Nome do ataque/vulnerabilidade, versão Blockchain, categoria e relação com a segurança.

[2] A plataforma Hyperledger permite o desenvolvimento e aplicação, implementados, de vários consensos a escolha do desenvolvedor.

Tabela 3.5: Principais ataques e vulnerabilidades identificadas relacionadas a Blockchain classificadas em categorias.

Ataque/Vulnerabilidade	**Versão Blockchain**	**Categoria**	**Relação**
"Vulnerabilidade 51%"	Blockchain 1.0, 2.0, 3.0	Redes	Imutabilidade/Procedência
Chave Privada de Segurança	Blockchain 1.0	Usuário	Cifragem
DDoS	Blockchain 1.0, 2.0, 3.0	Redes	Transações/Procedência
Gasto Duplo	Blockchain 1.0, 2.0, 3.0	Redes	Arquitetura
Ataque Eclipse	Blockchain 1.0, 2.0, 3.0	Redes	Arquitetura/Transação/Procedência
Ataque de Vivacidade (*Liveness*)	Blockchain 1.0	Poder Computacional	Transações/Procedência
Mineração Egoísta	Blockchain 1.0, 2.0, 3.0	Poder Computacional	Procedência
Otimização *Smart Contract*	Blockchain 1.0, 2.0, 3.0	Usuário	Transparência/Transação
Privacidade de Transação	Blockchain 1.0	Redes	Arquitetura
Retenção de Blocos	Blockchain 1.0, 2.0, 3.0	Poder Computacional	Transparência
Smart Contract Malicioso	Blockchain 2.0	Usuário	Procedência/Transações

A partir da Tabela 3.5 é possível identificar as principais vulnerabilidades conhecidas na tecnologia Blockchain e quais versões da tecnologia estão sobre influencia destas vulnerabilidades. Analisando as vulnerabilidades identificadas (Tabela 3.5), foi realizada uma classificação inicial associando estas três categorias principais: Redes, Poder Computacional e Usuário.

1. **Redes:**
   Aspectos relacionados ao controle dos nós, forma das transações ou até mesmo incapacitação operacional da rede.

   (a) DDoS: O objetivo do atacante é tornar o serviço indisponível durante o processo de ataque. Os sistemas de defesa contra (DDoS) normalmente não são capazes de resistir sozinhos contra ataques em larga escala [37]. É importante notar que devido a natureza totalmente distribuída/replicada de Blockchains, estes se tornam naturalmente resilientes a ataques de negação de serviço distribuídos. Além disto, o modelo de custo associado a Blockchains públicos (impondo taxas para as transações), evita que usuários maliciosos realizem o envio em massa de transações impondo um alto custo a este tipo de ataque. *Distributed Ledger*s (DLTs) que não possuem um modelo de custo associado a cada transação estão suscetíveis a ataques de negação de serviço por parte de nos maliciosos dentro de um consorciou ou organização. Porém, entidades com permissões de escrita em uma DLT possuem um certo grau de confiança dentro da organização/consórcio.

   (b) Ataque Eclipse: O Ataque Eclipse tem como intenção ganhar controle sobre os nós, desta forma controlando grande maioria do tráfego da rede. Quando um ataque Eclipse é bem sucedido, permite que o invasor controle todo tráfego de sobreposição, permitindo negação arbitrária de serviço ou de censura [38].

   (c) Gasto Duplo: Esta vulnerabilidade está diretamente atribuída às criptomoedas, nas quais atacantes fazem múltiplas transações com a mesma moeda. Para este ataque ser realizado é necessário que o atacante minere privativamente, tentando estender ao máximo o bloco minerado sem publicar o cálculo, então é transmitida a transação para a organização de interesse e esperar para que a

transação seja registrada, e então minerado até que este bloco seja maior do que o bloco público, assim é publicado o cálculo apagando a transação feita com a organização [39]. O problema do gasto duplo ocorre de maneira diferente para os diferentes tipos de mecanismos de consenso. Por exemplo, em um Blockchain baseada em PoW, o gasto duplo exige que o atacante tenha controle sobre 51% do poder computacional da rede, decidindo, deste modo, priorizar uma cadeia de blocos em detrimento a outra. No entanto, em um Blockchain baseada em em PoS, na qual não se exige um gasto de recursos como prova de trabalho, a rede fica mais suscetível a criação de cadeias paralelas (uma vez que estas cadeias podem ser geradas sem esforço computacional). Neste caso, a alternativa para Blockchains baseados em PoS é um mecanismo de consenso híbrido entre PoS e PoW no qual introduz-se um esforço computacional para a geração de blocos, mas mantendo a determinação dos mineradores baseando-se em seus recursos ("*stake*").

(d) Privacidade de Transação: Esta vulnerabilidade está relacionada com a possibilidade de rastreabilidade do Blockchain, conforme o modo de programação o destinatário pode ser detectado através da transação da rede [40]. A maioria das criptomoedas públicas utilizam um esquema de pseudo-anonimidade, na qual usuários são identificados por endereços (*hash* gerado criptograficamente) não revelando, deste modo, sua identidade. Neste sentido, é possível observar todas as transações financeiras de contas mas não a identidade do usuário que gerencia a conta. No entanto, a conversão de uma criptomoeda para um dinheiro final requer a utilização de uma casa de cambio, que por sua vez exige a identificação do usuário, e assim estabelecer uma relação entre usuário-conta. No entanto, existem criptomoedas como Monero [41] que utiliza um protocolo (CryptoNote) que obfusca por meio de uma primitiva baseada em *ring signatures* as três partes essenciais de uma transação: remetente, valor e destinatário.

(e) "Vulnerabilidade 51%": Este ataque é realizado a partir do mecanismo de consenso do Blockchain, em que se o invasor obter 51% do *hashing* do *pool* ele tem o controle sobre o bloco. A partir disto, existe a possibilidade de realizar mudanças entre outras questões [40].

2. **Poder Computacional:**
Ataques relacionados ao aumento de *hashing* tem o intuito de obter benefícios sobre mineradores honestos ou simplesmente reduzir a recompensa que um grupo de mineradores tem direito.

(a) Retenção dos Blocos: O objetivo deste ataque é sabotar mineradores honestos, fazendo com que desistam do *pool*. O minerador inicia o processo como um minerador honesto, mas o atacante envia apenas uma parcial da prova de trabalho. Se encontrar uma solução completa que constitua uma prova de trabalho descarta a solução, reduzindo o rendimento total [42].

(b) Mineração Egoísta: Este ataque tem como finalidade obter recompensa ou perda de poder computacional de mineradores honestos. Especificamente, a

mineração egoísta força os mineradores honestos a gastar seus ciclos computacionais em blocos destinados a não fazer parte do Blockchain. Mineradores egoístas atingem esse objetivo revelando seletivamente seus blocos minados para invalidar o trabalho dos mineradores honestos [43].

(c) Ataque de Vivacidade(*Liveness*): Este ataque é realizado para atrasar o máximo possível o tempo de confirmação de um alvo transação [44]. Este ataque passa por três fases: a primeira que é quando o minerador malicioso cria vantagem sobre os mineradores honestos, a segunda é de negação de serviço (*Denial-of-Service* (DoS)) e a terceira que é a retardadora do Blockchain.

3. **Usuário**:
   Ataques relacionados a categoria usuários são relacionados a programação, intenções maliciosas e até mesmo falta de conhecimento.

   (a) Chave Privada de Segurança: Este ataque é relacionado a chave de segurança privada de cada usuário. Caso a chave for roubada ou perdida o usuário dificilmente conseguirá recupera-lá.

   (b) *Smart Contract* Criminoso: Este contrato pode facilitar o vazamento de informações confidenciais, roubo de chaves criptográficas e vários tipos de comportamentos do usuário [40].

   (c) Otimização *Smart Contract*: Relacionado a programação do contrato, em que este é pouco otimizado causando grandes perdas para quem o utiliza. Segundo [45] foram detectados alguns padrões em códigos que demonstram funções inúteis, que não são utilizadas, e também códigos em *loop*.

A classificação apresentada foi dividida em apenas três categorias, que de modo geral, abrangem as principais necessidades que são conhecidas pelo Blockchain. Há autores que realizam a classificação em cinco diferentes categorias, explorando um maior número de vulnerabilidades, mas vale ressaltar que em uma classificação completa estas vulnerabilidades podem ser divididas em sub-categorias para maior refinamento das vulnerabilidades.

3.3. Analisando mecanismos de consenso PoW e PBFT quanto a DoS

A partir dos dados apresentados, nas Seções 3.1 e 3.2, é perceptível que o Blockchain possui uma adaptabilidade e versatilidade para várias realidades e necessidades diferentes. Porém, a tecnologia apresenta algumas adversidades relacionadas as questões de custo computacional e também às questões de segurança, que variam de acordo com o modelo de aplicação.

Com essa premissa, torna-se necessário o estudo de viabilidade na aplicação do Blockchain, as necessidades da instituição, assim como as tecnologias aplicadas em conjunto com o Blockchain. Quanto ao Blockchain é necessário considerar as questões relacionadas a tecnologia como: Modelo, versão, mecanismos de consenso, etc.

3.3.1. Definição e motivação

O crescimento exponencial da tecnologia Blockchain é incontestável, as demandas por aplicações da tecnologia possuem diferentes contextos e realidades de usuários. Quanto a estas aplicações do Blockchain, busca-se melhorar a qualidade e eficiência da tecnologia, mas também existe a preocupação ligada ao custo computacional que é necessário para o funcionamento da tecnologia Blockchain. Neste sentido, percebe-se que há diversos mecanismos de consenso, que são responsáveis em realizar várias operações e entre estas está o processo de validação do Blockchain. Contudo, não foram encontrados estudos que tenham como objetivo analisar diferentes mecanismos de consenso do Blockchain em contexto com nenhuma outra tecnologia.

Como exemplo de aplicação, possui-se dois mecanismos de consenso que possuem seus benefícios e problemas, como o PBFT e o tradicional mecanismo PoW. Para avaliar o impacto que a escolha do mecanismo possui, há a necessidade de entender sobre os aspectos da tecnologia utilizada. Isto é, se é necessário aplicar o Blockchain ou não, os usuários envolvidos e também que atenda as demandas e objetivos das aplicações. Exemplificando, aplicar um Blockchain para registrar operações de alocação de VM em nuvens computacionais implica em obter requisitos de transparência, procedência dos dados, proteção e gerenciamento, mudança de modelo de ameaça, *etc* [46].

A decisão de da escolha de qual tecnologia aplicar é complexa, impactando diretamente no desempenho e possíveis vulnerabilidades da aplicação. Além das ameaças e vulnerabilidades padrões que diversas tecnologias possuem, percebe-se, a partir, da Tabela 3.5 alguns dos ataques e vulnerabilidades existentes para a tecnologia Blockchain. Com este levantamento, torna-se relevante analisar os mecanismos de consenso Blockchain junto aos aspectos de segurança definidos por organizações, como o *National Institute of Standards and Technology* (NIST).

No atual cenário do uso da tecnologia Blockchain, há incertezas quanto a viabilidade da aplicação da tecnologia, principalmente pelo fato de sua popularidade, de qual melhor modelo e mecanismo deve-se aplicar com outras tecnologias, principalmente pela variabilidade de possibilidades existentes. O problema em questão é a falta de critérios para comparar os mecanismos de consenso do Blockchain e sua relação com vulnerabilidades, ameaças e riscos existentes que são intrínsecos a tecnologias e arquiteturas empregadas na solução Blockchain.

3.3.2. Motivação da análise

A tecnologia Blockchain tem apresentado considerável versatilidade no contexto de possíveis aplicações, tornando-se benéfica para diferentes contextos, *e.g.*, contratos inteligentes, internet das coisas (IoT), segurança da informação, *etc* [1, 2]. A utilização do Blockchain de forma paralela a estas tecnologias tem proporcionado um ambiente íntegro, seguro, descentralizado e transparente [47].

A proposta de experimento neste trabalho consiste na realização de uma breve análise de segurança dos mecanismos de consenso PoW e PBFT aplicados ao Blockchain quando submetidos ataques simples de DoS. Dentre as necessidades atuais das tecnologias baseadas em Blockchain estão aspectos como: auditoria, procedência dos dados,

proteção e gerenciamento dos dados e a segurança no gerenciamento do ciclo de vida das informações. Em que estes aspectos relevantes podem ser supridos com o emprego do Blockchain, que disponibiliza meios para incorporar questões como transparência, autonomia, imutabilidade, anonimado, descentralização dentre outros.

Atualmente, há pesquisas que indicam que os principais responsáveis pelas violações de dados que ocorrem em nuvens computacionais são causadas por pessoal internos da instituição que está conectado diretamente a sua rede local [48, 49]. Este fato justifica uma necessidade de análises voltadas para soluções Blockchain baseadas no Modelo Privado e de Consórcio. Outro preceito para a realização das análises são as vulnerabilidades apresentadas na Seção 3.2.5, que delimitam em grupos a realização da análise.

A escolha do ataque a ser aplicado pertence ao grupo de redes, que envolvem ataques como: DoS, Ataque Eclipse, Gasto Duplo, Privacidade das Transações e a Vulnerabilidade "51%". As questões relacionadas a este grupo impactam no desempenho, funcionalidade, custo computacional e podem atingir não somente a rede Blockchain, mas todo o sistema em si [50].

O intuito desta análise é a identificação das melhores alternativas de aplicação para os mecanismos de acordo com o contexto do Blockchain. Com o objetivo de facilitar que usuários da tecnologia realizem integrações ao seus sistema com uma tecnologia segura, eficiente, viável, custo benefício, desempenho e funcionalidade para o seu cenário.

3.3.3. Critérios de Análise

A partir das atuais necessidades das tecnologias que podem ser aplicadas em conjunto com o Blockchain, nos trabalhos relacionados e nos ataques e vulnerabilidade listados na Subseção 3.2.5, foi possível identificar e analisar as principais preocupações com os mecanismos de consenso para segurança do Blockchain. Com base nestas preocupações, foram definidos os critérios:

- Procedência dos dados: A procedência dos dados é um processo de auditoria que mantém um registro, não somente de *logs*, mas de todas operações realizadas com um objeto. Basicamente a partir da procedência é possível compreender tudo que foi realizado com determinado objeto e suas possíveis alterações e até mesmo fraudes. De acordo com o *Cloud Security Alliance* (CSA) os grandes desafios das nuvens computacionais são: Visibilidade detalhada, maior escopo de aplicação e transparência reduzida. A garantia de procedência dos dados tem como visão auxiliar que estes desafios possam ser minimizados, para então auxiliar o provedor da rede e também ao usuário possíveis garantias de que seu conteúdo esteja protegido e somente com as reais modificações que foram aprovadas pelos mesmos.

  Quanto a verificabilidade da confiabilidade do sistema, baseia-se pela inviolabilidade do mecanismo de consenso. A segurança do algoritmo de consenso é baseado no problema dos generais bizantinos [51], em que o Blockchain utiliza um livro-razão no qual todas alterações realizadas em determinado objeto é feita a conferência das assinaturas digitais para geração das transações, após isso a atualização e validação dos blocos. Alguns ataques descritos na Subseção 3.2.5 podem gerar problemas de funcionalidade ao consenso, ataques como: "Vulnerabilidade 51%",

Ataque Eclipse, DDoS, Ataque de Vivacidade, *etc*. Ataques estes que levam a necessidade de verificação dos mecanismos de consenso como seguros para determinadas aplicações.

A verificação desta inviolabilidade é feita a partir do livro-razão e da verificação do tamanho de bytes dos objetos. Para garantir que realmente nenhuma mudança realizada através dos ataques seja validada e represente periculosidade ao sistema.

- Controle e Gerenciamento do Nó: O controle de nó é um processo que visa obter o controle sob todo o nó, permitindo a existência de vulnerabilidade e de alterações sobre o bloco. Uma parte considerável dos problemas voltados as vulnerabilidades da tecnologia Blockchain vistos na Subseção 3.2.5 são causados por problemas de segurança que podem ser evitados com maior eficiência e controle dos nós, evitando desta forma que conteúdos protegidos possam ser aceitos, acessados ou alterados. Alguns ataques como: Ataque Eclipse, DDoS e Mineração Egoísta podem dificultar a validação de consenso ou até mesmo gerar alterações nas transações para determinados objetos.

  Quanto a forma de medidas para determinada questão, são utilizados ataques que promovam controle do nó ou que instabilizem a rede para após o término que possa ser verificado a influência do ataque de modo geral na rede Blockchain. A principal forma de identificação da eficiência do ataque é obtida a partir de dados coletados durante a execução do mesmo e também a partir de transações e blocos que sejam validados ou contestados perante a rede.

3.4. Ambiente de Experimentação

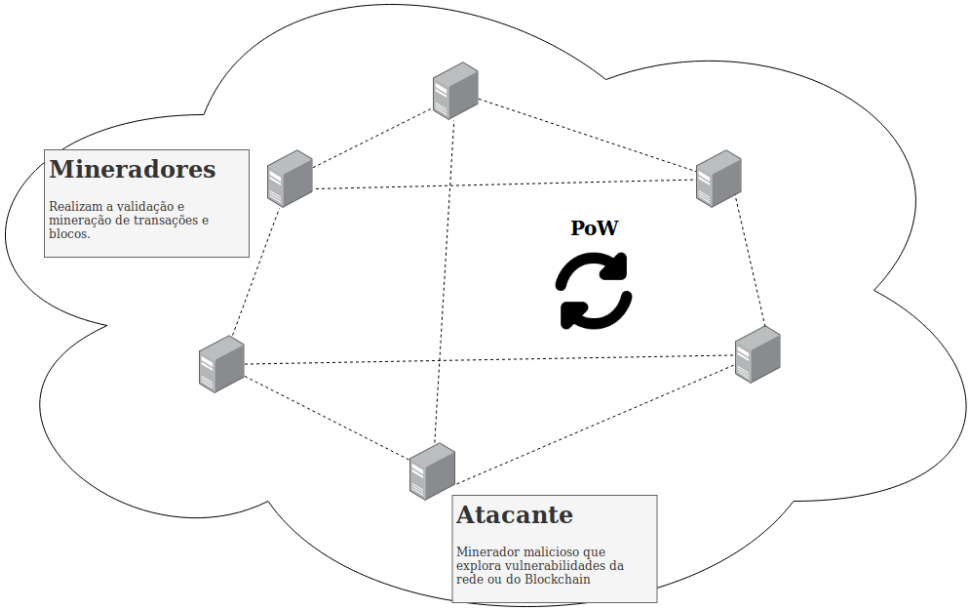
Os experimentos visam mostrar como alguns ataques podem explorar algumas soluções Blockchain e como mecanismos de consenso são afetados. Deste modo, foi definido um plano de testes e um ambiente de testes que possibilitasse a execução e replicação dos experimentos de modo facilitado. Todos os cenários usam como base uma topologia *flat* com seis VMs, todas com a distribuição GNU/Linux Ubuntu Server 16.04 LTS.

3.4.1. Plano de Testes

Este plano de testes foi desenvolvido baseando-se nos critérios estabelecidos na Seção 3.3. O objetivo é realizar análise dos métodos de consenso Blockchain, para isso são empregados dois cenários, as configurações de ambiente de testes utilizadas são as exigidas pelos sistemas Blockchain Ethereum e Multichain, a distinção destes cenários é, essencialmente para estes experimentos, o mecanismo de consenso aplicado. Os cenários definidos são:

- Cenário 1 - Ethereum: O cenário utiliza o modelo de rede privada de Blockchain, em que as seis instâncias realizam o processo de mineração e validação de transações do Blockchain (Figura 3.6).

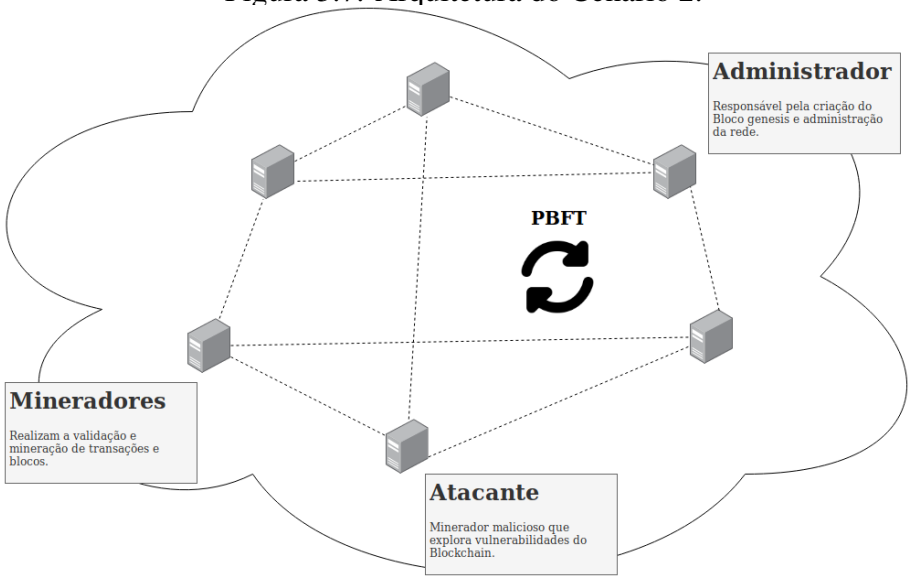
Figura 3.6: Arquitetura do Cenário 1.



O objetivo desta experimentação é investigar as características e comportamentos iniciais do mecanismo de consenso PoW. A partir da análise inicial de comportamento, uma destas instâncias se tornara atacante tendo em vista as vulnerabilidades da rede e da implementação do Blockchain. A partir da finalização do experimento, é realizado um processo de análise do cenário após a execução dos ataques. O intuito deste experimento é averiguar a procedência dos dados, controle e gerenciamento dos nós e por fim a estabilidade da rede, identificando desta forma as influências do ataque em relação com o funcionamento da rede.

- Cenário 2 - Multichain: O cenário possibilita que as seis instâncias realizem o processo de mineração e validação de transações e blocos, mas somente uma destas VM possui o nível de administrador, o que a torna parcialmente descentralizada (Figura 3.7).

Figura 3.7: Arquitetura do Cenário 2.



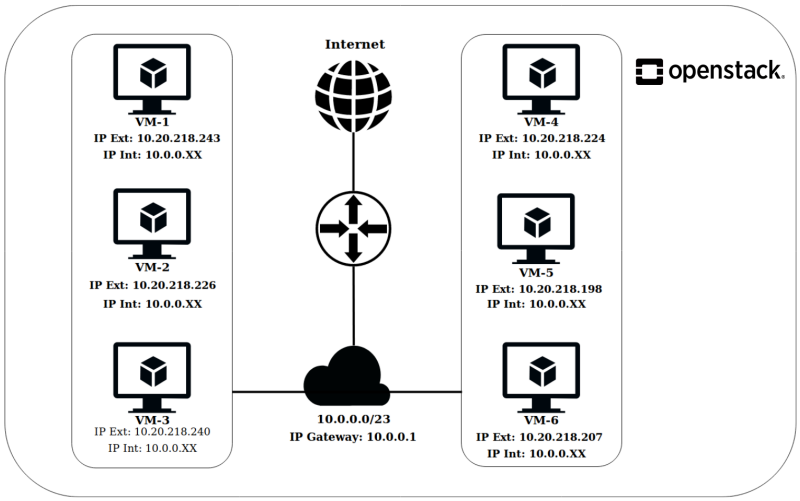
O Cenário 2 possui o objetivo de investigar as características e comportamentos da rede Blockchain privada e parcialmente descentralizada com o mecanismo de consenso PBFT. Ao fim das análises iniciais do comportamento da rede, uma destas instâncias se torna atacante que tem como objetivo comprometer o funcionamento da rede e do Blockchain. Após a realização do ataque, um processo de análise do cenário é realizado a partir dos dados coletados durante a execução destes. O experimento tem como objetivo de averiguar a procedência dos dados, o controle/gerenciamento dos nós e também a estabilidade da rede. O intuito deste experimento é a identificação das influências causadas pelo ataque no sistema, em relação ao seu funcionamento e consenso.

Quanto as VMs em ambos ambientes não possuem habilitados recursos de memória virtual *swap*, utilizando apenas a memória principal. A escolha desta abordagem é necessária pois para obtenção de informações mais claras e com maior facilidade é necessário o esgotamento dos recursos.

3.4.2. Arquitetura do Ambiente

O Laboratório de Processamento Paralelo e Distribuído (LabP2D) possui uma nuvem computacional *Infrastructure as a Service* (IaaS) baseada na solução aberta OpenStack versão Mitaka, com nome de Nuvem Tchê. Quanto ao ambiente de testes, o mesmo foi construído como um projeto OpenStack na Nuvem Tchê, tal que a abordagem escolhida foi o uso executado de VMs para execução das soluções Ethereum e Multichain usando o Modelo Privado. Embora não seja comum o uso de um sistema Blockchain inteiro em uma nuvem apenas, a escolha se deu pela praticidade de execução dos experimentos e isolamento de outros fatores (e.g., tráfego de *background*) que podem tornar a análise mais subjetiva e complexa. Pensando nisso, a arquitetura foi construída através de três componentes principais: Roteador, Rede e VM de acordo com a Figura 3.8. Quanto a arquitetura da rede utilizada, é uma rede *overlay*, que é conectadas por enlaces e *switches* virtuais no OpenStack. É importante ressaltar que não foram criados mais roteadores (saltos) ou inserida latência, pois os experimentos visam averiguar o consumo de recursos.

Figura 3.8: Arquitetura de ambiente de testes no OpenStack.



A partir da Figura 3.8 é possível observar que a comunicação com a Internet é realizada através do Roteador que possui como interface interna o IP 10.0.0.1 que é do *gateway* da Rede interna 10.0.0.0/24. Na rede interna possui as instâncias executando o GNU/Linux Ubuntu Server 16.04 LTS, que possui diferentes IP 10.0.0.3-20 denominados entre as VM-1 a VM-6. Sobre estas VMs foram instaladas diferentes plataformas de Blockchain, constituindo um Blockchain usando o Modelo Privado. Vale mencionar que, dependendo do tipo de ataque realizado, é necessário que mais de uma VM possa realizar os ataques que são mencionados na Seção 3.3.

3.5. Experimentos aplicados

Para cada um dos cenários é apresentado o objetivo do experimento, fluxo da rede, comportamento do mecanismo de consenso, os ataques realizados e por fim os possíveis problemas que foram causados na plataforma.

3.5.1. Cenário 1

O primeiro cenário de experimento é utilizado a plataforma Ethereum com seis VMs que utilizam sistema operacional GNU/Linux Ubuntu Server 16.02 LTS. O Blockchain Ethereum é um sistema que por padrão aplica o modelo público do Blockchain, mas permite que o sistema utilize os modelos privado e de consórcio.

A tecnologia utiliza uma rede totalmente descentralizada, na qual diferentes pares se conectam e realizam trocas de informações entre si, estas trocas de informações utilizam os protocolos de rede TCP e UDP. Na Listagem 3.1 é apresentado o formato de um bloco usado neste cenário.

Listagem 3.1: Informações do Bloco

```
eth.getBlock(6821)
{
 difficulty: 2871846,
 extraData: "0xd88301081b846765746888676f312e31302e34856c696e7578",
 gasLimit: 8000000,
 gasUsed: 21000,
 hash: "0x85eef246d6b1fb07e415160bb0cd965c45e58025dcd82b6f26a66adf7905aaab",
 logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000000000000000000000000000
     0000",
 miner: "0xea1cefe73fd12583bfd7911db2d4726a3aedee0b",
 mixHash: "0xa0a129927080a4f81dfe1033b5e7a3c087ec992a315e79a563124425f3b48de0",
 nonce: "0x699c8a2dbccdf163",
 number: 6821,
 parentHash: "0x5928b23f71c07af600236c52978f95cb438a8853fcd2a41ca0b6d2359f2fa4f
     7",
 receiptsRoot: "0x13cbb7efc83a39275d711a65415daa8d9050f62143a257a82be6a10e0c77
     da7f",
 sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d4934
     7",
 size: 655,
 stateRoot: "0xb05a6fc64f193b45d4bf068d3635f8c00dbfd9c7830f8e7feb40feebe8b1f5c8
     ",
 timestamp: 1559889720,
 totalDifficulty: 6566222689,
 transactions: ["0xfe9e7da787548c7160cd34bc0bac0450a62a29031a8647398a613e1c1809
```



```
        bafa"],
21    transactionsRoot: "0x17e7db8c04457551df93e6e9fee5a286aad50e6b93d28e1b6303628
        aed12248c",
22    uncles: []
23  }
```

A partir da Listagem 3.1 observa-se informações disponíveis para consulta sobre os blocos que pertencem a rede Blockchain. Há informações relevantes, como a linha 9, em que apresenta a carteira utilizada para a mineração do bloco, já as linhas 20 e 21 apresentam as transações que foram realizadas e estão registradas e auditadas. Quanto a realização de uma transação, é possível identificá-la na Listagem 3.2.

Listagem 3.2: Informações de uma transação

```
1   eth.getTransaction("0xfe9e7da787548c7160cd34bc0bac0450a62a29031a8647398a613e1c1
        809bafa")
2   {
3     blockHash: "0x85eef246d6b1fb07e415160bb0cd965c45e58025dcd82b6f26a66adf7905aaab
        ",
4     blockNumber: 6821,
5     from: "0xea1cefe73fd12583bfd7911db2d4726a3aedee0b",
6     gas: 90000,
7     gasPrice: 1000000000,
8     hash: "0xfe9e7da787548c7160cd34bc0bac0450a62a29031a8647398a613e1c1809bafa",
9     input: "0x",
10    nonce: 0,
11    r: "0xc56d5e83f9d21a2e6a36c881194c2fbc41e847b41f016c21698d4349e786b525",
12    s: "0x16086477c6640e9d922b41ae9b2ec3f3118c118769f0af75ec792201631d8e83",
13    to: "0xf6a285239af6faa74e1686fac7c317192cd88768",
14    transactionIndex: 0,
15    v: "0x75bcd186",
16    value: 10000000000000000000
17  }
```

Na Listagem 3.2 é verificado uma transação realizada entre a VM-1 a partir da carteira, presente na linha 5, para a VM-2 em que utiliza a carteira apresentada na linha 13. As informações que são fornecidas para consulta, permitem a verificação e validação não somente da transação, mas também do bloco em que esta encontra-se indexada.

A partir das Listagens 3.1 e 3.2 é possível averiguar o estado inicial da arquitetura e do comportamento desta rede Blockchain privada. Seis VMs participam do processo de validação das transações e dos blocos com o mecanismo de consenso PoW, possibilitando o monitoramento e auditoria de toda movimentação e mudanças ocorridas nos objetos.

Estas informações permitem a verificação da existência e validação não somente da transação, mas também do bloco que ela pertence com associação as carteiras que foram envolvidas durante o processo. Ressalta-se que estas carteiras apresentam somente os dados das mesmas, garantindo anonimato total dos dados dos envolvidos.

3.5.1.1. Ataque de Negação de Serviço

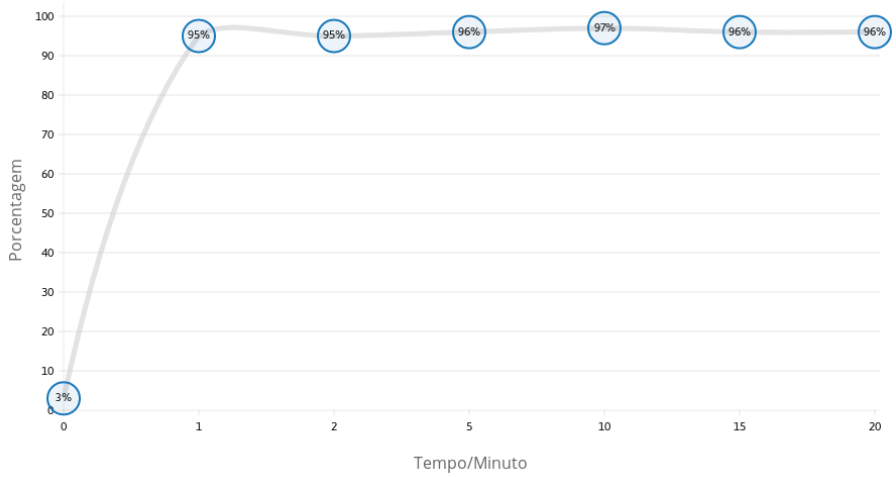
Foram escolhidos dois ataques:

- *Transactions Flood*: realizado a partir da VM-2 e VM-6 endereçados para a carteira presente na VM-1. O Blockchain Ethereum, como característica sua, não possui um administrador da rede ou um nó centralizado que monopoliza e distribui os

processos para criação e validação de blocos e transações. Com a falta de um meio centralizados, o ataque em um Blockchain usando o Modelo Privado não afetou em proporções consideráveis o desempenho de memória e processamento da VM, pois a descentralização permite que a rede funcione normalmente, validando as transações e minerando novos blocos.

- *UDP Flood* foi realizado a partir da VM-6 que atacavam a VM-1. O Ethereum utiliza o serviço *User Datagram Protocol* (UDP) para troca de informações entre os nós. Uma aspecto relevante quanto ao uso do mecanismo PoW, em um Modelo Privado, é a não necessidade do uso de uma GPU, deixando a CPU responsável pela mineração dos blocos, que é observado a partir da Figura 3.9.

Figura 3.9: Análise do consumo de processador, sem o ataque *UDP Flood*.



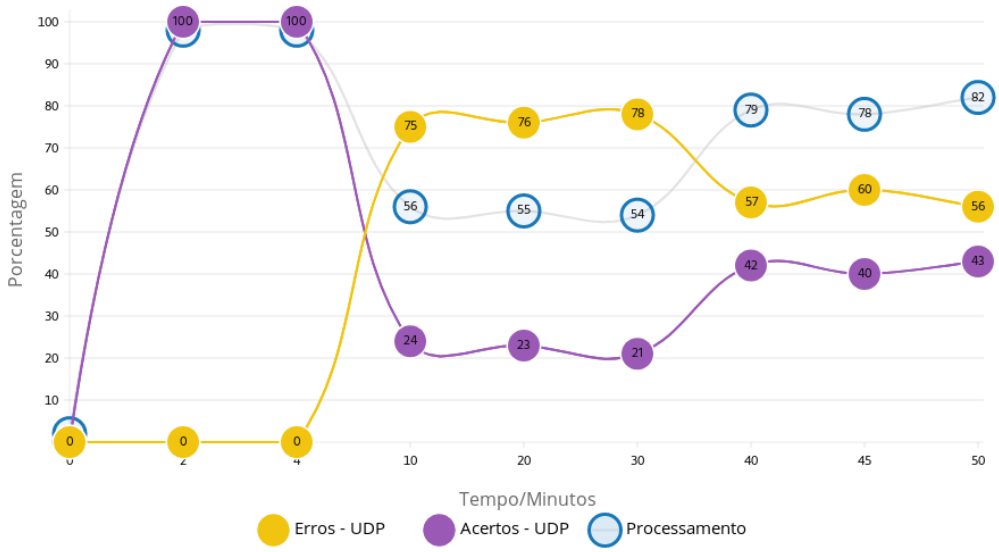
Os dados apresentados na Tabela 3.6 e pela Figura 3.9 é possível observar, a partir dos pontos, o consumo do processamento da VM, que é realizado pela mineração de novos blocos, vale lembrar que a Figura 3.9 não há a ocorrência de nenhum ataque, somente o consumo normal do processamento.

Tabela 3.6: Tabela do consumo de processador sem o ataque *UDP Flood*.

Tempo/Min	0	1	2	5	10	15	20
Processamento	3,1%	95,74%	95,08%	96,47%	97,40%	96,56%	96,89%

A Figura 3.10 apresenta o consumo da *Central processing unit* (CPU) quando a instância VM-1 sofre um ataque *UDP Flood*.

Figura 3.10: Análise do consumo do processador durante o ataque *UDP Flood*.



A partir da Figura 3.10 e Tabela 3.7 observa-se três pontos: O consumo da CPU do sistema, Pacotes UDP que foram recebidos/respondidos e o terceiro indica os pacotes UDP que apresentaram erro. Quanto ao consumo de processamento, é notável que seu percentual de uso cai consideravelmente com o aumento da quantidade de pacotes UDP com erros, é observado no sistema que a partir desta diminuição do uso de processamento paralelamente ocorre uma diminuição significativa na mineração e validação de blocos e transações ou até mesmo a mineração é suspensa temporariamente.

Tabela 3.7: Tabela do consumo de processador durante o ataque *UDP Flood*.

Tempo/Min	0	2	4	10	20	30	40	45	50
Processamento	2,3%	98,78%	98,70%	56,28%	55,86%	54,56%	79,73%	78,39%	82,45%
Acertos - UDP	0%	100%	100%	24,5%	23,7%	21,7%	42,3%	40,0%	43%
Erros - UDP	0%	0%	0%	75,5%	76,3%	78,3%	57,7%	60,0%	56,8%

Na Figura 3.11 e Tabela 3.8 é possível observar um crescimento do nível de consumo de memória, que é gerado a partir das filas de requisição da troca de informações entre os nós da rede Blockchain. Contudo, a mesma apresenta estabilidade no consumo durante a realização do ataque, indicando que o consumo de memória não é afetado durante o processo.

Figura 3.11: Análise do consumo de memória durante o ataque *UDP Flood*.

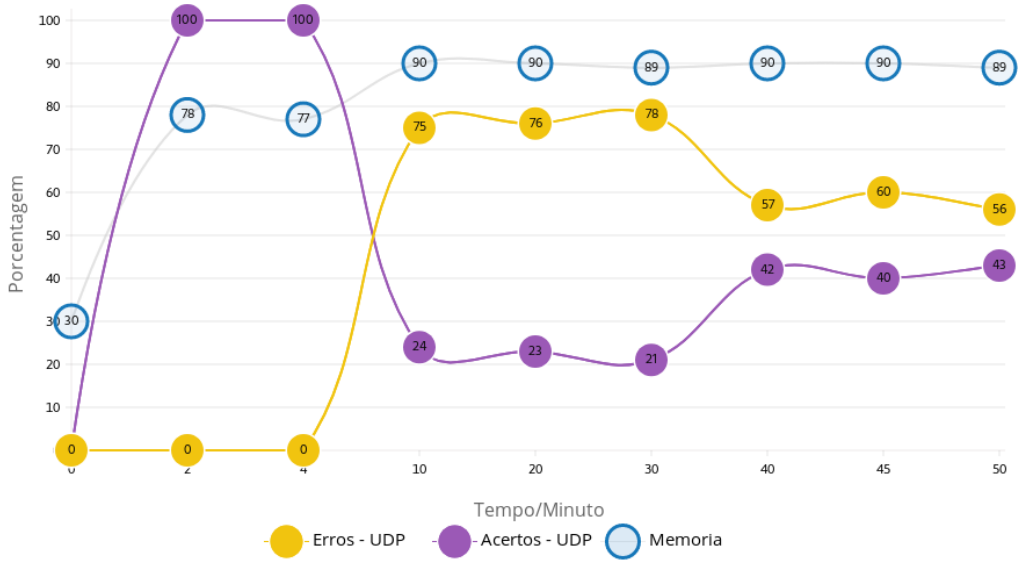


Tabela 3.8: Tabela do consumo de memória durante o ataque *UDP Flood*.

Tempo/Min	0	2	4	10	20	30	40	45	50
Memoria	30,00%	78,7%	77,30%	90,84%	90,30%	89,45%	90,14%	90,37%	89,92%
Acertos - UDP	0%	100%	100%	24,5%	23,7%	21,7%	42,3%	40,0%	43,0%
Erros - UDP	0%	0%	0%	75,5%	76,3%	78,3%	57,7%	60,0%	56,8%

A Figura 3.12 apresenta o tráfego *Transmission Control Protocol* (TCP) durante o período do ataque *UDP Flood*. Observa-se nos dados da Tabela 3.9 que ocorre um fluxo mais intenso na rede TCP, identificando desta forma possíveis atrasos na troca de informações e pacotes entre os nós da rede Blockchain.

Figura 3.12: Análise de tráfego TCP durante o ataque *UDP Flood*

Tabela 3.9: Análise de tráfego TCP durante o ataque *UDP Flood*.

Tempo/Min	0	2	4	10	20	30	40	45	50
Entrada/seg	0,0	15,77	11,60	1195,7	1361,83	919	128	12,13	9,0
Saida/seg	0,0	19,57	14,5	4691,03	5241,37	3132,47	9,83	13,37	8,10

Este experimento permite revelar algumas das implicações do ataque em relação ao desempenho e operação do Blockchain Ethereum. Com um esforço limitado (uso de poucos nós equivalentes) por parte do atacante, é possível detectar uma redução considerável na taxa de mineração e também impacto na troca de informações da VM-1, atacada, com os outros nós pertencentes a rede.

3.5.2. Cenário 2

Este cenário de experimento utiliza a plataforma Multichain com a versão 1.0.5 e possui seis VMs que utilizam sistemas operacionais GNU/Linux Ubuntu Server 16.02 LTS. O Blockchain Multichain é um sistema projetado para sua aplicação com o Modelo Privado e Consórcio. O Multichain utiliza uma rede parcialmente descentralizada, que possui a existência de um administrador criador do Bloco Gênesis, em que este possui o poder de permitir diferentes conexões na rede Blockchain e que também permite o recebimento, envio e mineração de blocos na rede. As VMs aplicadas no ambiente utilizam o protocolo de rede TCP para realizarem trocas de informações entre si. A Listagem 3.3 apresenta as principais informações sobre a rede Blockchain Multichain.

Listagem 3.3: Informações da Rede

```
1  {"method":"getinfo","params":[],"id":1,"chain\_name":"chain1"}
2  {
3  "version" : "1.0.5",
4      "nodeversion" : 10005901,
5      "protocoloversion" : 10011,
6      "chainname" : "chain1",
7      "description" : "MultiChain chain1",
8      "protocol" :    "multichain",
9      "port" : 7317,
10     "setupblocks": 60,
11     "nodeaddress" : "chain1@10.0.0.3:7317",
12     "burnaddress" : "1XXXXXXXGiXXXXXXTNXXXXXX64fp8",
13     "incomingpaused" : false,
14     "mininpaused" : false,
15     "walletversion" : 60000,
16     "balance" : 0.00000000,
17     "walletdbversion" : 2,
18     "reindex" : false,
19     "blocks" : 507,
20     "timeoffset" : 0,
21     "connections" : 6,
22     "proxy" : "",
23     "difficulty" : 0.00000006,
24     "testnet" : false,
25     "keypoololdest" : 1557377147,
26     "keypoolsize" : 2,
27     "patyxfee" : 0.0000000,
28     "relayfee" : 0.0000000,
29     "errors" : ""
30 }
```

O método **getinfo**, na linha 1 da Listagem 3.3, apresenta as informações relevantes sobre o atual estado da rede. É possível observar, linha 11, o IP interno da máquina

seguido pela porta utilizada para a conexão, outra informação relevante é sobre as carteiras existentes na rede para realizarem transações. Na Listagem 3.3 é observado na linha 19 a quantidade de blocos que foram minerados e também na linha 21 a quantidade de nós conectados a rede.

A partir da Listagem 3.3 observa-se a existência de duas carteiras na rede, que interagem entre si. A VM-1 e VM-6 possuem propriedades sobre estas carteiras e como experimento de funcionamento foi realizada uma transação da carteira da VM-6 para a carteira da VM-1 que pode ser identificada na Listagem 3.4.

Listagem 3.4: Informações de verificação de uma transação realizada

```
1  {"method":"getwallettransaction","params":["21a38f4784ad02636ce421369f8af805b929
       92aad6edeca44771e711b8553d59"],"id":1,"chain_name":"chain1"}
2      {
3          "balance" : {
4              "amount" : 0.00000000,
5              "assets" : [
6                  {
7                      "name" : "assets1",
8                      "assetref" : "71-265-5957",
9                      "qty" : -300.00000000
10                 }
11             ]
12         },
13         "myaddresses" : [
14             "1JBQVuzEZRzQhJwUuNARwc2oA2KTX82189iokZ"
15             ],
16         "addresses" : [
17             "1T5wAoTzv5TzZFnLdDPgWeU7DWci2HiBWABMwm"
18         ],
19         "permissions" : [
20         ],
21         "items" : [
22         ],
23         "data" : [
24         ],
25         "confirmations" : 505,
26         "blockhash" : "00418a75ebb5723b484b8dca00e9373983a953f3fdb7d0ee1b4a09946
               742fefd",
27         "blockindex" : 1,
28         "blocktime" : 1558627823,
29         "txid" : "21a38f4784ad02636ce421369f8af805b92992aad6edeca44771e711b8553d
               59",
30         "valid" : true,
31         "time" : 1558627808,
32         "timereceived" : 1558627808
33 }
```

Na Listagem 3.4 é possível observar informações da transação de VM-6 para VM-1. As linhas 15 e 17 visualiza-se os endereços das carteiras e também é possível observar informações como quantidade de confirmações, valor e data que a transferência ocorre e também outras informações relevantes para garantir a contabilidade da rede.

Listagem 3.5: Blocos que foram minerados e inseridos pela rede Blockchain

```
1      {
2          "hash" : "00d27a5d98675d6eb217160bc88f678e2f647e69fdbf3c5c46a5d8ed171e74
               3c",
3          "miner" : "1T5wAoTzv5TzZFnLdDPgWeU7DWci2HiBWABMwm",
4          "confirmations" : 185,
5          "height" : 828,
6          "time" : 1558903615,
7          "txcount" : 1
```



```
8       },
9       {
10          "hash" : "00d7f0c124e7cb850c92eb305a2f8ea975e9d776f10f9fe825073a2e97eaa2
                7d",
11          "miner" : "1JBQVuzEZRzQhJwUuNARwc2oA2KTX82189iokZ",
12          "confirmations" : 184,
13          "height" : 829,
14          "time" : 1558903615,
15          "txcount" : 1
16      },
17      {
18          "hash" : "00435d2ce2e2c426c63197adce3d3a046da2e6f6a1cc5da4a5c8ea47e51fdf
                89",
19          "miner" : "1T5wAoTzv5TzZFnLdDPgWeU7DWci2HiBWABMwm",
20          "confirmations" : 183,
21          "height" : 830,
22          "time" : 1558903646,
23          "txcount" : 1
24      },
25      {
26          "hash" : "0031338db14d866b306d451c195ce4067adc0dde5dfeedb124239627e753
                bdde",
27          "miner" : "1bABXfY7cshesG8d1UwTZdxiM1ehT9yY6Zh4Jw",
28          "confirmations" : 182,
29          "height" : 831,
30          "time" : 1558903654,
31          "txcount" : 4
32  }
```

A Listagem 3.5 possibilita a identificação e funcionamento quanto a mineração dos blocos realizados pela rede Blockchain. O Blockchain Multichain utiliza o mecanismo de consenso PBFT e como algoritmo de escalonamento de prioridades o sistema utiliza o *round-robin*. Uma outra averiguação que a Listagem 3.5 permite fazer é sobre o funcionamento do mecanismo, em que apresenta diferentes mineradores em sequência, observados no código pelas linhas 3, 11, 19 e 27, na mineração de novos blocos para a rede.

Com as informações apresentadas é possível averiguar o estado de funcionamento da arquitetura e comportamento da rede, sendo esta uma rede Blockchain com modelo privado e parcialmente descentralizada. A rede é composta por seis nós, sendo um deles o administrador, que delimita as permissões para cada uma destas VMs. Quanto ao sistema, é necessário avaliar seu desempenho a partir da realização de um ataque DoS em sua rede, com objetivo de avaliar o comprometimento da rede durante e após o ataque.

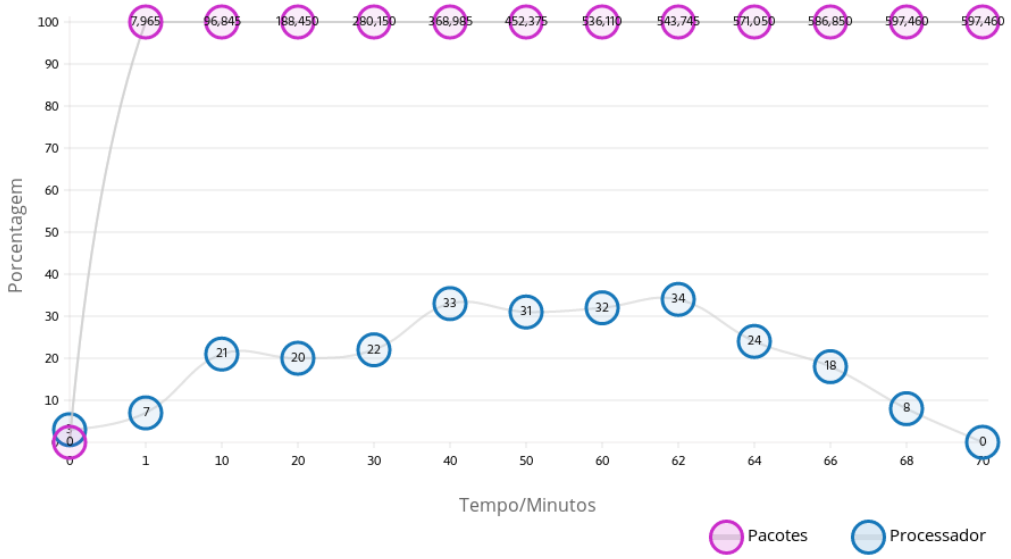
3.5.2.1. Ocorrência do Ataque de Negação de Serviço

A solução de Blockchain Multichain utiliza o protocolo de rede TCP. Como aplicação destes ataques foram escolhidos os ataques: *SSH Flood* e o *Transaction Flood*.

O ataque SSH foi realizado a partir da VM-6 endereçado para a VM-1. A partir deste ataque foi observado que não causava nenhum dano na VM-1 da rede Blockchain Multichain. Não foram identificados alterações de processamento e nem do uso de memória que causassem impactos negativos para o nó ou propriamente para a rede. O principal motivo pela não ocorrência de nenhum efeito, é a necessidade de aplicação de cargas no ataque.

Quanto ao ataque *Transaction Flood* foi realizado a partir da carteira VM-6 que realizava transações para a carteira pertencente a VM-1. O mecanismo de consenso PBFT é um mecanismo com baixo consumo de CPU, ou seja, não necessita de considerável uso de processamento para validação de transações ou criação de novos blocos. Este comportamento pode ser observado a partir da Figura 3.13 durante o período que a VM recebeu os ataques gerados pela VM-6.

Figura 3.13: Análise do consumo de processador durante a o ataque *Transactions Flood*.



A partir da Figura 3.13 e Tabela 3.10 é possível observar um baixo crescimento no uso de CPU durante o processo de ataque. Na Figura 3.13 possui alguns pontos em que há pequenas alterações durante a execução do ataque, isso deve-se ao fato de ocorrer retransmissões TCP que foram captadas durante a execução do ataque.

Tabela 3.10: Tabela do consumo de processador durante o ataque *Transaction Flood*.

Tempo/Min	0	1	10	20	30	40	50	60	62	64	66	68	70
Processador	3,3%	7,25%	21,76%	20,23%	22,94%	33,00%	31,09%	32,69%	34,08%	24,40%	18,25%	8,07%	0,9%
Pacotes	0	7965	96845	188450	280150	368985	452375	536110	543745	571050	586850	597460	597460

Na Figura 3.14 e Tabela 3.11 observa-se um expressivo impacto do ataque no uso da memória da VM. A partir da Figura 3.14 que a ponto que novas transações eram criadas havia um crescimento significante na quantidade de memória que era consumida pela rede Blockchain. Há alguns pontos em que a memória subitamente reduzia a taxa de uso, existe a necessidade de uma análise mais profunda para compreender o motivo desta redução, uma possível explicação seria o descarte em blocos realizado pela plataforma.

Um ponto relevante é que após uma hora de ataque o sistema apresentou estouro/-falta de memória. A falta de memória implica no encerramento abrupto da aplicação e a partir deste encerramento foi percebido que do montante de transações enviadas entorno de 10% à 15% foram validadas pela aplicação, sendo as demais foram perdidas.

Figura 3.14: Análise do consumo de memória durante o ataque *Transactions Flood*.

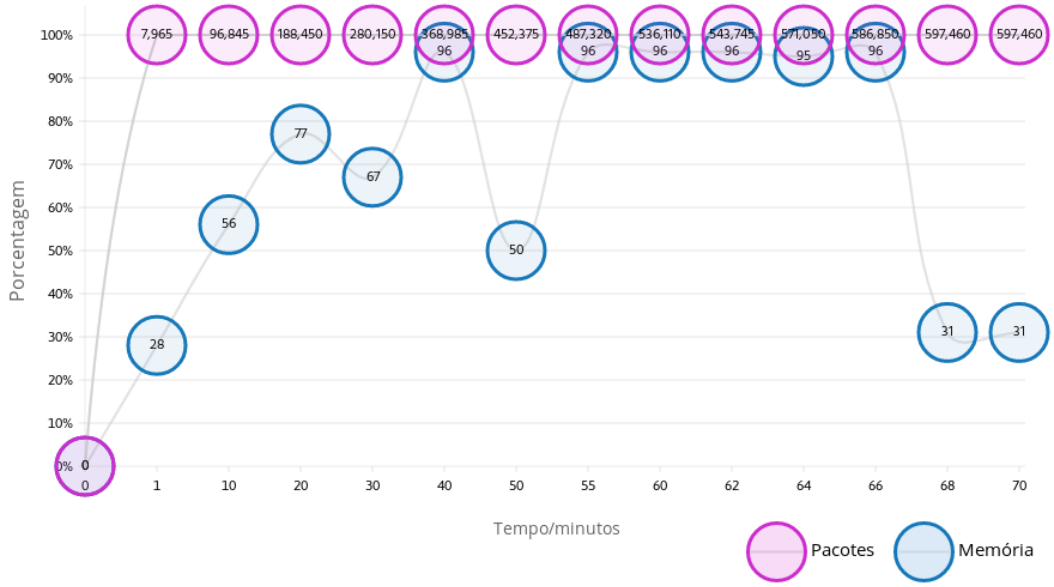


Tabela 3.11: Tabela do consumo de memória durante o ataque *Transaction Flood*.

Tempo/Min	0	1	10	20	30	40	50	55	60	64	66	68	70
Memória	0%	28,65%	56,28%	77,06%	67,09%	96,14%	50,57%	96,16%	96,18%	95,84%	96,72%	31,15%	31,15%
Pacotes	0	7965	96845	188450	280150	368985	452375	487320	536110 §	571050	586850	597460	597460

Na Figura 3.15 é observado o tráfego TCP durante o período da realização do ataque *Transactions Flood*. É possível verificar, a partir dos dados da Tabela 3.12, que a aplicação estava enviando muito mais dados que recebendo, demonstrando atrasos na rede Blockchain e também perda de transações e pacotes.

Figura 3.15: Análise de tráfego TCP durante o ataque *Transaction Flood*.

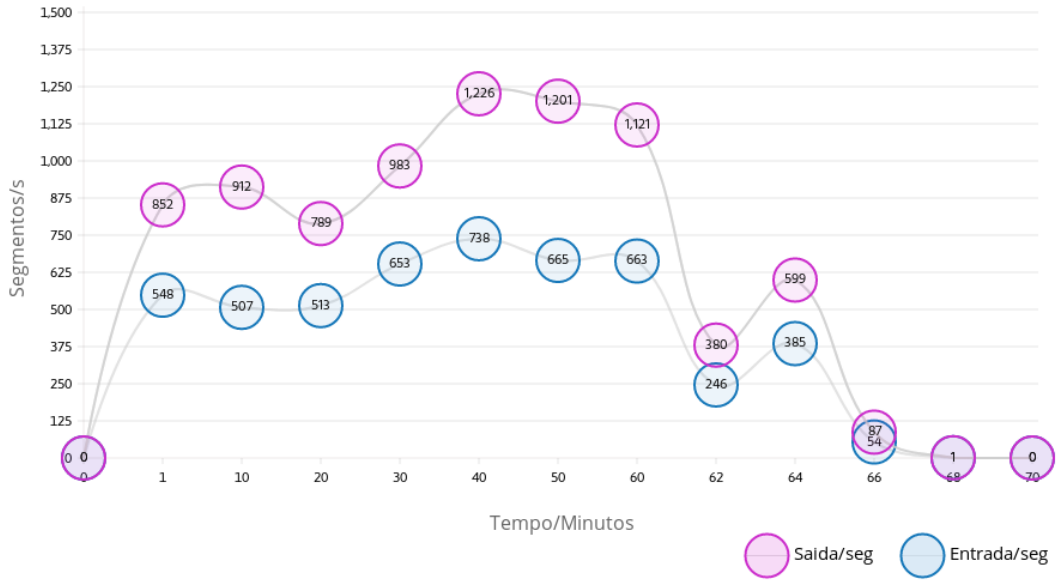


Tabela 3.12: Tabela do tráfego TCP durante o ataque *Transaction Flood*.

Tempo/Min	0	1	10	20	30	40	50	55	60	64	66	68	70
Entrada/seg	0	548,78	507,50	513,38	653,85	738,33	665,62	663,38	246,78	385,97	54,15	1,23	0,85
Saida/seg	0	852,33	912,08	789,08	983,45	1226,33	1201,48	1121,55	380,57	599,48	87,25	1,35	0,85

A partir deste experimento é observado as implicações do ataque *Transactions Flood* no desempenho da rede Blockchain. Este cenário indica uma fragilidade de uma rede parcialmente descentralizada e privada perante seus pares que também pertencem a mesma rede. Afinal, com apenas um atacante foi possível realizar estouro de memória no nó que administra a rede, causando instabilidade de modo geral na rede Blockchain

3.5.3. Análises dos resultados

Com base na experimentação realizada, é possível observar várias potenciais vulnerabilidades e ameaças que podem ser exploradas nas redes tipo Blockchain. Os resultados desta análise são condizentes com as necessidades que foram abordadas por [50]. O trabalho apresentam a questão de Ataques de Negação de Serviço como um ataque que além de causar danos para a rede de Blockchain também influencia problemas em sistemas que podem ser aplicados conjuntamente. Contudo, para estas vulnerabilidades existe a possibilidade de mitigação dos impactos dos ataques através de mecanismos de segurança e flexibilização na configuração do sistema. Para consolidação dos resultados obtidos, a Tabela 3.13 apresenta a relação entre os critérios analisados, os problemas encontrados relacionados a este critérios e possíveis boas práticas para minimizar os problemas.

Tabela 3.13: Visão geral dos resultados da análise do trabalho.

Critérios	**Consenso**	**Problemas**	**Boas práticas**
Procedência dos dados	PoW	Não apresentou problemas em relação a segurança dos dados	Realizar verificações de transações e blocos para garantia contínua da seguridade
	PBFT	Problemas com validação de Transações, Possibilidade de alteração do conteúdo das transações	Monitoramento CPU/Memória
Controle e Gerenciamento do Nó	PoW	Problemas com Mineração, Tráfego intenso em seus protocolos	Monitoramento de tráfego interno, Monitoramento de processamento
	PBFT	Problemas com administração da rede Blockchain	Monitoramento de transações

Para o critério de procedência dos dados foram encontrados problemas relacionados a validação de transações e blocos, gerados por dificuldades de mineração ou queda do sistema. Quanto a esta procedência dos dados, existe uma outra questão relevante apresentadas por Blockchain privados que é a questão do anonimato. Esta vulnerabilidade é conhecida como Privacidade de Transações, é considerada uma vulnerabilidade para uma rede Blockchain, mas é interessante para instituições que desejam auditar seu sistema e detectar possíveis causadores de problemas em sua rede. Uma possível solução encontrada para ambos os casos, é a realização de monitoramentos de CPU e memória e também monitoramento de violações da rede interna, que causam danos mais graves ao funcionamento do sistema.

No critério de controle e gerenciamento do nó, os principais problemas encontrados foram relacionados a mineração, tráfego intenso em suas redes internas e por parte do Blockchain parcialmente descentralizado a questão de administração da rede Blockchain. Estes problemas relacionados ao controle e gerenciamento dos nós abrem portas para diversas outras vulnerabilidades, principalmente vinculadas a rede e poder computacional, Seção 3.2. Como possível boa prática para evitar que estas vulnerabilidades sejam exploradas é interessante que haja monitoramento de tráfego, transações e de processamento.

3.6. Considerações

Em relação às versões, modelos, mecanismos de consenso do Blockchain sobressai-se a preocupação em apresentar as principais características envolvidas. A partir desta preocupação, a realização de comparações entre os modelos, algoritmos de consenso, versões entre outras características para maior entendimento das diversas possibilidades de combinações para aplicações do Blockchain. Sobre as características, foram apresentadas sistematicamente os principais meios, ou plataformas, em que a tecnologia Blockchain se faz presente sua utilização. além disso, são citadas as três plataformas mais utilizadas no uso de contratos inteligentes e aplicativos descentralizados que são: Ethereum, Hyperledger e o MultiChain. Em cada uma destas plataformas foram elencadas características da aplicação e realizado comparações entre elas, para ficar mais claro as necessidades e conhecimentos necessários para uso das mesmas.

Relacionado a preocupação com a segurança do Blockchain, o estudo apresentado inclui uma proposta de classificação das principais ataques e vulnerabilidades podem afetar as implementações do conceito de Blockchain. Neste estudo foram elencadas algumas vulnerabilidades do Blockchain que são: DoS, Ataque Eclipse, Ataque de Vivacidade, Gasto Duplo, Mineração Egoísta, Otimização de Contratos Inteligentes, Privacidade de Transações, Retenção de Blocos e Vulnerabilidade 51%. Devido a não ser identificado pelos autores um padrão para classificar estes ataques/vulnerabilidades, estes foram classificados em três grupos: Redes, Poder Computacional e Usuário.

De um modo geral, quanto ao uso dos mecanismos de consenso, é perceptível que há possibilidade, através das boas práticas, destes problemas serem minimizados. Contudo, por mais críticas que sejam as vulnerabilidades existentes, o uso da tecnologia Blockchain tem crescido e mostrado ao mercado que o investimento e desenvolvimento da tecnologia, assim como de seus métodos, pode proporcionar diversos benefícios às instituições envolvidas.

Agradecimentos

O presente trabalho foi em parte financiado pelo CNPq, através das Bolsas de Produtividade em Pesquisa 301198/2017-9 e Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora 311667/2014-7.

Os autores agradecem o apoio do Laboratório de Arquitetura e Redes de Computadores (LARC) e Laboratório de Sustentabilidade (LASSU) do Departamento de Engenharia de Produção e Sistemas Digitais (PCS) da Escola Politécnica da Universidade de São Paulo (USP).

Os autores agradecem o apoio do Laboratório de Processamento Paralelo e Distribuído (LabP2D) no Centro de Ciências tecnológicas (CCT) da Universidade do Estado de Santa Catarina (UDESC).

Esse trabalho foi financiado com recursos da Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC).

Referências

[1] M. Swan, *Blockchain: Blueprint for a New Economy*. "O'Reilly Media, Inc.", Jan. 2015.

[2] P. Tasca and C. Tessone, "A taxonomy of blockchain technologies: Principles of identification and classification," *Ledger*, vol. 4, no. 0, 2019. [Online]. Available: http://www.ledgerjournal.org/ojs/index.php/ledger/article/view/140

[3] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352864818301536

[4] B. Rodrigues, E. J. Scheid, R. Blum, T. Bocek, and B. Stiler, "Blockchain and Smart Contracts – From Theory to Practice," in *Tutorials of IEEE International Conference on Blockchain and Cryptocurrency*. Seoul, South Korea: IEEE Computer Society Press, May 2019, p. 31. [Online]. Available: https://files.ifi.uzh.ch/CSG/staff/rodrigues/extern/publications/CNSM18-Tutorial.pdf

[5] I.-C. Lin and T.-C. Liao, "Survey of blockchain security issues and challenges," *International Journal of Network Secutiry*, 2017.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *International Journal of Network Secutiry*, 2008. [Online]. Available: "https://bitcoin.org/bitcoin.pdf"

[7] T. E. Foundation, "Ethereum homestead documentation," 2018.

[8] Hyperledger, "An introduction to Hyperledger," White Paper. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf, 2018.

[9] G. Greenspan, "Multichain private blockchain – white paper," www.multichain.com/download/MultiChain-White-Paper.pdf, 2015.

[10] Plurasight, "Blockchain architecture," 2017. [Online]. Available: https://www.pluralsight.com/guides/blockchain-architecture

[11] NIST, *FIPS 180-4: Secure Hash Standard (SHS)*, National Institute of Standards and Technology, Gaithersburg, MD, USA, August 2015.

[12] R. C. Merkle, "Protocols for public key cryptosystems," in *IEEE Symposium on Security and Privacy*, April 1980, pp. 122–134.

[13] J. Garay and A. Kiayias, "SoK: A consensus taxonomy in the blockchain era," Cryptology ePrint Archive, Report 2018/754, 2018, https://eprint.iacr.org/2018/754.

[14] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, ser. PODC '87. New York, NY, USA: ACM, 1987, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/41840.41841

[15] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef, and E. Zenner, "Ripple: Overview and outlook," in *Trust and Trustworthy Computing*, M. Conti, M. Schunter, and I. Askoxylakis, Eds. Cham: Springer International Publishing, 2015, pp. 163–180.

[16] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proc. of the 3rd Symposium on Operating Systems Design and Implementation (OSDI)*. Berkeley, CA, USA: USENIX Association, 1999, pp. 173–186.

[17] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 104–121.

[18] R. Patterson, "Alternatives for Proof of Work, Part 2: Proof of Activity, Proof of Burn, Proof of Capacity, and Byzantine Generals — Bytecoin Blog," Mar. 2016. [Online]. Available: https://web.archive.org/web/20160304055454/https://bytecoin.org/blog/proof-of-activity-proof-of-burn-proof-of-capacity/

[19] G. Kostarev, "Review of blockchain consensus mechanisms," Jul. 2017. [Online]. Available: https://blog.wavesplatform.com/review-of-blockchain-consensus-mechanisms-f575afae38f2

[20] J. Kwon, "Tendermint: Consensus without mining - v0.6," White Paper. Available: http://docplayer.net/50173080-Tendermint-consensus-without-mining.html, 2014.

[21] V. Buterin, "On public and private blockchains," 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

[22] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[23] namecoin, "Name coin," 2018. [Online]. Available: https://namecoin.org/

[24] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*. Berlin, Heidelberg: Springer, 2001, pp. 46–66.

[25] J. Benet, "IPFS: content addressed, versioned, P2P file system," arXiv preprint arXiv:1407.3561, 2014, see also https://ipfs.io.

[26] E. Larcheveque, A. Caswell, and A. Ferron, "BitID: Bitcoin authentication open protocol," https://github.com/bitid/bitid, 2016.

[27] G. Wood, "Polkdaot: Vision for a Heterogeneous Multi-Chain Framework," November 2016, Accessed: 2019-07-04. [Online]. Available: https://polkadot.network/PolkaDotPaper.pdf

[28] Aelf, "aelf - A Multi-Chain Parallel Computing Blockchain Framework," June 2018, Accessed: 2019-07-04. [Online]. Available: https://aelf.io/gridcn/aelf_whitepaper_EN.pdf?v=1.6

[29] F. Greve, L. Sampaio, J. Abijaude, A. A. Coutinho, I. Brito, and S. Queiroz, *Blockchain e a Revolução do Consenso sob Demanda*. Sociedade Brasileira de Computação (SBC), 2018, ch. Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC).

[30] B. Curran, "What is delegated proof of stake consensus? (DPoS) complete beginner's guide," 2018.

[31] M. E. Peck, "Blockchains: How they work and why they'll change the world," *IEEE Spectrum*, vol. 54, no. 10, pp. 26–35, October 2017.

[32] BitCoin.org, "How does Bitcoin work?" Bitcoin Official website: https://bitcoin.org/en/how-it-works, 2018.

[33] G. Greenspan, "(white paper) decentralized financial system – Credits – v 2.1," Credits, Tech. Rep., 2018, available: credits.com/Content/Docs/TechnicalWhitePaperCREDITSEng.pdf.

[34] BR, "Ethereum white paper made simple," Blockchain Review, Tech. Rep., 2018, available: https://cryptoverze.com/wp-content/uploads/2018/11/02.01._final_Ethereum-White-Paper-Made-Simple.pdf.

[35] V. Buterin, "Ethereum white paper – a next generation smart contract & decentralized application platform," Ethereum.org, Tech. Rep., 2018, available: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.

[36] G. Greenspan, "Multichain private blockchain," 2018.

[37] B. Rodrigues, T. Bocek, and B. Stiller, "Enabling a cooperative , multi-domain DDoS defense by a Blockchain signaling system (BloSS)," in *Proc. of the 42nd IEEE Conference on Local Computer Networks 2017 (LCN) – Demos*, 2017, pp. 1–3.

[38] A. Singh, T. Ngan, P. Druschel, and D. S. Wallach, "Eclipse Attacks on Overlay Networks: Threats and Defenses," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, Apr. 2006, pp. 1–12.

[39] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, "Security implications of Blockchain cloud with analysis of block withholding attack," in *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2017.

[40] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, Aug. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X17318332

[41] E. Le Jamtel, "Swimming in the monero pools," in *2018 11th International Conference on IT Security Incident Management IT Forensics (IMF)*, May 2018, pp. 110–114.

[42] I. Eyal, "The Miner's Dilemma," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 89–103.

[43] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *CoRR*, vol. abs/1311.0243, 2013. [Online]. Available: http://arxiv.org/abs/1311.0243

[44] A. Kiayias and G. Panagiotakos, "On trees, chains and fast transactions in the blockchain," Cryptology ePrint Archive, Report 2016/545, 2016, https://eprint.iacr.org/2016/545.

[45] T. Chen, X. Li, X. Luo, and X. Zhang, "Under-Optimized Smart Contracts Devour Your Money," *arXiv:1703.03994 [cs]*, Mar. 2017, arXiv: 1703.03994. [Online]. Available: http://arxiv.org/abs/1703.03994

[46] P. Mell and T. Grance, "(SP 800-145) the NIST definition of cloud computing," National Institute of Standards & Technology, Gaithersburg, MD, United States, Tech. Rep., 2011.

[47] S. Shetty, V. Red, C. Kamhoua, K. Kwiat, and L. Njilla, "Data provenance assurance in the cloud using blockchain," in *Disruptive Technologies in Sensors and Sensor Systems*, vol. 10206. International Society for Optics and Photonics, May 2017, p. 102060I. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10206/102060I/Data-provenance-assurance-in-the-cloud-using-blockchain/10.1117/12.2266994.short

[48] S. Report, "Funcionários são responsáveis por nove em cada dez violações de dados na nuvem," May 2019. [Online]. Available: http://www.securityreport.com.br/overview/funcionarios-sao-responsaveis-por-nove-em-cada-dez-violacoes-de-dados-na-nuvem/

[49] G. Zyskind, O. Nathan, and A. . Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 180–184.

[50] A. Rot and B. Blaicke, "Blockchain's future role in cybersecurity. analysis of defensive and offensive potential leveraging blockchain-based platforms," in *9th International Conference on Advanced Computer Information Technologies (ACIT)*, 2019.

[51] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982, availble: people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf.

