UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO - CTC
SISTEMAS DE INFORMAÇÃO

CINTHIA CAROLINA SHIRATORI

SEGURANÇA DE PROTOCOLOS DE ROTEAMENTO COM CUSTOS OPERACIONAIS ASSOCIADOS À BLOCKCHAINS NAS REDES DE NOVA GERAÇÃO

Florianópolis
2024

CINTHIA CAROLINA SHIRATORI

SEGURANÇA DE PROTOCOLOS DE ROTEAMENTO COM CUSTOS OPERACIONAIS ASSOCIADOS À BLOCKCHAINS NAS REDES DE NOVA GERAÇÃO

Trabalho apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação, no Centro Tecnológico - CTC, da Universidade Federal de Santa Catarina.

Orientador(a): Prof.º DR.º Carlos Becker Westphall

Coorientador(a): Rodolfo Borges dos Santos de Carvalho

Florianópolis

2024

CINTHIA CAROLINA SHIRATORI

SEGURANÇA DE PROTOCOLOS DE ROTEAMENTO COM CUSTOS OPERACIONAIS ASSOCIADOS À BLOCKCHAINS NAS REDES DE NOVA GERAÇÃO

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Sistemas de Informação, no Centro Tecnológico - CTC, da Universidade Federal de Santa Catarina.

Florianópolis (SC), 18 de novembro de 2024.

Banca Examinadora:

Prof. Dr. Carlos Becker Westphall
Orientador

Profª. Drª. Carla Merkle Westphall
Membro(a)
UFSC

Doutorando Rodolfo Borges dos Santos de Carvalho
Membro(a)
UFSC

Este trabalho de conclusão de curso é dedicado
a minha família e amigos, por sempre acreditarem no meu potencial,
especialmente ao meu querido pai, cuja sabedoria, força e dedicação
sempre foram fontes de inspiração ao longo dessa jornada.

AGRADECIMENTOS

Agradeço a Deus, pela força e serenidade que me guiaram ao longo desta caminhada, e por me permitir chegar até aqui, superando desafios e celebrando cada conquista.

Aos meus queridos pais Milton Tadashi Shiratori e Ana Lucia Fagundes Shiratori, pelo amor incondicional e pelo apoio constante. Principalmente ao meu querido pai, Gudo, que sempre me orientou e acreditou em mim, sem medir esforços para apoiar meus estudos e me mostrar o caminho.

À minha querida irmã Ana Paula Shiratori por ser minha amiga, pelo carinho, companheirismo e incentivo em todos os momentos.

Às amigas de longa data, Monique Jocken, Fernanda Zarth e Maiara Aguiar, que muitas vezes à distância permaneceram presentes, me inspirando com a constância de nossa amizade.

Aos amigos que fiz durante esta jornada acadêmica, que tornaram os dias mais leves, compartilhando aprendizado e tornando essa trajetória mais rica e significativa.

Ao professor Carlos, meu orientador, pelas orientações e pelo comprometimento com o meu crescimento acadêmico e profissional. Ao Rodolfo, meu coorientador, pela dedicação valiosa, pelos inúmeros conselhos e ensinamentos.

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho, muito obrigada!

“ A única maneira de fazer um excelente trabalho é amar o que você faz ”

Steve Jobs

RESUMO

As redes de malha sem fio (WMNs) enfrentam desafios de segurança que afetam sua expansão e confiabilidade. A implementação da tecnologia Blockchain surge como uma solução promissora para mitigar essas vulnerabilidades, garantindo autenticidade e integridade nas transações de roteamento. A proposta de realizar uma análise aprofundada do desempenho do protocolo Babel em Redes Mesh Sem Fio (WMNs) visa avaliar sua eficiência e eficácia para futura implementação de identidades digitais únicas e imutáveis para os nós da rede, contribuindo para o fortalecimento da segurança e resiliência do sistema. Essa abordagem não apenas fortalece a autenticidade dos nós, mas também contribui para a integridade geral das comunicações na WMN, essencial para a segurança e eficiência das operações em ambientes sem fio. Essa solução representa um avanço significativo no fortalecimento da segurança das WMNs, pavimentando o caminho para uma expansão mais segura e confiável dessas redes em todo o mundo.

Palavra-chave: Blockchain. Wireless. Mesh. Network. Segurança.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A expansão das redes e de sua infraestrutura tem sido um princípio norteador do desenvolvimento econômico e social [Telecomunicações 2022]. A ampliação no acesso à rede, principalmente em áreas onde a oferta é inadequada, permite que diversos setores de uma sociedade sejam melhorados, como a saúde, educação e segurança para toda a população atendida [Telecomunicações 2022]. Um dos grandes desafios é levar a conexão de rede para áreas urbanas desatendidas, áreas rurais, e também áreas remotas onde se encontram algumas comunidades.

Os altos custos para implantação da rede cabeada, ainda no ano de 2022, justificam a carência de acesso estável à rede de internet por 37,5% da população mundial [Reportal 2022]. Dentro de áreas mais remotas do território, tais problemas de custos também são somados à qualidade precária das redes existentes, e a falta de sustentação e manutenção dos serviços a longo prazo.

A falta de acesso a rede em áreas remotas é mais evidente em países em desenvolvimento, que possuem um vasto território, como é o caso do Brasil. Considerado o quinto maior país do mundo em extensão territorial, é também o terceiro país com o maior tempo médio de acesso a rede no mundo [Maccari e Lo Cigno 2015], mas ainda apresenta uma imensa desigualdade no acesso da rede. Em áreas urbanas segundo censo feito em 2020 [Union 2020], cerca de 14% dos lares brasileiros não tinham conexão de rede, já em áreas rurais a situação era ainda pior, com cerca de 35% das residências não possuindo conexão de rede.

Dentro do contexto de desigualdade, as WMN (Wireless Mesh Networks) são redes capazes não apenas de fornecer conectividade sem fio a largas áreas (como shoppings e centros urbanos), mas também levar tal conectividade a locais muito mais remotos do território [Akyildiz, Wang e Wang 2005]. Por conta da redução de preços dos aparelhos e facilidade em sua implantação, as WMNs levaram ao surgimento de diversas redes comunitárias [Baig et al. 2015], que ajudam a resolver o problema de última milha, relacionado a áreas mais remotas que não possuem acesso à rede.

A fim de garantir a segurança das WMNs é possível associar a tecnologia blockchain e seus três princípios fundamentais de descentralização, imutabilidade e consenso, aplicados para fortalecer a integridade e a confiabilidade da rede e comunicação, buscando criar uma infraestrutura mais resistente e capaz de enfrentar os desafios de segurança nas redes mesh sem fio.

A arquitetura das WMNs promove a descentralização ao distribuir a responsabilidade entre múltiplos nós da rede, reduzindo pontos únicos de falha e aumentando a resiliência e a confiabilidade do sistema, isso reduz a vulnerabilidade a ataques únicos, pois não há um único ponto central que, se comprometido, possa comprome-

ter toda a rede. Já a imutabilidade, no contexto blockchain, garante que os dados são registrados em um bloco e adicionados à uma cadeia, que criam uma conexão sequencial. Assim, qualquer tentativa de manipulação pode ser prontamente identificada, assegurando mais uma vez a integridade dos dados. E o consenso garante que cada operação incluída na blockchain seja validada por vários usuários promovendo a segurança entre nós maliciosos, responsável, também, por tornar a rede confiável.

Entretanto, dentro da área de segurança ainda é possível identificar diversos pontos de vulnerabilidade em ambientes de WMNs, quando utilizada a tecnologia blockchain para compensações financeiras entre os participantes da rede. Os principais pontos associados aos resultados esperados estão atrelados a correção de possíveis vulneabilidades nesse ambiente como, por exemplo, comportamentos egoístas que levam a situações como free-riding, buracos negros (black-holes) com perdas constantes de pacotes que atrasam a comunicação, vandalismos e também ataques que exploram vulnerabilidades dos protocolos, que dificultam a estabilidade da rede e a confiabilidade entre participantes que cooperam com o ambiente de WMN.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo geral deste trabalho é analisar o desempenho do protocolo Babel em Redes Mesh Sem Fio (WMNs), comparando a versão original com a versão Babel Cost, que incorpora uma variável de custo, a fim de avaliar sua eficiência, capacidade de roteamento e impacto na otimização da rede.

1.1.2 Objetivos Específicos

Os objetivos específicos que podem ser listados neste trabalho são os seguintes:

- Realizar uma revisão detalhada do estado da arte sobre o uso de blockchain em WMNs, com foco nos avanços, desafios e soluções propostas na literatura, a fim de fundamentar a análise crítica e a aplicação dessa tecnologia em redes sem fio.
- Adaptar e modificar o protocolo Babel original, incorporando métricas de custo para desenvolver a versão Babel Cost, visando aprimorar o desempenho e a eficiência em WMNs.
- Implementar cenários de teste para avaliar e comparar o desempenho das versões Babel original e Babel Cost, analisando aspectos como eficiência de roteamento, consumo de recursos e estabilidade da rede.
- Analisar as desvantagens identificadas nos dados obtidos a partir dos testes realizados, avaliando os impactos.

2 FUNDAMENTAÇÃO TEÓRICA

O trabalho realizado depende inicialmente de alguns conceitos e definições, que apesar de extensos e envolverem vários tópicos, foram concentrados em algumas áreas principais que serão abordadas no capítulo atual, formando assim a base essencial para o entendimento do assunto abordado

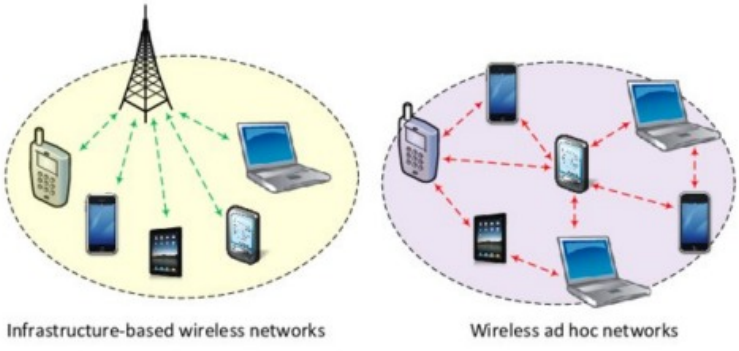
Dentre os principais assuntos, o capítulo atual começa abordando as Redes Ad Hoc e suas características principais que as fazem tão convenientes em ambientes sem cabeamentos, posteriormente é apresentada uma subárea dentro das redes Ad Hoc denominada por WMNs, considerada uma rede de nova geração, abordando também as suas especificidades.

2.1 REDES AD HOC

Redes ad hoc são redes sem fio descentralizadas onde cada nó se comunica diretamente com outros nós, formando uma rede dinâmica e auto-organizada sem necessidade de infraestrutura fixa, como roteadores ou pontos de acesso. para se comunicarem, não dispõem de infraestrutura cabeada associada aos mesmos (Figura 1), ou links de comunicação disponíveis de maneira organizada e pré-determinada. Segundo Ramanathan (2002), as redes ad hoc são caracterizadas pela sua capacidade de configuração automática e adaptação a mudanças na topologia, o que as torna ideais para ambientes onde a infraestrutura de rede tradicional não está disponível ou é impraticável.

.

Figura 1 – Comparação entre redes de Infraestrutura e redes Ad Hoc



Fonte: (DINH THAI et al., 2015)

Uma outra característica das redes ad hoc é a constante mudança nos links e na conectividade dos nós, causada pela mobilidade das conexões sem fio e pelos controles de consumo de energia. Com um maior número de participantes, a comunicação se torna mais complexa para garantir que cada pacote chegue ao seu destino. No entanto, essa complexidade não impede o uso da rede, especialmente em situações onde a infraestrutura de cabeamento não está disponível ou é pouco confiável, nessas circunstâncias, tecnologias sem fio, como infravermelho e radiofrequência (RF), são essenciais para a comunicação móvel em áreas remotas.

2.1.1 Redes Wireless Mesh Network

As WMNs (Wireless Mesh Networks) são redes ad hoc aprimoradas com características de topologia Mesh, permitindo uma comunicação auto-organizável e descentralizada entre os participantes. Isso favorece a comunicação em grandes áreas e a extensão da conectividade a participantes vizinhos, sem a necessidade de controle centralizado ou cabeamento conectando cada nó.

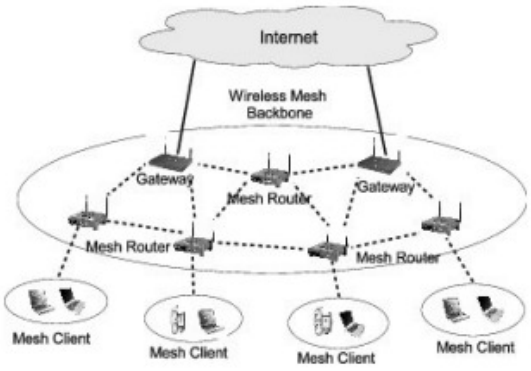
O roteamento existente dentro da rede permite que os nós, independentemente da sua capacidade e poder de processamento, possam encaminhar pacotes de dados entre nós que não estão dentro do alcance direto um do outro. Esse comportamento serve como uma ponte dentro da rede sem fio, a qual leva conectividade até o último nó possível, desde que exista ao menos um vizinho entre ele e o restante da rede.

A vantagem de redes com caminhos de múltiplos saltos se torna mais evidente se comparado às redes tradicionais cabeadas, utilizadas pelas grandes operadoras e através das quais todo pacote salta diretamente do usuário para a comunicação de links backhaul. Esses links são controlados pela operadora, e são responsáveis por realizar a comunicação entre os backbones e as pequenas sub-redes, muito presente em redes móveis e que acaba centralizando todas as transferências de dados.

Os benefícios de utilização das redes WMNs sempre estão vinculados ao ambiente em que são utilizadas, alcançando soluções para diversos desafios de implementação encontrados em redes sem fio.

.

Figura 2 – Wireless Mesh Network



Fonte: (JUNMO et al., 2006)

2.1.2 Segurança em WMNs

A segurança ainda é um campo aberto no que diz respeito às redes mesh, além dos problemas normais de segurança em redes sem fio tradicionais, ainda existe o problema de garantir a privacidade dos dados que estão trafegando entre os nós [Breuel 2004].

Nas WMNs, entender e abordar adequadamente esses problemas e desafios é muito necessário. Devido à mudança dinâmica da topologia da rede, à arquitetura de rede distribuída e aos meios sem fio compartilhados, as WMNs carecem de soluções de segurança. Nas redes mesh sem fio, existem diferentes tipos de arquitetura que podem usar abordagens distintas para a segurança das redes mesh.

Ainda, diferentes tipos de ataques podem ocorrer em diferentes camadas de rede, o que pode prejudicar o tráfego e os dados, caracterizando as ameaças relacionadas às WMNs.

2.1.2.1 Ameaça de free-riders

A ameaça do free-rider em redes Mesh sem fio refere-se a usuários ou dispositivos que utilizam a rede sem contribuir para seu custo, manutenção ou infraestrutura. Esses ”aproveitadores”conectam-se à rede para tirar vantagem de seus recursos sem a devida autorização ou sem fornecer algo em troca, o que pode causar diversos problemas de desempenho e segurança.

Assim, os free-riders são nós que consomem mas não doam recursos ao sistema, conhecidos também como ”nós de carona”.

2.1.2.2 Ameaça de Black Hole e Gray Hole

A ameaça do tipo Blackhole são ativos, pois o atacante descarta os pacotes que deveria encaminhar na rede. Ou seja, o invasor explora os procedimentos dos protocolos de descoberta de rota sob demanda. Quando um nó legítimo faz uma requisição de

descoberta de rota o invasor anuncia para a rede como sendo parte de uma rota mais recente, desta forma pode conseguir ser incluído na tabela de roteamento como parte integrante de várias rotas na rede. Uma vez incorporado a tabela de roteamento o nó invasor descarta os pacotes recebidos ao inves de retransmití-los [Kurosawa and Jamalipour, 2007]. Este comportamento provoca um ”buraco” na rede e consequantemente perda de informação. A severidade do ataque depende de quantas rotas o invasor foi bem sucedido em fazer parte.

Já a ameaça do tipo Greyhole pode ser considerado um caso particular do ataque Black Hole, onde o invasor captura as rotas, torna-se integrante delas e depois descarta pacotes seletivamente [Sen et al. 2007]. Os criterios para o descarte dos pacotes podem variar dependendo das intenções do atacante, como por exemplo pacotes vindos ou com destino a algum nó específico, escolhidos estatísticamente ou seguindo qualquer outro critério. A diferença entre eles é que a ameaça Blackhole absorve todos os pacotes recebidos e o tipo Greyhole absorve apenas uma parte deles. Logo, a ameaça Greyhole é mais difícil de ser detectada.

2.2 PROTOCOLOS DE ROTEAMENTO

Em redes de computadores os protocolos de roteamento são algorítmos que determinam o caminho mais eficiente e otimizado para transmissão de pacotes de dados entre dispositivos, como os roteadores, desempenhando um papel fundamental na comunicação entre redes.

Segundo o autor Douglas Comer[Comer 2016], protocolos de roteamento desempenham duas funções importantes, primeiro calculando o conjunto de caminhos mais curtos e segundo respondendo a falhas de rede ou mudança de topologia atualizando continuamente as informações de roteamento. Portanto, os protocolos de roteamento pesquisam os destinos, calculam o menor caminho para cada destino e passam a informação para o software do protocolo de roteamento, armazenando na tabela de encaminhamento do roteador. Essa tabela, por sua vez, é uma estrutura que contém as informações necessárias para o roteador ”decidir”qual o melhor trajeto para chegar ao endereço de destino do pacote.

Ainda, os protocolos de roteamento podem ser classificados em duas grandes categorias: estático e dinâmico. No roteamento estático, o roteador conhece as rotas e sub-redes exclusivamente por meio da configuração manual das informações realizadas pelo administrador. Ou seja, as rotas são predefinidas e não há troca automática de informações entre os roteadores. Nesse caso, sua principal vantagem é que essa técnica consome menos recursos de processamento no roteador, já que não é necessário executar algoritmos de roteamento dinâmico. Isso também implica em menos tráfego na rede, uma vez que não há a necessidade de enviar mensagens de atualização entre os dispositivos. No entanto, sua principal desvantagem está na falta

de flexibilidade. Em redes maiores ou em ambientes dinâmicos, a configuração manual pode tornar-se inviável, pois qualquer alteração na topologia da rede (como falha de links ou adição de novos dispositivos) exige que o administrador faça alterações manuais em cada roteador. Com o crescimento da rede, o roteamento estático se torna cada vez mais difícil de gerenciar, tornando-o mais adequado para redes pequenas ou muito controladas.

Já no roteamento dinâmico, as informações de rotas são automaticamente aprendidas por meio da troca de mensagens entre os roteadores, o que permite que eles compartilhem informações sobre a topologia da rede e adaptem as rotas conforme as mudanças ocorrem. Os protocolos de roteamento dinâmico podem ser classificados em internos (IGP) ou externos (EGP). O roteamento interno (ou IGP, do inglês Interior Gateway Protocol) é utilizado dentro de uma única rede ou organização, como em redes corporativas, para definir as rotas entre os roteadores internos de uma empresa. Já o roteamento externo (ou EGP, do inglês Exterior Gateway Protocol) é utilizado para o roteamento entre diferentes sistemas autônomos, como no caso da Internet, onde protocolos como o BGP (Border Gateway Protocol) são usados para definir como os dados são encaminhados entre diferentes redes independentes. A principal vantagem do roteamento dinâmico é sua capacidade de adaptação a mudanças na rede, como falhas de links ou a adição de novos roteadores, sem necessidade de intervenção manual. No entanto, ele consome mais recursos, já que exige processamento adicional para calcular e atualizar as rotas, além de gerar tráfego de controle para a troca de informações entre os roteadores.

Desta forma, os protocolos de roteamento estático são usados em redes pequenas ou controladas, como em escritórios de empresas ou conexões de filiais a uma sede, onde as rotas são fixas e não mudam com frequência. Já os protocolos de roteamento dinâmico são usados em provedores de internet (ISPs), redes corporativas grandes e data centers, onde a topologia da rede muda constantemente e as rotas precisam se ajustar automaticamente para garantir a melhor entrega dos dados.

Tendo conceituado os protocolos de roteamento, é importante ressaltar que há uma grande variedade de protocolos que atendem a diferentes necessidades de redes, cada um com suas características, vantagens e limitações. Os protocolos de roteamento estático, por exemplo, são usados em redes pequenas ou controladas, onde as rotas são configuradas manualmente e não há necessidade de adaptação dinâmica à topologia da rede. Já os protocolos de roteamento dinâmico, como OSPF e RIP, são mais indicados para redes de maior escala, que exigem uma adaptação automática a mudanças na topologia e podem ser classificados em dois tipos: internos (IGP) para redes dentro de uma organização, e externos (EGP) para interconectar redes independentes, como no caso da Internet. Além disso, existem protocolos específicos para redes ad hoc e Wireless Mesh Networks (WMNs), como o HWMP e o

B.A.T.M.A.N Advanced, que são voltados para cenários de alta mobilidade e topologias dinâmicas, onde os dispositivos podem entrar ou sair da rede a qualquer momento.

Nesse contexto, a escolha do protocolo Babel se justifica, principalmente quando se observa o trabalho do Programa de Pós-Graduação em Ciência da Computação da UFSC, intitulado ”Uma extensão do protocolo Babel para inclusão de mecanismo de incentivo com a proposta de custo operacional”, onde o estudo comparativo realizado por e Rodolfo Borges dos Santos Carvalho entre os três principais protocolos de roteamento em WMNs — HWMP, B.A.T.M.A.N Advanced e Babel — mostrou que o Babel se destaca por sua eficiência, robustez e menor custo operacional, apresentando um desempenho superior devido à sua capacidade de adaptação rápida a mudanças na topologia da rede, menor sobrecarga de controle e melhor utilização da largura de banda, características essenciais para o bom funcionamento de redes mesh com alta mobilidade e topologias dinâmicas. Assim, o protocolo Babel se mostrou a escolha mais adequada para cenários em que a rede está em constante mudança, com dispositivos móveis ou temporários, como em ambientes urbanos ou redes comunitárias.

2.2.1 BABEL

O protocolo Babel é um protocolo de roteamento dinâmico designado para ser eficiente tanto em redes de roteamento hierárquico, quanto também em redes sem fio, como são as WMNs. Segundo o autor CHROBOCZEK, o Babel foi formalizado tecnicamente no ano de 2011 e originalmente, o Babel foi baseado no algoritmo de roteamento Bellman-Ford(CHROBOCZEK, Juliusz, 2021b), um dos algoritmos mais conhecidos e utilizados para determinar a melhor rota entre dois nós em uma rede, garantindo um roteamento eficaz não apenas em redes cabeadas, mas principalmente em redes sem fio com topologias dinâmicas, que estão em constante mudança.

Redes que precisam de adaptação rápida a alterações em sua topologia, característica comum em redes sem fio, configuram o cenário para o qual, particularmente, o Babel é adaptado. Em ambientes como as WMNs, em que nós (roteadores) podem se mover e a conectividade entre eles pode mudar rapidamente, o protocolo Babel se destaca por sua capacidade de se ajustar automaticamente a essas variações.

O funcionamento do protocolo Babel pode ser entendido como uma dinâmica de troca de informações entre os nós da rede, para garantir que todos tenham conhecimento das melhores rotas para alcançar outros dispositivos. A cada período de tempo, os nós enviam informações sobre suas rotas para os seus vizinhos. Essas informações incluem a distância até outros nós, que pode ser medida em saltos ou tempo de transmissão, além de dados sobre a qualidade das rotas, como a largura de banda disponível e a latência.

O protocolo Babel utiliza uma abordagem de vetor de distância para calcular e ajustar rotas. Cada nó mantém uma tabela de roteamento com as melhores rotas

para alcançar outros dispositivos na rede. Quando a topologia muda — por exemplo, quando um nó se move ou se desconecta — o Babel recalcula as rotas e propaga essas alterações para os vizinhos. Isso garante que a rede continue funcionando de maneira eficiente, mesmo diante de mudanças constantes.

Uma das grandes inovações do Babel em relação a outros protocolos de roteamento, como o RIP (Routing Information Protocol) ou OSPF (Open Shortest Path First), é o foco na resolução de problemas específicos de redes sem fio. O protocolo implementa estratégias para evitar a formação de loops de roteamento, que ocorrem quando os pacotes ficam presos em um ciclo infinito entre os nós. Além disso, o Babel previne a ocorrência de black holes, situação em que um nó falha em retransmitir pacotes e os descarta sem encaminhá-los para o destino correto.

Esses refinamentos tornam o protocolo Babel altamente eficaz e robusto, mesmo em redes que apresentam alta mobilidade e mudanças frequentes na topologia. De acordo com Chroboczek (2021b), o protocolo é especialmente bem-sucedido em ambientes onde há a necessidade de convergência rápida e a minimização de erros durante a troca de rotas.

Quanto ao seu funcionamento o protocolo Babel é projetado para garantir a convergência eficiente em redes dinâmicas, esta convergência ocorre sempre que há uma modificação na rede, sendo detectada através da troca de pacotes TLVs (Type-Length-Value) entre os nós. O processo inicia com o envio regular de pacotes Hello, nos quais cada nó indica que está ativo na rede. Quando um nó envia um pacote Hello para um vizinho, o nó receptor usa o histórico recente dos pacotes Hello recebidos para calcular um valor denominado rxcost (reception cost), que representa o custo da rota do ponto de vista do nó receptor, ou seja, o custo do link do nó que enviou o pacote Hello até o nó que o recebeu.

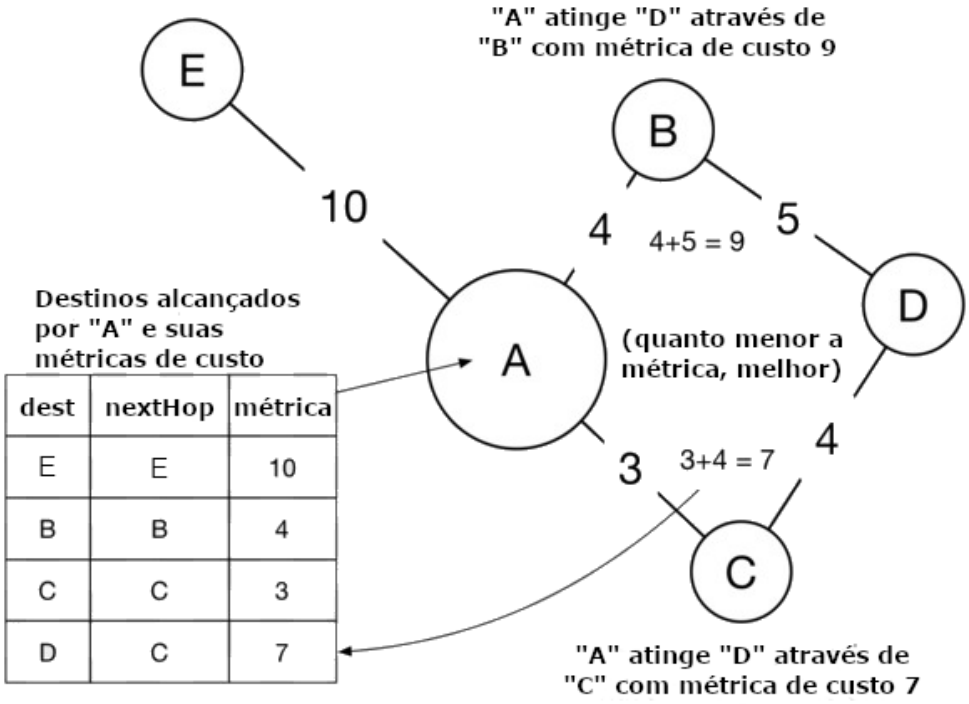
Como resposta a esses pacotes Hello, os nós enviam mensagens IHU (I Heard You), garantindo a visibilidade bidirecional entre os nós. A troca das mensagens IHU é crucial, pois permite que todos os nós da rede saibam sobre a conectividade e o estado das rotas entre si, mantendo a consistência das informações de roteamento. Dentro de cada pacote IHU, o nó remetente envia o valor de rxcost, que representa o custo da rota do ponto de vista do nó que enviou a mensagem, e o intervalo de tempo utilizado para o envio dos próprios pacotes IHU.

Ao receber a mensagem IHU de um vizinho, o nó receptor atualiza o custo da rota associada a esse vizinho, atribuindo o valor recebido da mensagem IHU à variável txcost (transmission cost), que indica o custo do link entre os dois nós do ponto de vista do nó que recebe a mensagem IHU. O cálculo final do custo do link entre dois nós, desde que sejam alcançáveis, combina esses dois fatores: o rxcost, que reflete o custo da rota a partir do histórico de pacotes Hello recebidos, e o txcost, que é o custo calculado com base nas mensagens IHU. Além desses, o protocolo também leva em

conta o RTT (Round-Trip Time), que mede o tempo de ida e volta de um pacote entre os nós, e inclui essa métrica no cálculo final do custo da rota.

Esse processo contínuo de troca de pacotes e atualização de informações de custo permite que o Babel se adapte rapidamente às mudanças na topologia da rede e calcule rotas eficientes, levando em consideração tanto a qualidade quanto a estabilidade dos links.

Figura 3 – Exemplo de cálculo da tabela de roteamento.



Fonte: [mesh.org 2013]

Assim, as vantagens do protocolo Babel em WMNs podem ser resumidas em sua eficiência no uso de recursos, sendo um protocolo leve que consome pouca largura de banda, o que é essencial em redes com dispositivos de recursos limitados, como roteadores pequenos e sensores. Além disso, o Babel oferece alta resiliência: em redes sem fio dinâmicas, onde falhas e movimentações de nós podem desestabilizar temporariamente a rede, o protocolo reage rapidamente, recalculando as rotas para manter a operação da rede de forma eficiente. Outra vantagem é sua escalabilidade, permitindo o funcionamento tanto em redes pequenas quanto em grandes, adaptando-se bem a diferentes ambientes e topologias. Por fim, o Babel é eficaz na prevenção de problemas de roteamento, como loops e black holes, comuns em redes sem fio, garantindo uma comunicação robusta e sem falhas em ambientes altamente dinâmicos.

Por outro lado, o Babel pode gerar overhead em redes grandes devido à troca constante de pacotes de controle, o que pode afetar a performance, especialmente em

ambientes com recursos limitados. Sua complexidade também é uma desvantagem, pois exige mais processamento e memória, o que pode ser um desafio em dispositivos com capacidade limitada. Em redes com alta mobilidade, o protocolo pode melhorar a convergência, resultando em menor atraso e perda de dados. Além disso, como outros protocolos de vetor de distância, o Babel pode ser vulnerável a ataques de injeção de informações falsas, embora existam mecanismos para mitigar esses riscos.

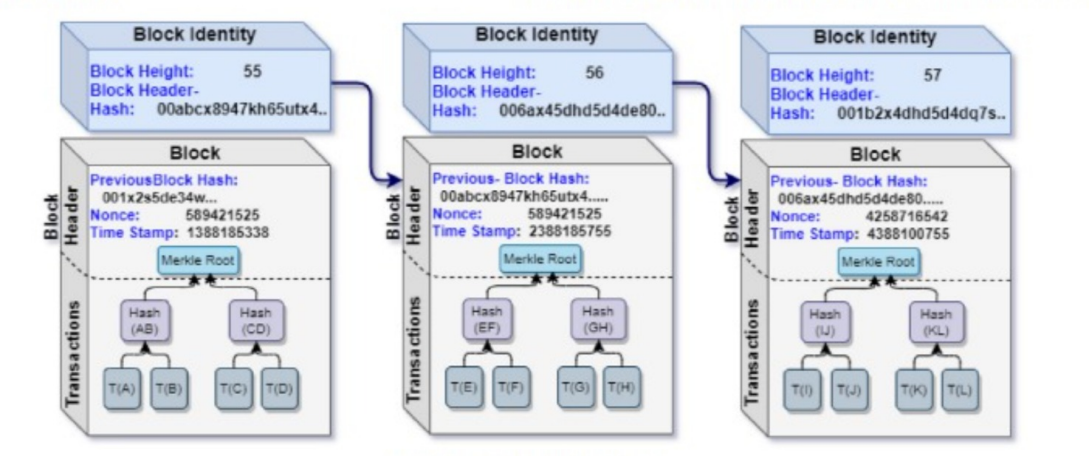
2.3 BLOCKCHAIN

Blockchain pode ser traduzida como corrente de blocos. De uma forma simples, trata-se de uma tecnologia que agrupa um conjunto de informações que se conectam por meio de criptografia. A utilização dessa tecnologia permite o compartilhamento seguro e descentralizado de informações sem a dependência de terceiros (NAKAMOTO, SATOSHI, 2008).

Aitzhan e Svetinovic (2006) explicam a blockchain como uma cadeia cronologicamente ordenada de blocos protegidoss por vários tipos de consenso, como o Proof-of-Work. O encadeamento é feito adicionando o hash do bloco anterior ao bloco atual. O alinhamento dos blocos de forma cronológica faz com que uma transação não possa ser alterada com antecedência sem alterar seu bloco e todos os blocos a seguir.

Desta forma a estrutura do blockchain oferece uma camada robusta de segurança através de sua natureza descentralizada e imutável. Cada bloco contém um registro de transações que é validado por uma rede de nós independentes, tornando extremamente difícil para qualquer entidade alterar ou manipular dados sem ser detectada. Como Satoshi Nakamoto afirmou em seu artigo seminal sobre Bitcoin, 'o blockchain é seguro pela honestidade coletiva da rede, que converge em um único histórico de transações provado por consenso'. Essa arquitetura única não só protege contra fraudes, mas também assegura a integridade e transparência das informações registradas.

Figura 4 – Estrutura de blocos da Blockchain



Fonte: Journal of King Saud University - Computer and information sciences 34 (2022)

2.3.1 Públicas ou Privadas

A Blockchain pública é uma infraestrutura descentralizada na qual qualquer pessoa pode participar sem necessidade de permissão. Nesse tipo de blockchain, todos os dados são acessíveis publicamente e qualquer usuário pode enviar transações, validar blocos e participar do processo de consenso. A segurança é mantida através de algoritmos criptográficos robustos e de um sistema de consenso distribuído, como o Proof of Work (PoW) ou Proof of Stake (PoS). A transparência e a imutabilidade dos registros são garantidas pela replicação dos dados em todos os nós da rede, o que dificulta a manipulação e garante a confiança nas transações.

Logo, podem ser auditados por qualquer pessoa, e cada nó tem tanto poder de transmissão quanto qualquer outro. Antes de uma transação ser considerada válida, ela deve ser autorizada por cada um de seus nós constituintes por meio do processo de consenso da cadeia. Desde que cada nó cumpra as estipulações específicas do protocolo, suas transações podem ser validadas e, assim, adicionadas à cadeia.

Em contraste, a blockchain privada é uma rede controlada e operada por um consórcio ou organização específica. Ela define regras de acesso, participação e governança que limitam quem pode ler a blockchain, enviar transações e validar blocos. Diferente das blockchains públicas, onde qualquer pessoa pode participar, as blockchains privadas possuem confidencialidade e o controle sobre os dados são prioridades. Os participantes geralmente são identificados e autorizados, o que permite uma governança mais centralizada e a capacidade de ajustar regras e políticas conforme necessário.

2.3.2 Permissivas ou Não permissivas

Blockchains permissivas e abertas são sistemas descentralizados que oferecem diferentes graus de acesso e participação, adaptando-se a diversas necessidades e

contextos. As blockchains permissivas são caracterizadas por regras definidas sobre quem pode participar na validação de transações e na operação da rede.

Esses sistemas frequentemente requerem permissão para acesso completo aos recursos da blockchain, garantindo maior controle e segurança em ambientes como empresas e consórcios. Embora menos abertas em comparação com as blockchains públicas, as permissivas proporcionam eficiência e flexibilidade para aplicações onde a governança centralizada é valorizada.

Já a Blockchains não permissivas, por outro lado, são projetadas com princípios de total descentralização e acesso irrestrito. Esses sistemas permitem que qualquer pessoa participe livremente, enviando transações, validando blocos e contribuindo para o consenso da rede sem a necessidade de permissão prévia. A transparência é um dos pilares fundamentais das blockchains abertas, pois todos os dados são publicamente visíveis e auditáveis por qualquer usuário. Isso promove a confiança e a integridade nas transações, eliminando a necessidade de intermediários tradicionais e reduzindo custos operacionais em diversas aplicações, desde financeiras até sociais.

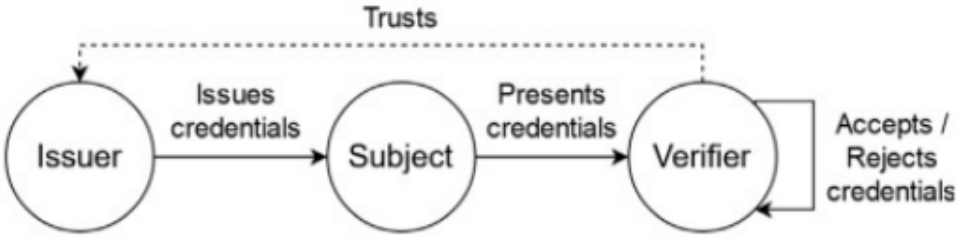
2.3.3 Identidade auto-soberana

A identidade auto-soberana é um conceito inovador que coloca o controle dos dados de identidade digital diretamente nas mãos dos usuários, eliminando a dependência de intermediários ou provedores centralizados. Esse modelo pressupõe que o indivíduo é o único responsável por gerenciar e compartilhar suas informações. Diferente dos sistemas tradicionais, onde as identidades digitais são armazenadas e gerenciadas por terceiros, a identidade auto-soberana permite que os dados sejam armazenados de forma descentralizada, muitas vezes localmente com o próprio usuário, que os apresenta conforme necessário.

Essa ideia ganhou força ao longo dos anos com base nos princípios propostos por diversos estudiosos. Foram estabelecidos 10 princípios que moldam a base para as identidades auto-soberanas, incluindo a existência, que defende que o usuário deve ter uma identidade independente de provedores, e o controle, que reforça que os usuários devem ter total autonomia sobre suas informações. Além disso, princípios como acesso, transparência, persistência, e portabilidade destacam a necessidade de as identidades serem acessíveis, duráveis e utilizáveis em diversos sistemas.

Para viabilizar essa gestão descentralizada e autônoma, a tecnologia blockchain se apresenta como uma solução ideal. A blockchain, por sua estrutura descentralizada e imutável, possibilita que as informações de identidade sejam armazenadas de maneira segura, garantindo a integridade e a autenticidade dos dados. Quando um usuário apresenta sua identidade ou credencial, a parte verificadora pode consultar a blockchain para validar essas informações, sem a necessidade de uma autoridade

Figura 5 – Demonstração do funcionamento da estrutura de identidade auto-soberada



Fonte: Journal of Network and Computer Applications 212 (2023)

central para confirmar a autenticidade. Isso elimina pontos de falha únicos e reduz a possibilidade de manipulação ou fraude.

Embora o uso da blockchain não seja obrigatório para implementar identidades auto-soberanas, ela oferece vantagens significativas. A natureza imutável da blockchain garante que essas informações permaneçam confiáveis e seguras ao longo do tempo, sem a necessidade de confiar em intermediários ou autoridades centralizadoras. Portanto, a combinação de identidade auto-soberana e blockchain representa um avanço importante na segurança digital, conferindo ao usuário o poder sobre sua própria identidade e garantindo maior privacidade e proteção de seus dados.

3 ESTADO DA ARTE

A compreensão do estado da arte no uso da blockchain em WMNs (Wireless Mesh Networks) é de extrema importância para o desenvolvimento deste Trabalho de Conclusão de Curso, pois proporciona a base teórica necessária para a análise crítica e aplicação adequada dessa tecnologia em redes sem fio. Dado o potencial da blockchain para aprimorar a segurança, a descentralização e a transparência em redes dinâmicas, como as WMNs, entender as abordagens mais recentes e as soluções propostas na literatura é fundamental para a inovação e o sucesso do projeto. Para alcançar esse entendimento, foi realizada uma revisão detalhada da literatura, abordando os principais estudos e avanços sobre a integração da blockchain em redes sem fio, com foco em aspectos como algoritmos de consenso, escalabilidade, consumo de recursos e desafios específicos de segurança. Essa análise permitiu identificar as principais vantagens e limitações do uso da blockchain em WMNs, além de fornecer uma visão crítica sobre as soluções já implementadas e os caminhos futuros para a pesquisa. Portanto, essa compreensão do estado da arte não só fundamenta as escolhas metodológicas deste trabalho, mas também orienta a proposição de soluções e inovações para superar os desafios encontrados.

3.1 BASE DE DADOS E STRING DE BUSCA

A fim de identificar os estudos que mais se relacionam ao assunto abordado por este trabalho, foi montada uma string de busca contendo termos e conceitos dispostos em uma única linha, todos unidos através de operadores booleanos (AND, OR), resultando na seguinte string:

$$"blockchain \quad AND \quad (mesh \quad OR \quad multihop \quad OR \quad MANET) \quad AND \quad security \quad AND \quad trust \quad AND \quad wireless"$$

O critério utilizado para a seleção dos estudos foi baseado na pesquisa de todos os trabalhos existentes de 2022 até o ano de 2023, retratando o tema nesses dois anos de publicações da literatura.

A pesquisa foi realizada em quatro bases de dados (IEEE, ACM, Wiley, Science Direct), por serem utilizadas para produção científica na área de Ciência da Computação, terem uma boa cobertura de trabalhos dentro do tema discutido, estarem regularmente atualizadas com novas publicações, apresentarem suporte para exportação dos resultados alcançados, trabalharem com mecanismos de busca intuitivos e que disponibilizam filtros que atendem os critérios de busca definidos. Durante a análise preliminar dos estudos encontrados, foram identificadas as seguintes quantidades:

Base	String	Filtros	Resultados
ACM	[All: blockchain] AND [[All: mesh] OR [All: multihop] OR [All: manet]] AND [All: security] AND [All: trust] AND [All: wireless]	Publication Date: 2022-2023.	152
IEEE	blockchain AND (mesh OR multihop OR MANET) AND security AND trust AND wireless	Range: 2022-2023.	3
Science Direct	blockchain AND (mesh OR multihop OR MANET) AND security AND trust AND wireless	Years: 2022-2023.	218
Wiley	blockchain AND (mesh OR multihop OR MANET) AND security AND trust AND wireless	Publication Date: 2022-2023.	205
TOTAL			578

Tabela 1 – Busca realizada em cada base de dados

3.2 CRITÉRIOS DE INCLUSÃO E EXCLUSÃO

Critérios de Inclusão

IC1 Trabalhos que abordem a utilização de blockchain dentro de WMNs.
IC2 Trabalhos que adotem mecanismos de segurança para roteamento de pacotes dentro da rede.
IC3 Trabalhos publicados desde o ano de 2022 até a 2023 data de realização deste trabalho.
IC4 Trabalhos escritos em língua Inglesa e Portuguesa.

Critérios de Exclusão

EC1 Serão excluídos trabalhos que não abordem o uso da tecnologia blockchain no seu contexto.
EC2 Serão excluídos trabalhos que não abordem propostas de segurança em redes WMNs.
EC3 Serão excluídos trabalhos publicados como artigos curtos ou pôsteres.
EC4 Serão excluídos trabalhos que apresentam avaliações sem apresentar o método utilizado.
EC5 Serão excluídos trabalhos que o acesso estiver indisponível.

3.3 PROCEDIMENTOS DE SELEÇÃO

O processo de seleção ocorrerá conforme descrito pelas seguintes fases:

SP1 Serão construídas strings de busca com as palavras-chave e seus sinônimos que estejam relacionados ao objetivo deste processo de revisão sistemática.
SP2 As strings serão submetidas aos mecanismos de busca das bases escolhidas.
SP3 Os resultados obtidos passarão por uma retirada de duplicatas presentes.
SP4 Após leitura do título, resumo e palavras-chave, serão aplicados critérios de inclusão e exclusão, extraindo os trabalhos que tenham relevância para o estudo.
SP5 Os artigos então serão lidos na íntegra, mantendo a aplicação dos critérios de inclusão e exclusão para filtrar os trabalhos de maior relevância.
SP6 O revisor fará a extração dos dados de cada trabalho para demonstrar os resultados obtidos.

3.4 ANÁLISE DE DADOS - EXECUÇÃO

3.4.1 Seleção dos Estudos

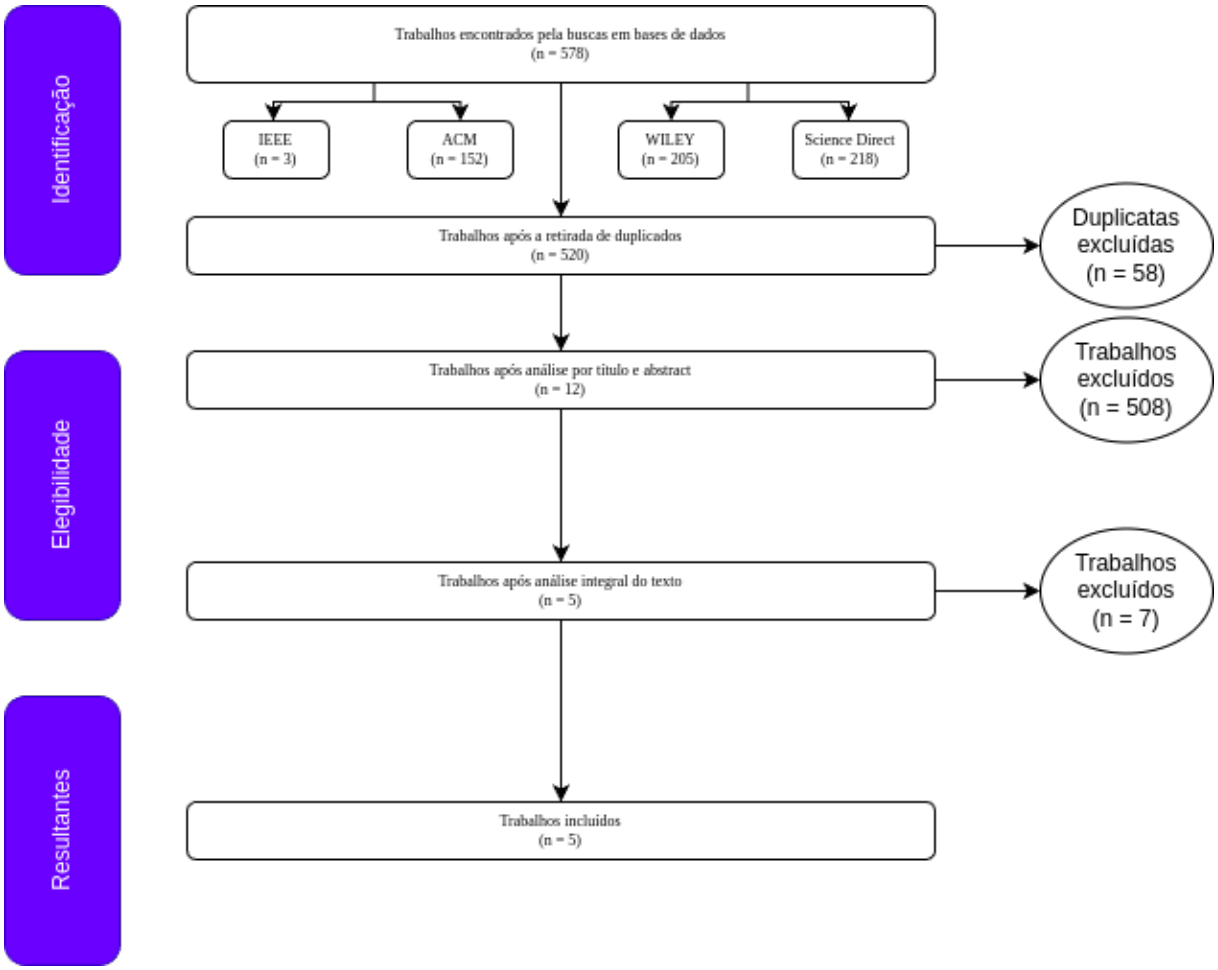
O processo de execução foi realizado conforme ilustrado na Figura 6. A realização do processo foi por meio de uma planilha disponibilizada neste link e está represetando no diagrama acima.

3.5 TRABALHOS CORRELATOS

O trabalho de [Bala et al. 2023] aborda a necessidade de autenticação segura para proteger a privacidade dos usuários VANETs - Veicular Ad Hoc Networks, buscando resolver problemas de gerenciamento de confiança a fim de não permitir que veículos hostis transmitam informações falsas, focando na segurança das comunicações. Dentro das possíveis alternativas que atendam a tais necessidades, o trabalho foca em uma arquitetura de gerenciamento de confiança baseada em blockchain privado e contratos inteligentes com um sistema de autenticação que protege a privacidade dos veículos e permite que a autoridade de confiança reconheça e proíba veículos maliciosos. Os resultados alcançados mostram que a solução proposta é eficiente e prática, o sistema de gerenciamento de confiança baseado em blockchain foi testado e demonstrou ser eficaz na proteção da privacidade e na garantia da confiabilidade dos dados.

Já os desafios relacionados aos ataques cibernéticos em dispositivos IoT, que representam sérias ameaças à integridade, confidencialidade e disponibilidade desses dispositivos e seus dados, segundo [Pourrahmani et al. 2023] é necesário medidas de mitigação para as vulnerabilidades existentes nas diferentes camadas do

Figura 6 – Ilustração do processo de execução



Fonte: autora

modelo de referência IoT. Além deste trabalho analisado fornecer uma fonte abrangente para identificar e descrever os principais ataques cibernéticos em dispositivos IoT, ele propõe a utilização da tecnologia Blockchain para melhorar a segurança, transparência e confiança no ecossistema IoT. Para os autores a criptografia do tráfego de rede e a utilização de protocolos de mensagens seguros foram as formas para solução das necessidades, sugerindo, também, um conjunto de recomendações de uma estratégia abrangente para mitigar os diversos tipos de ataques cibernéticos.

Ainda dentro dos desafios relacionados à segurança, o trabalho de [V et al. 2023] cita vulnerabilidades como o problema do ”cold start”, onde novos nós não possuem reputação estabelecida, e o ataque ”sybil”, onde um atacante utiliza múltiplas identidades fictícias para manipular o sistema. Contribuindo com uma proposta de um algoritmo de roteamento baseado em reputação e uma abordagem baseada em blockchain para melhorar a segurança e a confiabilidade da transmissão de dados em MANETs - Mobile Ad Hoc Networks. Como forma de solucionar o problema, a proposta de um framework integrado de agregação e segmentação de dados baseado em blockchain

descentraliza a coleta e o gerenciamento de dados, aumentando a segurança e a confiabilidade da transmissão de dados. Através de simulações, foi demonstrado que o framework melhora a confiabilidade e a segurança, no entanto, o trabalho também reconhece a necessidade de mais pesquisas para avaliar o desempenho do framework em diferentes condições de rede e explorar a escalabilidade em redes maiores.

Outro estudo acerca das adversidades da integridade, privacidade dos dados, autenticação e autorização robusta de usuários é citado por [Kumar e Sharma 2022], o trabalho discute sobre desafios e problemas dominantes na segurança e confiança do IoT, procurando soluções através do uso da tecnologia Blockchain. Assim, o estudo fornece uma pesquisa detalhada sobre os trabalhos existentes de gerenciamento de confiança que utilizam técnicas convencionais e baseadas em Blockchain, destacando as vantagens significativas da tecnologia Blockchain em garantir a confiança no IoT em comparação com métodos de segurança convencionais. Para os autores, a Blockchain pode ser utilizada como um serviço para alocação de recursos entre dispositivos e para a troca segura de informações e em busca de mitigar as limitações da Blockchain, como a taxa de transação lenta e a falta de adaptabilidade, o estudo recomenda o uso de novas tecnologias emergentes como Holochain e Hashgraph.

Por fim, o estudo de [Mrabet, Bouanani e Ben-Azza 2023] busca resolver problemas em sistemas de reputação dinâmicos e descentralizados, focando principalmente na preservação da privacidade. Os autores abordam desafios como a acessibilidade, a perda de informações de reputação, segurança contra adversários maliciosos e a necessidade de um sistema que permaneça funcional mesmo quando participantes saem da rede. Como solução, é proposta a implementação de blockchain para armazenamento e atualização de informações de reputação, garantindo que essas informações não sejam perdidas e estejam sempre acessíveis, carcacterizando um novo sistema de reputação dinâmico, descentralizado e que preserva a privacidade. O sistema proposto não é afetado pela saída de participantes da rede e mantém as informações de reputação seguras e acessíveis. O estudo também menciona a necessidade de trabalhos futuros para focar na escalabilidade blockchain através de técnicas como Sharding e Nested Blockchain.

3.5.1 Comparativos dos estudos correlatos e proposta

Para melhor visualização foi criada uma tabela comparativa acerca dos trabalhos correlatos e a proposta do presente estudo.

Tabela 2 – Comparativo de Trabalhos Correlatos.

Trabalho	Compatível com Blockchain permissiva	Protocolo Babel	Uso de incentivos	Incentivo econômico	Autenticação segura	Grupo de Comparação
[V et al. 2023]	O		O		O	O
[Pourrahmani et al. 2023]	O		O			
[Kumar e Sharma 2022]			O			
[Bala et al. 2023]			O		O	
[Mrabet, Bouanani e Ben-Azza 2023]			O			
PROPOSTA	**O**	**O**	**O**	**O**	**O**	**O**

Fonte: Autora

4 CORPO DO PROJETO

4.1 ANÁLISE E MODIFICAÇÃO DO PROTOCOLO BABEL

Neste capítulo será apresentada a proposta de solução deste trabalho, no entanto, cabe ressaltar que no decorrer da pesquisa foi observado que para a inclusão de uma solução de segurança para mecanismos de incentivos associados a blockchains para WMNs, seria necessário inicialmente identificar o desempenho do protocolo babel, realizando um comparativo entre versão padrão do Babel e versão Babel Cost, que associa mecanismos de adição de custo operacional para cada nó. Neste sentido, as alterações para o protocolo Babel Cost versão 1.13 podem ser divididas em duas etapas, a primeira com mudanças feitas no roteamento e a segunda com mudanças feitas na troca de mensagens do Babel.

Ainda, neste capítulo é apresentada a metodologia, as alterações realizadas no protocolo Babel Cost, o ambiente de teste e a demonstração dos resultados obtidos entre comparativos na execução do protocolo Babel versão 1.13 e do protocolo Babel Cost versão 1.13-alterado. Ao final, é exposta uma análise dos testes realizados, pontuando as vantagens e desvatagens de cada uma das versões, bem como os resultados experimentais obtidos acerca do desempenho.

4.2 METODOLOGIA

O trabalho realizado é de natureza aplicada e baseado no método Hipotético-Dedutivo. A estratégia adotada para a aplicação da pesquisa é baseada no esquema proposto por [Marconi e Lakatos 2011], que mostra o método hipotético-dedutivo dividido em 4 etapas.

A etapa de conhecimento é baseada no levantamento de estudos relacionados com os objetivos do trabalho, através de uma revisão sistemática da literatura, que inclui artigos científicos, revistas e eventos de bases de dados relacionadas ao tema de computação.

A etapa de problema é baseada na identificação de desafios encontrados no levantamento de artigos e outros trabalhos da literatura, que estão relacionados com a segurança de WMNs incentivadas e blockchains.

A etapa de conjectura é realizada com o planejamento e inclusão de teste das alterações no funcionamento do protocolo Babel para WMNs, buscando realizar um comparativo entre versões e análise da seleção de rotas e ETX (Expected Transmission Cost).

A etapa de falseamento é representada pelos testes e avaliações realizadas sobre as implementações desenvolvidas, considerando tanto as análises qualitativas quanto

as quantitativas.

4.3 ALTERAÇÕES DE VERSÃO DO PROTOCOLO BABEL

O protocolo Babel é uma solução de roteamento dinâmico projetada para redes de malha (mesh networks), oferecendo uma maneira eficiente e flexível de gerenciar a distribuição de pacotes em ambientes complexos. Ao longo do tempo, o protocolo tem evoluído para melhorar sua performance, segurança e compatibilidade com diferentes ambientes de rede. As atualizações regulares buscam corrigir falhas, introduzir novos recursos e otimizar o comportamento do sistema, garantindo que ele continue a atender às crescentes demandas de redes dinâmicas. A seguir, são apresentadas as principais alterações introduzidas nas versões mais recentes e utilizadas neste trabalho, a versão 1.13.1 do protocolo Babel.

A versão 1.13, lançada em 16 de julho de 2023, trouxe várias mudanças importantes para o protocolo Babel. Houve uma otimização significativa na redistribuição de rotas. Em vez de realizar um despejo completo de rotas a cada mudança, a versão 1.13 agora instala apenas a rota que o kernel enviou, melhorando a eficiência do processo. Outra novidade foi a adição da opção ”shutdown-delay-ms”, que permite a configuração de um atraso opcional no processo de desligamento do daemon, proporcionando mais controle ao administrador da rede. A versão também corrigiu problemas de compilação em sistemas BSD, garantindo que o código fosse compilado corretamente nesses ambientes.

Já a versão 1.13.1, lançada em 26 de julho de 2023, trouxe duas mudanças notáveis. A primeira foi a implementação da opção ”probe-mtu”, que pode ser configurada por interface. Essa funcionalidade tem como objetivo descartar automaticamente links com MTU mal configurado, o que ajuda a melhorar a detecção e correção de problemas relacionados ao tamanho máximo de pacotes em redes. A segunda mudança foi a correção de um erro de digitação na manipulação da opção ”v4viav6”, o que assegurou o funcionamento adequado dessa configuração. Essas melhorias visaram tornar o protocolo mais robusto e confiável, com foco na estabilidade da rede e na precisão das configurações.

4.4 ALTERAÇÕES PARA INCLUSÃO DE CUSTO NO BABEL

A alteração realizada no protocolo Babel da versão 1.13 inclui métricas de custo econômico para o cálculo de seleção de rotas ETX (Expected Transmission Cost), utilizando um valor de custo a ser atribuído para cada nó da rede, denominado tocost (transmission operational cost), que será repassado aos vizinhos por meio de um pacote TLV (Type-Length-Value), no momento em que ocorrerem as troca de mensagens IHU (”I Heard You”).

Dentro de cada pacote TLV usado, do tipo IHU, será adicionada uma nova Sub-TLV denominada OC (Operational Cost), que será responsável por transportar o valor tocost de seu nó de origem para todos os vizinhos, a fim de que o custo dos links esteja coerente entre todos os nós da rede.

O valor tocost atribuído a cada participante, como representa o custo de operação que cada nó tem para o encaminhamento dos pacotes, será determinado pelo próprio usuário ou responsável do host pertencente à rede, podendo ser representados por alguma moeda ou token a serem livremente escolhidos conforme a necessidade dos participantes. Dependendo do local onde o nó se encontra e dos custos envolvidos, considerando hardware, manutenção, energia, e outros gastos, o valor de operação do nó em uma rede real sempre irá variar, por isso o custo é estipulado através de um número do tipo *unsigned short int*, que é estipulado durante a inicialização do nó.

4.4.1 Métricas de roteamento e alterações

No contexto Babel, os custos de rotas são utilizados para calcular métricas que definem os melhores caminhos em uma rede. Esses custos representam valores abstratos associados à qualidade de comunicação em um enlace específico. O objetivo do algoritmo é calcular, para cada nó origem da rede, a sua árvore de menor custo para chegar aos outros participantes da rede, atualizando assim a sua tabela de rotas.

Como elemento essencial, a tabela de rotas armazena informações sobre os caminhos disponíveis na rede, suas métricas e os vizinhos que os anunciam. Cada entrada da tabela inclui o destino da rota, o próximo salto, a métrica associada ao custo total do caminho e informações auxiliares, como o tempo de expiração da rota.

Esse tempo de expiração da rota refere-se ao intervalo máximo em que uma rota pode permanecer válida sem receber atualizações, ele é controlado por temporizadores que monitoram a atividade na rede, garantindo que as rotas obsoletas sejam removidas automaticamente, evitando inconsistências e problemas de desempenho. Em redes sem fio, onde a dinâmica de conectividade é alta, a tabela de rotas é atualizada frequentemente para refletir as mudanças rápidas nos enlaces.

O tempo padrão para atualizações no Babel varia conforme a configuração, mas em redes sem fio, pacotes Hello são enviados a cada 4 segundos, enquanto as rotas expiram em aproximadamente 20 segundos, se nenhuma atualização for recebida. Essa periodicidade garante a rápida detecção de mudanças na topologia da rede, mantendo a tabela precisa e eficiente.

Já os custos associados a cada rota na tabela de rotas podem ser ajustados para atender às necessidades específicas de um ambiente, permitindo flexibilidade na adaptação do protocolo. No entanto, esses ajustes devem respeitar a estrutura básica de cálculo, que tem como base a interação com os vizinhos para mensurar o custo de cada caminho.

Desta forma, por se tratar de redes sem fio, o cálculo da métrica de custo no protocolo Babel é baseado no conceito ETX (Estimated Transmission Cost) que reflete o número esperado de retransmissões, um cálculo que combina informações sobre a probabilidade de sucesso na transmissão e recepção de mensagens Hello.

O custo de recepção é representado por *rxcost*, cada nó mantém um histórico das mensagens *Hello* recebidas de seus vizinhos, como visto anteriormente. Esse histórico é utilizado para estimar a probabilidade *beta*, que retrata a probabilidade de sucesso no envio, portanto, o rxcost é calculado por um nó, mas o valor é compartilhado com o outro nó, enviando esse valor como parte do pacote IHU (I Heard You), o nó vizinho armazena essa métrica para utilizá-la em suas decisões de roteamento. O valor de *beta*, por sua vez, é calculado com base na proporção de mensagens Hello recebidas corretamente em relação ao total esperado. Assim, o rxcost, ou custo de recepção, é definido como:

$$rxcost = \frac{256}{\beta}$$

Já o custo de transmissão ou *txcost*, é calculado pelo próprio nó e reflete a qualidade da transmissão dos pacotes *TLV Hello* informado pelo vizinho, onde *alfa* retrata a probabilidade de sucesso na transmissão, assegurando que *alfa* nunca exceda o valor 1, mesmo que o *txcost* seja muito pequeno.

$$\alpha = \min\left(1, \frac{256}{txcost}\right)$$

A métrica de roteamento adotada no trabalho utiliza a base do algoritmo ETX acrescido da variável de custo denominada *tocost*, levando cada nó da rede a calcular o custo de um link baseado em quatro fatores:

1. o custo *rxcost* informado pelo vizinho;
2. o custo de transmissão *txcost* de recebimento de pacotes calculado pelo próprio nó;
3. o custo de atraso RTT (*round-trip time*) estimado para o link, inicialmente não presente no algoritmo ETX, mas adicionado pelo próprio time responsável pelo Babel para identificar as redes de melhor qualidade [Jonglez e Chroboczek 2019];
4. o custo operacional de transmissão *tocost*, calculado no nó de origem, representando o mecanismo de incentivo proposto pelo trabalho.

O cáclulo, então, para o custo de comunicação de um vizinho nesse protocolo de roteamento é feito no arquivo *neighbour.c*, no trecho de código da Figura(7) onde, a partir da linha 347 é realizada a implementação da função *neighbour cost*:

Figura 7 – Função de cálculo - neighbour cost

```
345 neighbour_cost(struct neighbour *neigh)
346 {
347     unsigned a, b, cost;
348 
349     if(!if_up(neigh->ifp))
350         return INFINITY;
351 
352     a = neighbour_txcost(neigh);
353 
354     if(a >= INFINITY)
355         return INFINITY;
356 
357     b = neighbour_rxcost(neigh);
358     if(b >= INFINITY)
359         return INFINITY;
360 
361     if(!(neigh->ifp->flags & IF_LQ) || (a < 256 && b < 256)) {
362         cost = a;
363     } else {
364         /* a = 256/alpha, b = 256/beta, where alpha and beta are the expected
365            probabilities of a packet getting through in the direct and reverse
366            directions. */
367         a = MAX(a, 256);
368         b = MAX(b, 256);
369         /* 1/(alpha * beta), which is just plain ETX. */
370         /* Since a and b are capped to 16 bits, overflow is impossible. */
371         cost = (a * b + 128) >> 8;
372     }
373 
374     cost += neighbour_rttcost(neigh);
375 
376     cost += neighbour_tocost(neigh);
377 
378     return MIN(cost, INFINITY);
379 }
```

Fonte: Autora

A função primeiro verifica se a interface está ativa usando if-up; caso contrário, retorna INFINITY. Se a interface estiver ativa, obtém os custos de transmissão (a) e recepção (b) com neighbour-txcost e neighbour-rxcost. Se algum desses valores for maior ou igual a INFINITY, o vizinho é considerado inviável, e a função retorna INFINITY.

Se os custos forem válidos, calcula o custo total. Caso a interface não suporte qualidade de enlace (ausência da flag IF-LQ) ou se a e b forem menores que 256, o custo é igual a a. Para outros casos, utiliza a fórmula baseada em ETX.

Depois, adiciona ao custo total os valores retornados por neighbour-rttcost (custo de tempo de ida e volta) e o campo configurável tocost da estrutura do vizinho. Por fim, limita o custo ao máximo permitido (INFINITY) com a função MIN. O resultado é uma

métrica que reflete a qualidade do enlace considerando probabilidades de transmissão e recepção bem-sucedidas.

Assim, a métrica utilizada é calculada somando os valores de cada variável de custo, o que possibilita a combinação de diferentes estratégias, sem comprometer as regras fundamentais do protocolo Babel. Isso ajuda a evitar a formação de loops nas rotas e garante a integridade da rede. De acordo com a RFC do protocolo Babel, existem três condições fundamentais para funcionamento do Babel.

A primeira dessas condições estabelece que se o custo local de um enlace for infinito, o custo total do enlace também será infinita, o que indica que esse enlace não pode ser utilizado para o roteamento. Isso impede que a rede tente usar um link que, por alguma razão, não esteja operacional ou não seja viável para o envio de pacotes.

A segunda condição exige que a métrica seja estritamente monotônica, o que significa que o custo do link calculada pelo nó deve ser maior do que a métrica que o nó vizinho reporta para o mesmo enlace. Essa propriedade é crucial para evitar a formação de loops, pois ela garante que os valores das métricas não voltem atrás de forma que um caminho já visitado seja revisitado repetidamente, criando um ciclo infinito de pacotes.

A terceira condição é a propriedade distributiva à esquerda, que especifica que se a métrica de um link vizinho for menor ou igual à métrica de outro vizinho , então a métrica calculada para o caminho com o primeiro vizinho deve ser menor ou igual à métrica calculada para o caminho com o segundo vizinho. Isso assegura que o algoritmo de roteamento do Babel sempre selecione o melhor caminho disponível, ou, pelo menos, o mais eficiente entre os caminhos possíveis, mesmo quando o caminho ótimo não estiver acessível. definido como:

- Se *c* é infinito, então *M(c,m)* também é infinito.
- *M* é estritamente monotônico: M(c,m) > m.
- *M* Atende a propriedade distributiva à esquerda: se *m* <= *m'* então *M(c, m)* <= *M(c, m')*.

As modificações realizadas através das métricas aditivas, para inclusão da variável *tocost*, além de outras alterações feitas no código, foram todas implementadas utilizando a linguagem C pura sobre a versão 1.13 do *babeld*.

Conforme evidenciado anteriormente nesse trabalho, as variáveis utilizadas no cálculo de custo pela métrica ETX sempre buscam considerar a estimativa de qualidade do link, em virtude da variação contínua da qualidade de cada link em conexões sem fio.

A inclusão do novo parâmetro *tocost* pode ser feita tanto pelo devido responsável pela rede, quanto pelo próprio usuário final da rede, desde que esse tenha acesso e

seja responsável pela área administrativa e de configuração do(s) seu(s) respectivo(s) dispositivo(s).

O layout para inclusão e/ou mudança do custo operacional, considerando um ambiente real, pode estar vinculado a uma tela de configuração de rede do nó, que é acessível pelo respectivo administrador, mas para efeito de testes, a inclusão do parâmetro *tocost* foi realizada através da modificação do arquivo *babeld.c*, onde foi incluída a nova opção de parâmetro definido pela letra Y (Figura 8 - linha 280). O novo parâmetro é considerado como opcional, para permitir a retrocompatibilidade do protocolo com redes antigas, além de permitir que cada componente da rede tenha a liberdade de escolha, possibilitando incluir ou não o custo operacional na somatória de sua métrica.

Figura 8 – Trecho de código arquivo babeld.c

```
280         case 'Y':
281             default_router_operacional_cost = parse_nat(optarg);
282             if(default_router_operacional_cost <= 0)
283                 goto usage;
284             break;
```

Fonte: Autora

A variável de custo *tocost*, depois de identificada por meio do parâmetro ”Y”, é então armazenada como uma variável global (Figura 8 - linha 281). A fim de que o parâmetro inserido seja considerado válido, as regras básicas requeridas pelo protocolo Babel, segundo as normas definidas pela própria RFC nos cálculos da métrica de custo [Chroboczek 2021], estipulam a necessidade do valor informado ser um inteiro curto (de 2 bytes) e não negativo (Figura 8 - linhas 281 e 282). O valor inteiro e não negativo garante que o custo não tenha valores fracionados, e não influencie no desbalanceamento de outras variáveis incluídas, que poderiam gerar inclusive métricas negativas, situação que apesar de permitida pelo algoritmo de Bellman-Ford, dentro das WMNs poderia gerar Loops de roteamento, portanto levando a métrica ETX a não considerar valores de custo negativo.

Devido à estrutura modular do protocolo Babel, é possível utilizar diferentes técnicas ou algoritmos para a política de seleção de rotas. Independente da técnica adotada, em todas elas existe a possibilidade de se utilizar a métrica de custo operacional OC, desde que obedecidos os princípios básicos especificados para funcionamento do protocolo. Assim, evitando a criação de *loops* nos caminhos de roteamento e problemas de *starvation* associados à alteração das condições de sustentabilidade da rede.

4.4.2 Troca de mensagens e alterações

O protocolo Babel organiza e executa as funções de roteamento na camada de rede por meio de troca de mensagens entre os dispositivos conectados. Cada dispositivo na rede é identificado de forma única dentro de um mesmo domínio por um identificador de roteador (Id-roteador), que é uma sequência de 8 bytes. Normalmente, esse identificador é gerado utilizando o método EUI-64 (Extended Unique Identifier - 64 bits), que se baseia no endereçamento IPv6.

As mensagens do protocolo Babel são enviadas em pacotes utilizando datagramas UDP. Esses pacotes possuem um cabeçalho de 4 octetos e são seguidos por uma ou mais TLVs (Type-Length-Value). Essas mensagens podem ser transmitidas entre endereços unicast ou multicast, com exceção das mensagens usadas para identificação e início de comunicação, como as mensagens Hello.

Para transmitir uma nova variável de custo chamada tocost, foi criada uma sub-TLV chamada OC (Operational Cost), adicionada ao final da TLV padrão IHU. A modularidade do protocolo Babel permite a inclusão de sub-TLVs desde que sejam relacionadas ao contexto da TLV principal. Essa sub-TLV OC, junto com a TLV IHU, é usada para informar o custo e a qualidade dos links aos vizinhos, ajudando a otimizar a detecção de mudanças na topologia e a reduzir o overhead na rede, sem a necessidade de criar novas TLVs.

A sub-TLV OC ocupa 32 bits. Os primeiros 16 bits formam seu cabeçalho, enquanto os outros 16 armazenam o valor de tocost. O cabeçalho define o tipo da sub-TLV, com o valor fixado em 4, que identifica essa sub-TLV no código, e o tamanho, fixado em 16, correspondente ao formato unsigned short int do tocost. O bit mais significativo do cabeçalho é definido como zero, indicando que essa sub-TLV não é obrigatória. Isso significa que, se um nó não reconhecer essa sub-TLV, ele pode ignorar seus dados sem comprometer a leitura das demais informações da TLV, garantindo compatibilidade com versões anteriores do protocolo Babel.

A construção de mensagens IHU no protocolo Babel começa com a chamada de uma função que prepara os dados a serem enviados como resposta aos vizinhos. Essa função recebe como parâmetros o vizinho a ser respondido e sua interface correspondente. Durante sua execução, ela calcula e organiza os valores necessários para a mensagem, incluindo a variável de custo operacional (tocost), que é incorporada à TLV no formato de um número inteiro (Figura 9.

Os cálculos incluem variáveis essenciais, como o intervalo entre mensagens Hello recebidas, a latência (RTT), e o custo do enlace (rxcost). Esses valores são utilizados para descrever a qualidade do enlace entre os nós e permitir uma comunicação eficiente. O valor de tocost é calculado com base no custo operacional configurado no nó (Figura 10 linha 2065) e então incorporado à mensagem. A função de montagem organiza todos os dados em um pacote IHU (Figura 10 linha 2067), começando pelo

Figura 9 – Trecho de código arquivo do message.c

```
2000 void
2001 send_ihu(struct neighbour *neigh, struct interface *ifp)
2002 {
2003     int rxcost, tocost, interval;
2004     int send_rtt_data;
2005     int unicast;
```

Fonte: Autora

cabeçalho, que também indica o tamanho total da mensagem.

Figura 10 – Definição do tocost - arquivo do message.c

```
2065     tocost = default_router_operacional_cost;
2066
2067     buffer_ihu(unicast ? &neigh->buf : &ifp->buf,
2068                ifp, rxcost, tocost, interval, neigh->address,
2069                send_rtt_data, neigh->hello_send_us,
2070                time_us(neigh->hello_rtt_receive_time));
2071
```

Fonte: Autora

A montagem da mensagem é iniciada com a inclusão do cabeçalho, que também armazena o tamanho total da mensagem (Figura 11 - linhas 1972 a 1975). O tamanho da mensagem IHU pode variar dependendo da presença de sub-TLVs, sendo que a sub-TLV de custo operacional é opcional, pois seu tipo é definido como não mandatório. O protocolo verifica primeiramente a existência do parâmetro *tocost* (Figura 11 - linha 1973) para determinar se a sub-TLV deve ser incluída. Caso o parâmetro esteja presente, são adicionados 4 bytes ao tamanho da mensagem: 2 bytes para o campo que indica o tamanho e 2 bytes para a área de dados, que contém o valor da variável *tocost*.

Depois do cabeçalho ser definido, o restante da mensagem vai sendo composta pela diversas concatenações. Os dados obrigatórios de um pacote TLV IHU (Figura 11 - linhas 1976 à 1979) inclui a variável *rxcost*, que é essencial para o cálculo da métrica de custo pelo nó que receberá a mensagem, e a variável *interval*, responsável por indicar o intervalo de tempo entre mensagens Hello recebidas. Os dados opcionais da mensagem IHU (Figura 11 - linhas 1984 à 1996) são anexados ao corpo final da estrutura, neles são inseridos a lista de sub-TLVs necessárias, desde que tenham a mesma proposta de utilização da TLV pai.

A estrutura da mensagem IHU permite adicionar tanto a sub-TLV *TIMESTAMP* como também a sub-TLV *Operational Cost*, tendo ambas a mesma estrutura de campos a serem inseridos, similar à TLV pai. A estrutura é composta inicialmente pelo

campo de tipagem, referente ao id da sub-TLV; o campo de tamanho, que armazena o tamanho do corpo que contém os dados da sub-TLV; e por fim o corpo da sub-TLV, usado para os dados necessários a serem transportados pela mensagem, incluindo assim a variável *tocost*. A inclusão de cada novo vizinho na rede, quando esse é recém descoberto, passa pelo processo de criação e inicialização de diversos dados referente a ele.

Os dados de RTT são essenciais para identificação de congestionamentos, evitando assim determinados links que estejam sobrecarregados. Os dados de tempo da mensagem IHU e o histórico das mensagens *Hello*s servem para o cálculo do *rxcost*; e por fim os dados de *txcost* e *tocost* são enviados aos vizinhos, para que estejam cientes dos custos de cada nó na rede.

Figura 11 – Montagem IHU - arquivo do message.c

```
1965 void
1966 buffer_ihu(struct buffered *buf, struct interface *ifp, unsigned short rxcost, unsigned short tocost,
1967            unsigned short interval, const unsigned char *address,
1968            int rtt_data, unsigned int t1, unsigned int t2)
1969 {
1970     int msglen, ll;
1971
1972     ll = linklocal(address);
1973     msglen = (ll ? 14 : 22) + (rtt_data ? 10 : 0) + (tocost ? 4 : 0 );
1974
1975     start_message(buf, ifp, MESSAGE_IHU, msglen);
1976     accumulate_byte(buf, ll ? AE_IPV6_LOCAL : AE_IPV6);
1977     accumulate_byte(buf, 0);
1978     accumulate_short(buf, rxcost);
1979     accumulate_short(buf, interval);
1980     if(ll)
1981         accumulate_bytes(buf, address + 8, 8);
1982     else
1983         accumulate_bytes(buf, address, 16);
1984     if(rtt_data) {
1985         accumulate_byte(buf, SUBTLV_TIMESTAMP);
1986         accumulate_byte(buf, 8);
1987         accumulate_int(buf, t1);
1988         accumulate_int(buf, t2);
1989     }
1990     if(tocost) {
1991        //fprintf(stderr,"rocost setado msg %hi\n", rocost);
1992        accumulate_byte(buf, SUBTLV_OPERATIONAL_COST);
1993        accumulate_byte(buf, 2);
1994        accumulate_short(buf, tocost);
1995     }
1996     end_message(buf, MESSAGE_IHU, msglen);
1997 }
```

Fonte: Autora

Durante o processo de troca de mensagens, utilizando pacotes TLV, todo nó pertencente à rede babel possui um interpretador responsável por ler os dados recebidos e validar a estrutura da respectiva TLV. O interpretador identifica e salva as informações que constam dentro de cada tipo de pacote, para então utilizar esses dados dentro do preenchimento da tabela de rotas.

A modificação realizada dentro da TLV IHU, com a inclusão da nova sub-TLV OC, envolve primeiramente a inclusão do novo valor ao parser para que possa ser lido (Figura 12 - linhas 238 e 239), e a definição do seu valor inicial como zero (Figura 12 - linhas 242 e 243). O valor zero possibilita a utilização do protocolo em redes antigas ou já existentes, já que impede que a somatória de sua métrica de custo seja alterada pelo parâmetro *tocost*, caso o mesmo não esteja sendo usado.

Figura 12 – Parser sub TLV OC - arquivo do message.c

```
233 static int
234 parse_ihu_subtlv(const unsigned char *a, int alen,
235                  unsigned int *timestamp1_return,
236                  unsigned int *timestamp2_return,
237                  int *have_timestamp_return,
238                  unsigned short *tocost_return,
239                  int *have_tocost_return)
240 {
241     int type, len, i = 0;
242     int have_timestamp = 0;
243     int have_tocost = 0;
244     unsigned int timestamp1 = 0, timestamp2 = 0;
245     unsigned short tocost;
```

Fonte: Autora

No decorrer do interpretador responsável pela leitura da TLV IHU, é feita inicialmente uma verificação de cada tipo de sub-TLV que existe dentro da mensagem, é recursivo e também inclui o novo tipo definido para o custo operacional (Figura 13 - linha 277). O byte mais significativo sempre está na primeira posição, e para que o número, considerando sua forma binária, não chegue invertido no host, é preciso realizar uma conversão no valor que representa o parâmetro *tocost*, para que este chegue na ordem de bytes do host e seja corretamente lido (Figura 13 - linha 279).

Após a leitura do parâmetro e seu reconhecimento por parte do pares, o valor *tocost* é então guardado para ser usado, e os parâmetros de indicação tanto da existência quanto da validade da sub-TLV OC são setados para a condição verdadeira (Figura 13 - linhas 300 a 307).

Após a validação e leitura da sub-TLV (Figura 14 - linha 760), os novos valores recebidos, que passam a ser utilizados pelo protocolo, também são comparados com os valores já armazenados anteriormente (Figura 14 - linha 765), para identificação de mudanças na topologia da rede. O resultado dessa comparação, nesse caso para os valores *txcost* e *tocost*, será informado à função responsável por realizar todo o cálculo de atualização da métrica de custo (Figura 14 - linha 770).

A existência de diferenças entre os dados recebidos e os anteriormente existentes significa que o respectivo nó, de onde partiu a TLV IHU, possui mudanças de custo para sua operação ou alguma variação na qualidade da rota, que exige a atualização da tabela de rotas com os caminhos mais coerentes para a situação da rede.

Figura 13 – Reconhecimento parser sub TLV OC - arquivo do message.c

```
277         } else if(type == SUBTLV_OPERATIONAL_COST) {
278           if(len >= 2) {
279             DO_NTOHS(tocost, a + i + 2);
280               //fprintf(stderr,"tocost recebido %hi\n", tocost);
281               have_tocost = 1;
282             } else {
283               fprintf(stderr,
284               "Received incorrect OPERATIONAL_COST sub-TLV on IHU.\n");
285               /* But don't break. */
286               }
287         } else {
288             debugf("Received unknown%s IHU sub-TLV %d.\n",
289                    (type & 0x80) != 0 ? " mandatory" : "", type);
290             if((type & 0x80) != 0)
291                 return -1;
292         }
293 
294         i += len + 2;
295     }
296     if(have_timestamp && timestamp1_return && timestamp2_return) {
297         *timestamp1_return = timestamp1;
298         *timestamp2_return = timestamp2;
299     }
300     if(have_timestamp_return)
301         *have_timestamp_return = have_timestamp;
302     if (have_tocost)
303         *tocost_return = tocost;
304     if (have_tocost_return)
305         *have_tocost_return = have_tocost;
306     return 1;
307 }
```

Fonte: Autora

Figura 14 – Função atualizada da métrica - arquivo do message.c

```
760               rc = parse_ihu_subtlv(message + 8 + rc, len - 6 - rc,
761                                     &hello_send_us, &hello_rtt_receive_time,
762                                     NULL, &tocost, NULL);
763               if(rc < 0)
764                   goto done;
765               changed = txcost != neigh->txcost || neigh->tocost != tocost;
766               neigh->txcost = txcost;
767               neigh->tocost = tocost;
768               neigh->ihu_time = now;
769               neigh->ihu_interval = interval;
770               update_neighbour_metric(neigh, changed);
```

Fonte: Autora

Sendo um protocolo pró-ativo, o Babel busca manter suas tabelas de rotas constantemente atualizadas para usar os melhores enlaces em termos de qualidade de serviço (QoS). As mensagens IHU, que incluem informações como o custo operacional, são enviadas em intervalos regulares, mas com menos frequência do que os pacotes Hello Multicast, que são mais eficientes para detectar alterações rápidas na rede.

4.5 AMBIENTE DE TESTE

As análises que serão feitas englobam atrasos de envio de pacote, latência, níveis de throughput, com taxa de recebimento de pacotes, dentro de um ambiente de teste virtual. Foi utilizado para execução o módulo babeld, tanto da versão 1.13.1 original, quanto da versão 1.13.1 contendo alterações para inclusão do *tocost*. Para execução, foi integrado à ferramenta INET framework (versão estável 4.5) dentro do software Omnet++ (versão estável 6.0.3), um simulador de evento discreto baseado na linguagem C++, e usado para gravação e análise de pacotes o software Wireshark (versão 3.6.2), com intuito de aferir os resultados dos testes.

As ferramentas que foram usadas para implementação dos testes foram escolhidas por causa da extensa documentação disponível das mesmas e também pela familiaridade existente entre as ferramentas e o laboratório onde os testes serão realizados, uma vez que já foram utilizadas em diversas outras pesquisas.

Todos os ambientes de testes tiveram seus testes executados em 10 repetições, com os resultados obtidos através da média das mesmas e utilizarou um mesmo hardware para medida de comparação. O computador que foi utilizado possui um sistema Ubuntu 22.04.4 LTS de 64 bits, com o GNOME na versão 42.9. Os componentes principais de hardware são um processador AMD® Ryzen 5 5500u with radeon graphics × 12, memória de 32GB e um disco rígido de 256GB de armazenamento.

A configuração e inicialização do ambiente foi feita através da adição de um novo *namespace* de rede para cada participante, com a inclusão de um dispositivo em cada. Cada interface virtual teve o seu modo de execução setado para *Tap*, configurando assim o tráfego na camada de enlace. Foi configurado, também, um endereço IP para cada dispositivo incluído, preparando o ambiente para o protocolo Babel usado pela camada de redes.

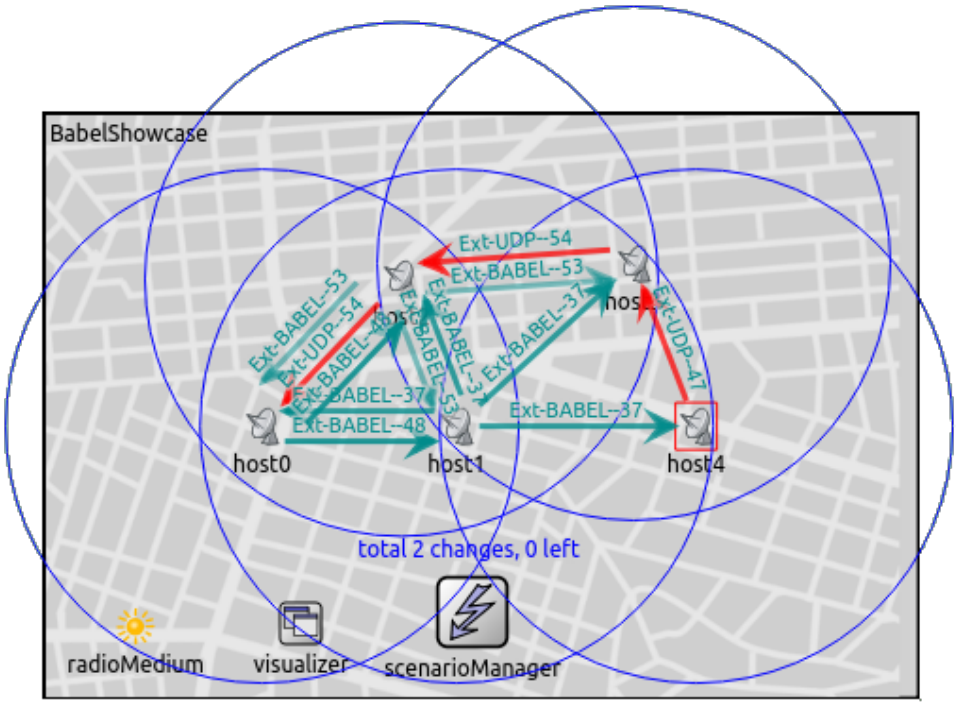
No exemplo demonstrado a seguir é possível identificar as configurações realizadas para um único host, dentro do primeiro ambiente de testes:

```
$ sudo ip netns add host0

$ sudo ip netns exec host0 ip tuntap add mode tap dev tap0
```

```
$ sudo ip netns exec host0 ip link set dev tap0 up

$ sudo ip netns exec host0 ip addr add 192.168.2.1/24 dev tap0
```

O primeiro ambiente de teste foi idealizado para simular 5 nós mesh interligados por meio de uma topologia de anel (Figura 15), sempre mantendo uma distância mínima para que a rede wireless alcance pelo menos 2 outros nós.

Figura 15 – Ambiente de teste



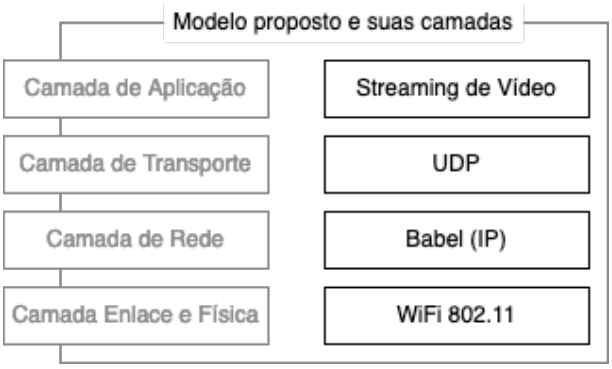
Fonte: (Elaborado pelo autora, 2024)

A simulação de tráfego dos pacotes nos ambientes de teste sempre ocorre entre dois nós da rede (*host0* e *host4*). O *host4* faz uma requisição de pacotes ao *host0*. A simulação acontece conforme as camadas da (Figura16):

- A camada de aplicação simula um Streaming de Vídeo configurado dentro do INET através do pacote ”UdpBasic”.
- A camada de transporte é configurada para utilização do UDP, que não contém garantia na entrega dos pacotes, sendo feitas algumas análises de throughput e nas taxas de perdas de pacotes.
- A camada de rede é composta pelo Babel, responsável por definir as tabelas de roteamento dentro da camada IP, para que cada pacote tenha um próximo salto em direção ao seu destino.

- Por fim a camada física que simula o tráfego de dados por meio de sinais de rádio WiFi 802.11.

Figura 16 – Modelo de Camadas proposto



Fonte: (Elaborado pelo autora, 2024)

A execução do protocolo Babel permite diversas mudanças e adaptações dentro dos argumentos da sua chamada pelo terminal, permitindo diversos comportamentos de acordo com a rede onde o protocolo será executado. Na simulação feita dentro dos dois ambientes virtuais, os argumentos utilizados para execução do protocolo foram os mesmos, exceto pela variável de custo operacional, que foi atribuída com valores diferentes para permitir justamente a análise do roteamento a nível da camada de redes, e simulando a utilização em ambientes reais, onde os custos operacionais costumam ser diferentes.

A Figura 17 demonstra as chamadas realizadas no primeiro ambiente de testes, assim como os argumentos utilizados em cada um dos 5 nós presentes na rede.

Figura 17 – Comandos de execução

```
# start babel routing daemons
sudo ip netns exec host0 babeld -I babel0.pid -S babel-state0 -r -w -h 2 -M 0 -C 'reflect-kernel-metric true' -C 'interface tap0 link-quality false' -Y 2 &
sudo ip netns exec host1 babeld -I babel1.pid -S babel-state1 -r -w -h 2 -M 0 -C 'reflect-kernel-metric true' -C 'interface tap1 link-quality false' -Y 2 &
sudo ip netns exec host2 babeld -I babel2.pid -S babel-state2 -r -w -h 2 -M 0 -C 'reflect-kernel-metric true' -C 'interface tap2 link-quality false' -Y 2 &
sudo ip netns exec host3 babeld -I babel3.pid -S babel-state3 -r -w -h 2 -M 0 -C 'reflect-kernel-metric true' -C 'interface tap3 link-quality false' -Y 2 &
sudo ip netns exec host4 babeld -I babel4.pid -S babel-state4 -r -w -h 2 -M 0 -C 'reflect-kernel-metric true' -C 'interface tap4 link-quality false' -Y 2 &
```

Fonte: Autora

As configurações adotadas na execução do Babel foram os seguintes argumentos:

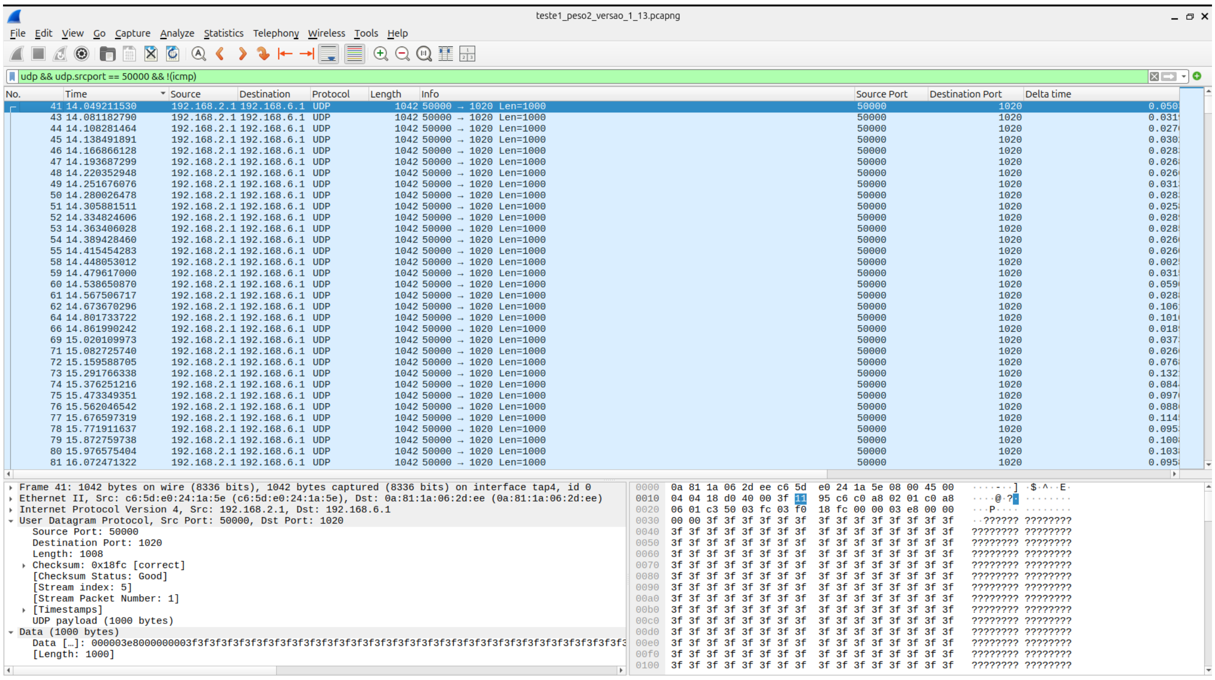
- *I* especifica um arquivo para registrar o id do processo;
- *S* especifica um arquivo usado para preservar informações do babel de longo prazo entre cada execução;
- *r* habilita a seleção aleatória de id para cada roteador;
- *w* desabilita otimizações para redes cabeadas, assumindo que todos os links possuem conexão sem fio;
- *h* especifica o intervalo em segundos entre o envio de cada pacote *Hello*;

- *M* especifica o intervalo em segundos do decaimento exponencial para suavizar as métricas durante a seleção da rota;
- *C* especifica comandos a serem inseridos diretamente na linha de comando do host;
- Por fim, como último argumento, foi adicionado o termo *Y* referente ao custo do nó, que pode ser representado por uma moeda ou qualquer tipo de incentivo a ser adotado.

Cabe ressaltar que para o primeiro teste todos os hosts foram inicializados com valor de Y = 2 e para o segundo teste com o valor y = 3000 para o host1, isso se deve ao fato de que o caminho mais curto e com menos saltos a ser realizado do host0 origem ao host4 destinho é por meio da rota, host0 - host1 - host4, logo para verificar o comportamento aumentou-se o custo do menor caminho para verificar a nova rota escolhida pelo protocolo.

Já para a captura e análise de testes de rede no protocolo Babel, uma das ferramentas essenciais foi o Wireshark, um analisador de pacotes amplamente utilizado que permite a captura e inspeção detalhada do tráfego de rede. O Wireshark oferece uma interface gráfica (Figura 18), capaz de exibir pacotes de rede em tempo real, facilitando a análise do comportamento do protocolo Babel. Para utilizá-lo de forma eficaz, foi necessário configurar adequadamente o ambiente de rede, garantindo que as interfaces de captura estivessem corretas e que o tráfego desejado estivesse visível.

Figura 18 – Interface Wireshark



Fonte: (Elaborado pelo autora, 2024)

Ao realizar a análise com o Wireshark, foi possível examinar os pacotes do protocolo Babel, verificar as rotas e entender o fluxo de informações dentro da rede. A ferramenta permitiu visualizar os detalhes dos pacotes, como os campos de cabeçalho, rotas redistribuídas e métricas associadas, proporcionando uma visão completa do tráfego de rede. Após a gravação de captura do intervalo de 60 segundos estipulado para todos os testes, filtrou-se apenas para os pacotes UDP, para então realizar análise e comparativos.

Para tanto, todas as gravações tiveram o intervalo de 60 segundos a fim de se aproximar ao máximo do exato intervalo de tempo analisado, tornando os resultados mais fidedignos. Esse valor de intervalo de tempo foi configurado no código para o parâmetro de **.scenarioManager.script*, do arquivo *omnetpp.ini*, linha 243, Figura 19, onde o *script* desliga o módulo do host1 após 60 segundos.

Figura 19 – Inclusão no arquivo omentpp.ini

```
242
243 *.scenarioManager.script = xml("<x><shutdown t='60s' module='host1'/></x>")
244
```

Fonte: Autora

4.6 DEMONSTRAÇÃO DOS RESULTADOS

Os resultados da simulação foram obtidos por meio da contagem de pacotes UDP transmitidos pela rede e que foram entregues com sucesso ao host de destino, buscando identificar o número de pacotes recebidos em função do tempo.

A fim de observar a execução da rede em um cenário de redes mesh, foram realizados vários testes em dois ambientes, o primeiro simulando uma transferência de pacotes com todos os hosts com o valor *tocost* Y = 2 e o segundo apenas com o host1 com o valor *tocost* Y = 3000, forçando com que o melhor caminho fosse o com mais saltos, decorrente da rota de menos saltos ter um custo altíssimo.

4.6.1 Atraso de pacotes - latência

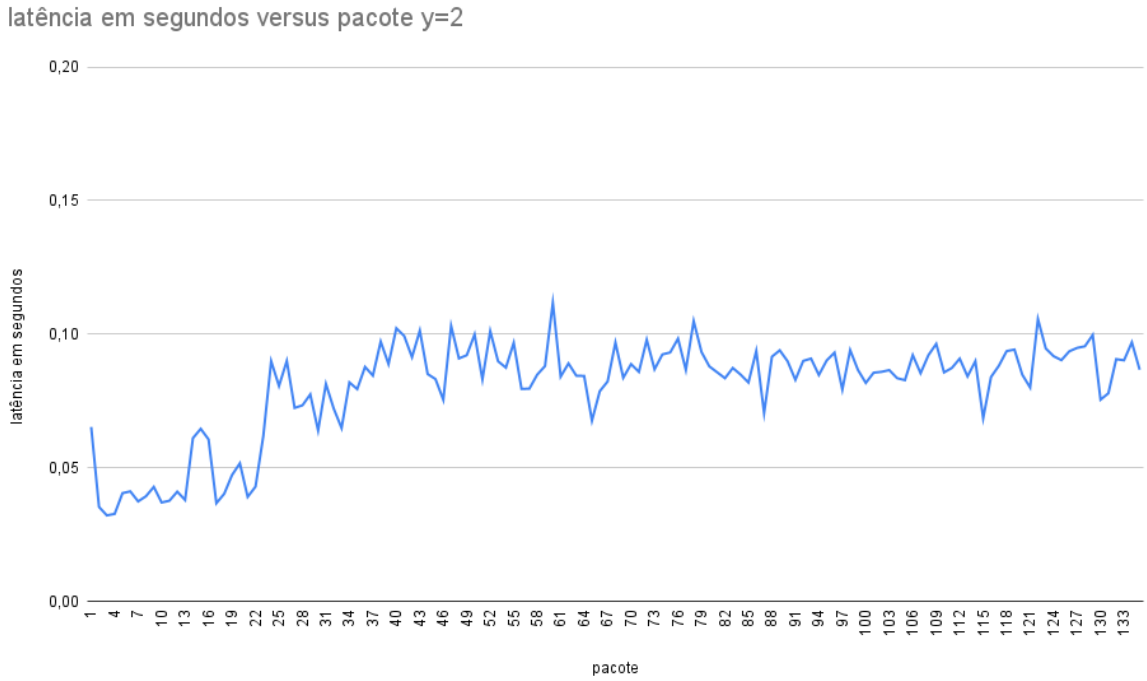
A simulação de teste de atraso de pacotes, ou latência, tem como objetivo analisar o tempo de atraso para cada um dos pacotes registrados na captura do teste, parâmetro fundamental para avaliar a performance de sistemas de comunicação em tempo real, onde a latência pode afetar diretamente a qualidade da rede.

Em uma análise detalhada, ao observar os pacotes UDP enviados em intervalos regulares, é possível obter os atrasos entre pacotes consecutivos. O campo *Delta Time* representa o campo **"Time Since Previous Frame"** indica o tempo decorrido entre a captura do pacote atual e o pacote imediatamente anterior na sequência de

captura. Ele mede a diferença de tempo entre os timestamps desses dois pacotes, fornecendo uma visão detalhada sobre os intervalos entre as transmissões na rede.

Sendo assim, foi possível verificar as variações para os dois cenários de testes realizados, a Figura (20) mostra o gráfico obtido através da média de cada Delta Time, dos dez testes realizados, onde o eixo X representa o número de cada pacote e o eixo Y exibe a latência em segundos, permitindo visualizar a variação do atraso de pacotes e identificando possíveis picos e padrões.

Figura 20 – Gráfico Teste 1 Pacotes X Latência



Fonte: Autora

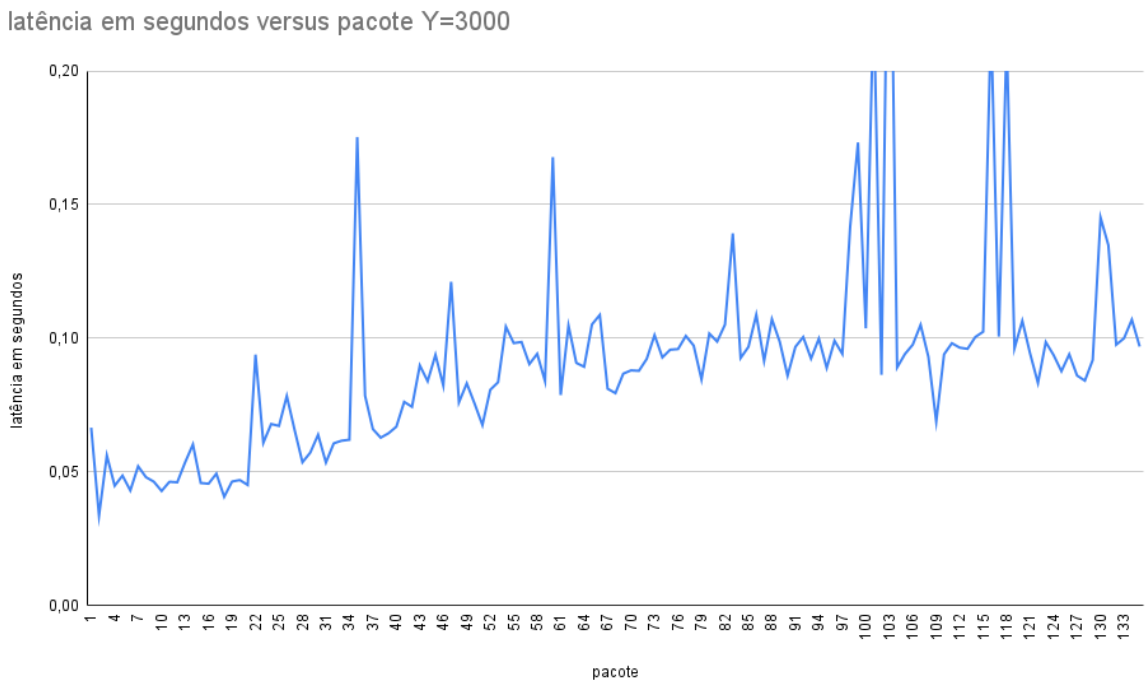
E o mesmo cenário foi simulado para o segundo teste (Figura 21), onde host1 possui o valor do *tocost* é Y = 3000.

4.6.2 Taxa de recebimento de pacotes

Para analisar a taxa de recebimento de pacotes por segundo, foram realizadas dez simulações em cada um dos dois cenários de teste previamente descritos. Essa abordagem possibilitou a coleta de dados para avaliar o comportamento da rede em diferentes condições, permitindo identificar variações no desempenho relacionadas à configuração do protocolo.

Como resultado das simulações, foram gerados gráficos que ilustram a quantidade total de Kilobytes transmitidos ao longo do tempo em cada cenário. Nos gráficos, o eixo X representa o momento específico em que a transferência de dados ocorreu,

Figura 21 – Gráfico Teste 2 Pacotes X Latência



Fonte: Autora

enquanto o eixo Y mostra a quantidade de dados transmitidos em Kilobytes. Essa visualização facilita a interpretação das taxas de transmissão e o impacto de diferentes configurações sobre o desempenho da rede.

Para o teste 1 o resultado obtivo foi o gráfico ilustrado na Figura abaixo(22):

Já para o teste 2 o gráfico obtido é demonstrado na Figura(23).

4.7 RESULTADOS EXPERIMENTAIS E ANÁLISE

Os resultados experimentais deste trabalho foram obtidos a partir de simulações realizadas dez vezes para cada um dos dois cenários propostos, buscando representar, de forma mais próxima, as oscilações que podem ocorrer em redes reais. Essa repetição permitiu identificar padrões e variações no desempenho da rede, aumentando a confiabilidade das análises. A abordagem adotada garantiu que as medições refletissem condições práticas e proporcionassem um conjunto de dados consistente para avaliação.

No primeiro cenário, o objetivo principal foi atribuir valores iguais ao parâmetro **to-cost** para todos os hosts da rede, simulando uma situação em que o peso das rotas era equilibrado e homogêneo. Esse teste foi essencial para observar o comportamento padrão da rede sem intervenções significativas, funcionando como uma base comparativa inicial. A ideia era verificar o desempenho em um ambiente controlado

Figura 22 – Gráfico: tempo por número de dados recebidos

Fonte: Autora

Figura 23 – Gráfico: tempo por número de dados recebidos

Fonte: Autora

e uniforme, com todos os nós apresentando as mesmas condições de prioridade no roteamento.

Já no segundo cenário, realizou-se uma alteração significativa no parâmetro **tocost** de um único host, atribuindo-lhe o valor de **Y = 3000**. Essa mudança simulou uma rede com características assimétricas, onde um dos nós apresentava maior custo de roteamento. O objetivo dessa configuração foi criar um contraste em relação ao primeiro cenário, destacando o impacto dessa diferença nos padrões de transmissão e recepção de pacotes.

A análise comparativa do atraso de pacotes, representada na Figura 24, apresenta os gráficos dos dois cenários de teste sobrepostos, destacando diferenças e semelhanças no comportamento da rede. No primeiro cenário, os resultados indicam um padrão de oscilação que se assemelha à algumas redes cabeadas. Nesse caso, as variações observadas são regulares e previsíveis, características típicas de uma transmissão de dados estável. Esse comportamento reflete uma rede onde as rotas possuem pesos iguais para todos os hosts, garantindo uma distribuição equilibrada e evitando picos significativos de latência.

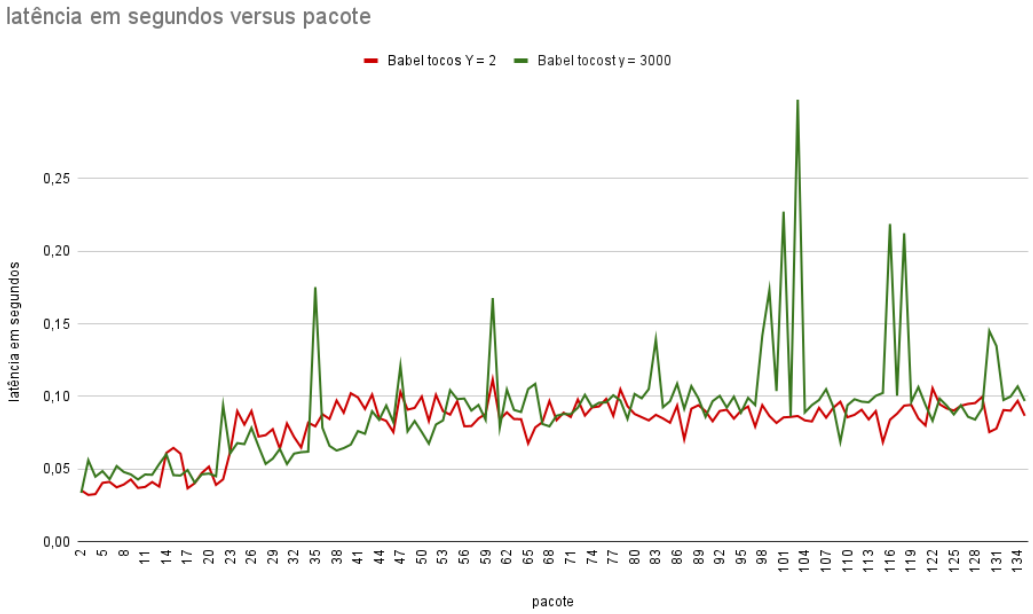
Por outro lado, no segundo cenário, os gráficos revelam picos evidentes de latência em momentos específicos, o que sugere que a rede está buscando rotas alternativas com custos mais elevados do que os previamente utilizados. Esses picos são representativos de eventos de reconfiguração da rota, influenciados pela alteração no parâmetro **tocost** de um dos hosts para o valor Y = 3000.

Apesar dessas diferenças, nota-se que, em ambos os casos, os primeiros segundos apresentam baixa oscilação, indicando um período inicial de convergência da rede. Essa fase inicial é um indicativo de que o protocolo Babel consegue estabelecer rotas básicas rapidamente antes de lidar com as flutuações e reconfigurações mais complexas que surgem em cenários com maior assimetria de custos.

O comparativo baseado no recebimento de pacotes entre os cenários de teste 1 e 2 (Figura 25) revelou diferenças significativas no tempo necessário para que os pacotes atingissem o destino, especialmente durante os primeiros momentos da transmissão. No cenário 2, que apresenta o menor caminho com o maior custo, observou-se um atraso médio de quase 0.4 segundos a mais em relação ao cenário 1 para que os pacotes alcançassem o **host4**. Esse resultado evidencia que redes com maior assimetria de custos podem demorar mais para convergir, uma vez que precisam superar os desafios de rotas menos eficientes no início da comunicação.

Apesar desse atraso inicial, a partir do momento em que a convergência é alcançada, os dois cenários passaram a apresentar comportamentos bastante similares no recebimento de pacotes no destino. Essa semelhança após a fase de convergência reforça a ideia de que, em condições estabilizadas, o protocolo Babel consegue gerenciar a transmissão de forma consistente, mesmo em redes com diferentes configurações de

Figura 24 – Gráfico Latência teste 1 e 2 sobrepostos



Fonte: Autora

custo de rota. Dessa forma, embora o cenário 2 represente uma rede mais complexa e com maior latência inicial, o desempenho final de ambos os testes demonstra a robustez do protocolo em lidar com diferentes condições de rede.

Figura 25 – Gráfico Taxa de Recebimento de pacotes teste 1 e 2 sobrepostos

Fonte: Autora

5 CONCLUSÃO

Este trabalho teve como objetivo inicial a exploração de soluções de segurança para mecanismos de incentivo associados ao uso de blockchains em redes sem fio em malha (WMNs) e para garantir uma base sólida para a implementação dessas soluções, foi necessário, primeiramente, investigar o desempenho do protocolo Babel. Assim, foi realizado um estudo comparativo entre a versão padrão do Babel e uma versão modificada, denominada **Babel Cost**, que incorporou alterações específicas para avaliar o impacto de diferentes configurações de custo nas rotas da rede.

No decorrer do estudo, foram apresentadas as modificações realizadas na versão 1.13.1 do protocolo Babel, com o objetivo de implementar e testar o parâmetro **to-cost** ajustado. Para validar essas mudanças, dois cenários distintos foram simulados, cada um explorando diferentes configurações de rede. Essas simulações permitiram observar tanto os benefícios quanto as limitações das alterações propostas, como o impacto direto no atraso de pacotes e no tempo de convergência. Em particular, identificou-se que, em cenários onde não há rotas alternativas viáveis, o aumento do custo informado por determinados nós pode levar a situações problemáticas, em que rotas menos eficientes ou até com valores altíssimos acabam sendo escolhidas por falta de outras opções.

Além disso, o trabalho incluiu uma revisão abrangente do estado da arte sobre o tema, englobando estudos prévios e abordagens existentes relacionadas ao uso de blockchain e protocolos de roteamento em WMNs. Essa pesquisa permitiu contextualizar o problema, identificar lacunas no conhecimento atual e destacar a relevância de mecanismos de segurança que considerem tanto os incentivos econômicos quanto a integridade do sistema de roteamento.

5.1 CONSIDERAÇÕES FINAIS

A utilização de redes sem fio em malha (WMNs) tem mostrado grande potencial para fornecer conectividade em áreas remotas, superando as limitações da infraestrutura tradicional de rede cabeada, especialmente em locais com dificuldades de acesso. O protocolo Babel, com sua flexibilidade e capacidade de adaptação a diferentes cenários de rede, se mostra uma alternativa viável para essas regiões, onde a implantação de redes convencionais seria economicamente inviável. No entanto, apesar das vantagens observadas, os testes realizados demonstraram que, em cenários com rotas mais complexas ou maior assimetria de custos, o desempenho do Babel ainda pode ser impactado. Isso evidencia a necessidade de um aprofundamento nas pesquisas sobre o protocolo, visando otimizar seu uso em redes mais desafiadoras,

como as encontradas em áreas remotas.

Ao realizar o comparativo entre a versão padrão do Babel e a versão Babel Cost, foi possível identificar os impactos das alterações implementadas, destacando tanto a robustez quanto as limitações do protocolo em cenários com rotas assimétricas.

Essa etapa foi fundamental para pavimentar o caminho para futuras pesquisas, que deverão focar na integração de soluções de segurança baseadas em blockchain para redes sem fio em malha, considerando as particularidades e os desafios observados durante esta análise. A incorporação de mecanismos de segurança, como descentralização, imutabilidade e consenso, se mostra essencial para garantir a integridade e a confiabilidade das redes WMNs, especialmente quando utilizadas para transações e compensações financeiras entre seus participantes. Assim, este estudo inicial contribui para a construção de uma infraestrutura mais segura e resiliente, capaz de enfrentar os desafios de segurança em redes sem fio de grande escala.

5.2 SUGESTÃO PARA TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, uma direção promissora seria a integração de soluções de segurança baseadas em blockchain para redes sem fio em malha (WMNs), levando em consideração as particularidades e os desafios identificados durante esta análise. A combinação da descentralização, imutabilidade e consenso da tecnologia blockchain pode fortalecer a segurança e a confiabilidade das redes, especialmente em cenários onde a comunicação entre os nós é vulnerável a ataques, comportamentos egoístas ou manipulação de dados. A implementação de mecanismos de segurança que assegurem a integridade das rotas de transmissão e a proteção contra fraudes.

Ademais, trabalhos futuros podem investigar como o protocolo Babel, em sua versão modificada, pode se beneficiar da integração da identidade autosoberana baseada em blockchain, juntamente com contratos inteligentes e incentivos, para promover a cooperação e a confiança entre os participantes da rede. Investigar o impacto de tais integrações na performance e na escalabilidade das redes sem fio em malha, especialmente em ambientes com desafios específicos, como áreas remotas, será essencial para avançar na criação de redes mais seguras e de maior alcance.

REFERÊNCIAS

[Akyildiz, Wang e Wang 2005]AKYILDIZ, I. F.; WANG, X.; WANG, W. Wireless mesh networks: a survey. *Computer Networks*, v. 47, n. 4, p. 445 – 487, 2005. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128604003457>.

[Baig et al. 2015]BAIG, R. et al. guifi.net, a crowdsourced network infrastructure held in common. *Computer Networks*, v. 90, p. 150 – 165, 2015. ISSN 1389-1286. Crowdsourcing. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128615002327>.

[Bala et al. 2023]BALA, K. et al. A blockchain-enabled, trust and location dependent - privacy preserving system in vanet. *Measurement: Sensors*, v. 30, p. 100892, 2023. ISSN 2665-9174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2665917423002283>.

[Chroboczek 2021]CHROBOCZEK, J. *Extension Mechanism for the Babel Routing Protocol*. 2021. Disponível em: <https://datatracker.ietf.org/doc/html/rfc8966>.

[Comer 2016]COMER, D. *Interligação de Redes com TCP/IP – Vol. 1: Princípios, Protocolos e Arquitetura*. São Paulo: Pearson, 2016.

[Jonglez e Chroboczek 2019]Jonglez, B.; Chroboczek, J. *Delay-based Metric Extension for the Babel Routing Protocol*. 2019. Disponível em: <https://datatracker.ietf.org/doc/html/draft-ietf-babel-rtt-extension-00>.

[Kumar e Sharma 2022]KUMAR, R.; SHARMA, R. Leveraging blockchain for ensuring trust in iot: A survey. *Journal of King Saud University - Computer and Information Sciences*, v. 34, n. 10, Part A, p. 8599–8622, 2022. ISSN 1319-1578. Disponível em: <https://www.sciencedirect.com/science/article/pii/S131915782100255X>.

[Maccari e Lo Cigno 2015]MACCARI, L.; Lo Cigno, R. A week in the life of three large wireless community networks. *Ad Hoc Networks*, v. 24, p. 175–190, 2015. ISSN 1570-8705. Modeling and Performance Evaluation of Wireless Ad-Hoc Networks. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1570870514001474>.

[Marconi e Lakatos 2011]MARCONI, M. d. A.; LAKATOS, E. M. Técnicas de pesquisa: planejamento e execução de pesquisas, amostragens e técnicas de pesquisa, elaboração, análise e interpretação de dados. *In*: *Técnicas de pesquisa: planejamento e execução de pesquisas, amostragens e técnicas de pesquisa, elaboração, análise e interpretação de dados*. [S.l.: s.n.], 2011. p. xiii–277.

[mesh.org 2013]MESH.ORG open. *B.A.T.M.A.N. IV (TQ)*. 2013. Disponível em: <https://www.open-mesh.org/projects/batman-adv/wiki/BATMAN$_IV$¿.

[Mrabet, Bouanani e Ben-Azza 2023]MRABET, K.; BOUANANI, F. E.; BEN-AZZA, H. Generalized secure and dynamic decentralized reputation system with a dishonest majority. *IEEE Access*, v. 11, p. 9368–9388, 2023.

[Pourrahmani et al. 2023]POURRAHMANI, H. et al. A review of the security vulnerabilities and countermeasures in the internet of things solutions: A bright future for the blockchain. *Internet of Things*, v. 23, p. 100888, 2023. ISSN 2542-6605. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2542660523002111>.

[Reportal 2022]REPORTAL, D. *DIGITAL 2022: GLOBAL OVERVIEW REPORT*. 2022. Disponível em: <https://datareportal.com/reports/digital-2022-global-overview-report>.

[Telecomunicações 2022]TELECOMUNICAçõES, A. A. N. de. *Redes Comunitárias*. 2022. Disponível em: <https://www.gov.br/anatel/pt-br/regulado/universalizacao/redes-comunitarias>.

[Union 2020]UNION, I. I. T. *Digital Development Dashboard*. 2020. Disponível em: <https://www.itu.int/en/ITU-D/Statistics/Dashboards/Pages/Digital-Development.aspx>.

[V et al. 2023]V, N. J. R. et al. Block chain based integrated data aggregation and segmentation framework by reputation metrics for mobile adhoc networks. *Measurement: Sensors*, v. 27, p. 100803, 2023. ISSN 2665-9174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2665917423001393>.

APÊNDICE A – ARTIGO ACADÊMICO

Segurança de Protocolos de Roteamento com Custos Operacionais Associados à Blockchains nas Redes de Nova Geração

Cinthia Carolina Shiratori

[1]Universidade Federal de Santa Catarina (UFSC)

[2]Departamento de Sistemas de Informação Florianópolis, SC – Brasil

cinthia.shiratori@grad.ufsc.br

Abstract. *Wireless Mesh Networks (WMNs) face security challenges that affect their expansion and reliability. The implementation of Blockchain technology emerges as a promising solution to mitigate these vulnerabilities, ensuring authenticity and integrity in routing transactions. This work analyzes the performance of the Babel protocol in Wireless Mesh Networks (WMNs), evaluating its efficiency and effectiveness for the future implementation of unique and immutable digital identities. The results strengthen the security and resilience of the system, paving the way for a more reliable expansion of WMNs.*

Resumo. *As redes de malha sem fio (WMNs) enfrentam desafios de segurança que afetam sua expansão e confiabilidade. A implementação da tecnologia Blockchain surge como uma solução promissora para mitigar essas vulnerabilidades, garantindo autenticidade e integridade nas transações de roteamento. Este trabalho analisa o desempenho do protocolo Babel em Redes Mesh Sem Fio (WMNs), avaliando sua eficiência e eficácia para futura implementação de identidades digitais únicas e imutáveis. Os resultados reforçam a segurança e resiliência do sistema, pavimentando o caminho para uma expansão mais confiável das redes WMNs.*

1. Introdução

A expansão das redes e de sua infraestrutura tem sido um princípio norteador do desenvolvimento econômico e social [de Telecomunicações 2022]. A ampliação no acesso à rede, principalmente em áreas onde a oferta é inadequada, permite que diversos setores de uma sociedade sejam melhorados, como a saúde, educação e segurança para toda a população atendida [de Telecomunicações 2022]. Um dos grandes desafios é levar a conexão de rede para áreas urbanas desatendidas, áreas rurais, e também áreas remotas onde se encontram algumas comunidades.

Os altos custos para implantação da rede cabeada, ainda no ano de 2022, justificam a carência de acesso estável à rede de internet por 37,5% da população mundial [Reportal 2022]. Dentro de áreas mais remotas do território, tais problemas de custos também são somados à qualidade precária das redes existentes, e a falta de sustentação e manutenção dos serviços a longo prazo.

A falta de acesso a rede em áreas remotas é mais evidente em países em desenvolvimento, que possuem um vasto território, como é o caso do Brasil. Considerado

o quinto maior país do mundo em extensão territorial, é também o terceiro país com o maior tempo médio de acesso a rede no mundo [Maccari and Lo Cigno 2015], mas ainda apresenta uma imensa desigualdade no acesso da rede. Em áreas urbanas segundo censo feito em 2020 [Union 2020], cerca de 14% dos lares brasileiros não tinham conexão de rede, já em áreas rurais a situação era ainda pior, com cerca de 35% das residências não possuindo conexão de rede.

Dentro do contexto de desigualdade, as WMN (Wireless Mesh Networks) são redes capazes não apenas de fornecer conectividade sem fio a largas áreas (como shoppings e centros urbanos), mas também levar tal conectividade a locais muito mais remotos do território [Akyildiz et al. 2005]. Por conta da redução de preços dos aparelhos e facilidade em sua implantação, as WMNs levaram ao surgimento de diversas redes comunitárias [Baig et al. 2015], que ajudam a resolver o problema de última milha, relacionado a áreas mais remotas que não possuem acesso à rede.

A fim de garantir a segurança das WMNs é possível associar a tecnologia blockchain e seus três princípios fundamentais de descentralização, imutabilidade e consenso, aplicados para fortalecer a integridade e a confiabilidade da rede e comunicação, buscando criar uma infraestrutura mais resistente e capaz de enfrentar os desafios de segurança nas redes mesh sem fio.

A arquitetura das WMNs promove a descentralização ao distribuir a responsabilidade entre múltiplos nós da rede, reduzindo pontos únicos de falha e aumentando a resiliência e a confiabilidade do sistema, isso reduz a vulnerabilidade a ataques únicos, pois não há um único ponto central que, se comprometido, possa comprometer toda a rede. Já a imutabilidade, no contexto blockchain, garante que os dados são registrados em um bloco e adicionados à uma cadeia, que criam uma conexão sequencial. Assim, qualquer tentativa de manipulação pode ser prontamente identificada, assegurando mais uma vez a integridade dos dados. E o consenso garante que cada operação incluída na blockchain seja validada por vários usuários promovendo a segurança entre nós maliciosos, responsável, também, por tornar a rede confiável.

Entretanto, dentro da área de segurança ainda é possível identificar diversos pontos de vulnerabilidade em ambientes de WMNs, quando utilizada a tecnologia blockchain para compensações financeiras entre os participantes da rede. Os principais pontos associados aos resultados esperados estão atrelados a correção de possíveis vulneabilidades nesse ambiente como, por exemplo, comportamentos egoístas que levam a situações como free-riding, buracos negros (black-holes) com perdas constantes de pacotes que atrasam a comunicação, vandalismos e também ataques que exploram vulnerabilidades dos protocolos, que dificultam a estabilidade da rede e a confiabilidade entre participantes que cooperam com o ambiente de WMN.

2. Metodologia

O trabalho é de natureza aplicada e segue o método Hipotético-Dedutivo, com base nas quatro etapas propostas por Marconi (2011). A primeira etapa consiste em uma revisão sistemática da literatura, abordando artigos, revistas e eventos sobre segurança em redes mesh sem fio (WMNs) e blockchain. A segunda etapa envolve a identificação dos desafios relacionados a esses temas. A terceira etapa foca no planejamento e execução de testes no protocolo Babel para WMNs, comparando versões e analisando a seleção

de rotas e o Expected Transmission Cost (ETX). A última etapa, de falseamento, realiza testes e avaliações das implementações desenvolvidas, considerando análises qualitativas e quantitativas para validar as hipóteses propostas.

O protocolo Babel é uma solução de roteamento dinâmico para redes mesh, oferecendo eficiência e flexibilidade em ambientes complexos. Ao longo do tempo, o protocolo passou por atualizações regulares para melhorar sua performance, segurança e compatibilidade com diferentes redes, corrigindo falhas e introduzindo novos recursos. A versão 1.13, lançada em julho de 2023, trouxe otimizações na redistribuição de rotas, evitando despejos completos e melhorando a eficiência. Também foi adicionada a opção "shutdown-delay-ms" para controle no desligamento do daemon e corrigidos problemas de compilação em sistemas BSD. A versão 1.13.1, lançada dias depois, implementou a opção "probe-mtu" para descartar links com MTU mal configurado e corrigiu um erro de digitação na opção "v4viav6", visando maior robustez e estabilidade para a rede.

A versão 1.13 do protocolo Babel introduziu métricas de custo econômico para o cálculo da seleção de rotas, utilizando o ETX (Expected Transmission Cost) e o valor de custo atribuído a cada nó da rede, denominado tocost (transmission operational cost). Esse valor é repassado aos vizinhos através de pacotes TLV (Type-Length-Value) durante as trocas de mensagens IHU ("I Heard You"). Dentro desses pacotes, foi adicionada uma nova Sub-TLV chamada OC (Operational Cost), responsável por transportar o valor tocost de cada nó para garantir que o custo dos links seja coerente entre todos os nós da rede. O valor tocost é determinado pelo usuário ou responsável pelo nó e pode ser representado por moedas ou tokens, variando conforme as necessidades de cada participante e os custos operacionais específicos de cada nó, como hardware, manutenção e energia. Esse valor é definido durante a inicialização do nó, utilizando um número do tipo *unsigned short int*.

No contexto do protocolo Babel, os custos de rotas são utilizados para calcular métricas que definem os melhores caminhos em uma rede, com o objetivo de calcular a árvore de menor custo a partir de cada nó origem. Cada nó mantém uma tabela de rotas com informações sobre os caminhos disponíveis e suas métricas, que são constantemente atualizadas com base em pacotes Hello enviados periodicamente. A métrica de custo, baseada no ETX (Estimated Transmission Cost), é influenciada por fatores como o custo de recepção (rxcost) e o custo de transmissão (txcost), ambos calculados a partir das probabilidades de sucesso nas transmissões de pacotes Hello entre vizinhos. Adicionalmente, a introdução do custo operacional tocost, que pode ser definido pelo usuário, permite uma adaptação mais flexível do protocolo, levando em consideração fatores como manutenção e consumo de energia. A fórmula de cálculo do custo total inclui o rxcost, txcost, RTT (round-trip time) e o tocost, garantindo que as métricas reflitam a qualidade da rede de forma eficiente e sem gerar loops de roteamento.

Além das métricas de custo, o protocolo Babel adota três condições fundamentais para o funcionamento correto do algoritmo de roteamento: (i) se o custo de um enlace for infinito, o link não pode ser utilizado; (ii) a métrica deve ser estritamente monotônica para evitar loops; e (iii) a propriedade distributiva à esquerda assegura a escolha do melhor caminho. As modificações implementadas na versão 1.13 do Babel incluem a adição do parâmetro tocost, que foi integrado ao código em C para garantir a compatibilidade com redes antigas e permitir a personalização das métricas de custo. O protocolo também assegura que a inclusão desse parâmetro não gere custos negativos, evitando problemas como

loops de roteamento. A flexibilidade do Babel permite que diferentes técnicas de seleção de rotas sejam aplicadas, desde que os princípios básicos sejam seguidos, promovendo a estabilidade e a eficiência das redes sem fio.

O protocolo Babel realiza o roteamento na camada de rede por meio da troca de mensagens entre dispositivos, identificados por um ID único. As mensagens são enviadas utilizando pacotes UDP com TLVs, que podem incluir dados obrigatórios, como o custo do enlace (rxcost) e o intervalo de mensagens Hello, além de dados opcionais, como a sub-TLV OC (Operational Cost), que carrega a variável tocost. Essa sub-TLV foi introduzida para permitir a comunicação sobre o custo operacional dos links de rede, sendo opcional para garantir compatibilidade com versões anteriores do protocolo. O valor de tocost é calculado e inserido nas mensagens IHU, que são usadas para atualizar constantemente a tabela de rotas, permitindo que Babel mantenha seus caminhos de roteamento eficientes, considerando a qualidade do serviço (QoS) e as mudanças na topologia da rede.

3. Resultados e Discussões

Os resultados experimentais deste trabalho foram obtidos por meio de simulações realizadas dez vezes para dois cenários, com o objetivo de representar oscilações típicas de redes reais. No primeiro cenário, todos os hosts da rede possuíam o mesmo valor para o parâmetro tocost, criando um ambiente homogêneo, enquanto no segundo, alterou-se o valor de tocost de um host para 3000, simulando uma rede assimétrica. A análise de latência revelou que o primeiro cenário apresentou oscilações regulares, típicas de redes com rotas equilibradas, enquanto o segundo cenário teve picos de latência, indicando a reconfiguração das rotas devido ao custo maior. A comparação dos tempos de recebimento de pacotes mostrou que, embora o segundo cenário tenha causado atrasos iniciais, ambos os cenários apresentaram comportamentos semelhantes após a convergência, evidenciando a robustez do protocolo Babel em gerenciar redes com diferentes configurações de custo de rota.

4. Conclusão

A utilização de redes sem fio em malha (WMNs) tem grande potencial para fornecer conectividade em áreas remotas, superando limitações das redes cabeadas tradicionais. O protocolo Babel se destaca por sua flexibilidade e adaptação, sendo uma alternativa viável para essas regiões. No entanto, os testes realizados mostraram que em cenários com rotas complexas ou assimétricas, seu desempenho pode ser impactado, apontando para a necessidade de mais pesquisas para otimizar seu uso em redes desafiadoras. O comparativo entre a versão padrão e a versão Babel Cost revelou os impactos das alterações implementadas. Futuros estudos podem focar na integração de soluções de segurança baseadas em blockchain, visando aumentar a confiabilidade e a integridade das redes WMNs, especialmente para transações e compensações financeiras. Além disso, a exploração da identidade autosoberana, contratos inteligentes e incentivos pode promover maior cooperação e segurança, avançando para redes mais seguras e escaláveis, especialmente em regiões remotas.

References

Akyildiz, I. F., Wang, X., and Wang, W. (2005). Wireless mesh networks: a survey. *Computer Networks*, 47(4):445 – 487.

Baig, R., Roca, R., Freitag, F., and Navarro, L. (2015). guifi.net, a crowdsourced network infrastructure held in common. *Computer Networks*, 90:150 – 165. Crowdsourcing.

de Telecomunicações, A. A. N. (2022). Redes comunitárias.

Maccari, L. and Lo Cigno, R. (2015). A week in the life of three large wireless community networks. *Ad Hoc Networks*, 24:175–190. Modeling and Performance Evaluation of Wireless Ad-Hoc Networks.

Reportal, D. (2022). Digital 2022: Global overview report.

Union, I. I. T. (2020). Digital development dashboard.