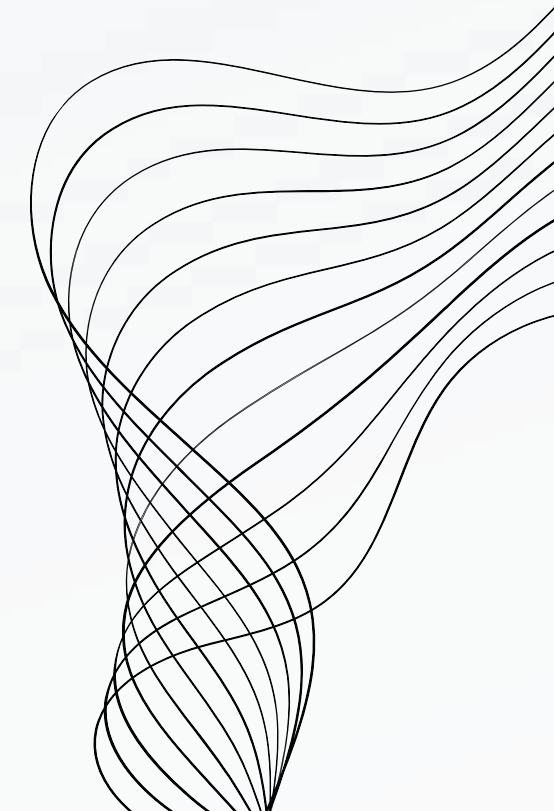
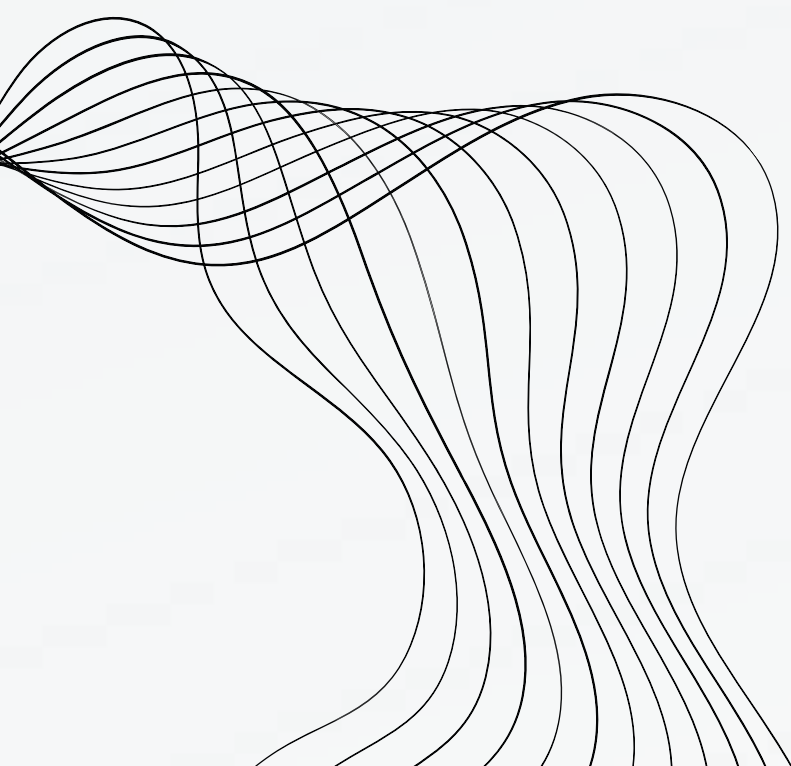


EM BUSCA DA APLICAÇÃO DE PROTOCOLOS DE ROTEAMENTO PARA EVITAR ATAQUES DO TIPO BURACO NEGRO

Alex Luiz Domingues Cassinelli

Orientador: Kelton Augusto Pontara da Costa

Universidade Estadual Paulista “Júlio de Mesquita Filho”
(UNESP) Faculdade de Ciências (FC) / Departamento de
Computação (DCo) Bauru, SP - Brasil



Sumário da Apresentação

- 01** Introdução
- 02** Fundamentação Teórica
- 03** Metodologia
- 04** Experimentação e Resultados
- 05** Conclusão

Introdução

Redes Ad-hoc Móveis (MANET)

As MANETs são redes sem fio, utilizadas em vários tipos de aplicações, como as redes de sensores. Elas permitem a comunicação entre aparelhos (nós) sem a necessidade de conexão física entre os mesmos. A principal forma de encontrar os caminhos entre os nós para realizar a comunicação na rede é através dos protocolos de roteamento.

Segundo Abdel-Fattah et al. (2019), as principais características destas redes são:

- Dinamismo;
- Flexibilidade;
- Mobilidade

Introdução

Protocolo AODV

O protocolo do Vetor de Distância Ad-hoc sob Demanda (AODV) é um dos principais protocolos de roteamento das MANETs, sendo muito eficiente e é capaz de realizar o descobrimento de caminhos a partir do envio e recebimento de pacotes próprios.

O principal objetivo deste protocolo é encontrar o melhor caminho de comunicação entre dois nós da rede, estes, chamados de nó origem e nó destino. É importante dizer que ele mantém apenas um caminho armazenado.

Introdução

Protocolo AOMDV

O protocolo do Vetor de Distância Multicaminhos Ad-hoc sob Demanda (AOMDV) é uma versão modificada do protocolo AODV, sendo capaz de manter armazenado mais de um caminho, caso seja necessária a utilização de uma alternativa, tendo em vista algum imprevisto que pode ocorrer. Estes caminhos buscam minimizar a quantidade de nós em comum entre eles.

Ele possui qualidades semelhantes ao AODV, já que ele modifica apenas a área da armazenagem dos caminhos.

Introdução

Protocolo OWL

O protocolo do Aprendizado da Caminhada Ordenada (OWL) é um algoritmo semelhante ao AODV, porém a base para o mecanismo de descobrimento de rotas é diferente.

Ele é relevante para este trabalho, pois ele proporciona uma perspectiva diferente, assim como conjunto de dados e testes que proporcionaram parte da base teórica para este estudo.

Introdução

Ataques do tipo Buraco Negro

É um tipo de ataque que ocorre em MANETs, especificamente, em decorrência do mecanismo de descobrimento de caminhos que os protocolos de roteamento usam. Além disso, este ataque é muito facilitado devido às características das MANETs, como a facilidade de conexão e a mobilidade.

Neste tipo de ataque, um nó malicioso se insere na rede, a partir disso, este nó “engana” o protocolo de roteamento, fazendo com que os pacotes sejam entregues a ele. Esses pacotes são, então, descartados, portanto, este tipo de ataque pode ser considerado um ataque de negação de serviço (DDoS).

Introdução

Objetivo do Trabalho

O objetivo deste trabalho é estudar se o algoritmo AOMDV, com algumas modificações, é capaz de ignorar um ataque do tipo buraco negro. Ou seja, se, com algumas modificações específicas, o AOMDV pode ignorar as consequências de um ataque na rede, especificamente, a negação de serviço.

Este algoritmo foi testado em um simulador, chamado Network Simulator 2 (NS 2), e foram medidas métricas relevantes, como taxa de entrega e consumo de energia médio dos nós.

Fundamentação Teórica

MANETs

As MANETs surgiram na década de 1970, com aplicações militares. Elas são redes que não precisam de infraestrutura preparada para realizar a comunicação entre dois nós.

As principais características das MANETs, segundo Mirza e López Bakshi (2018) são:

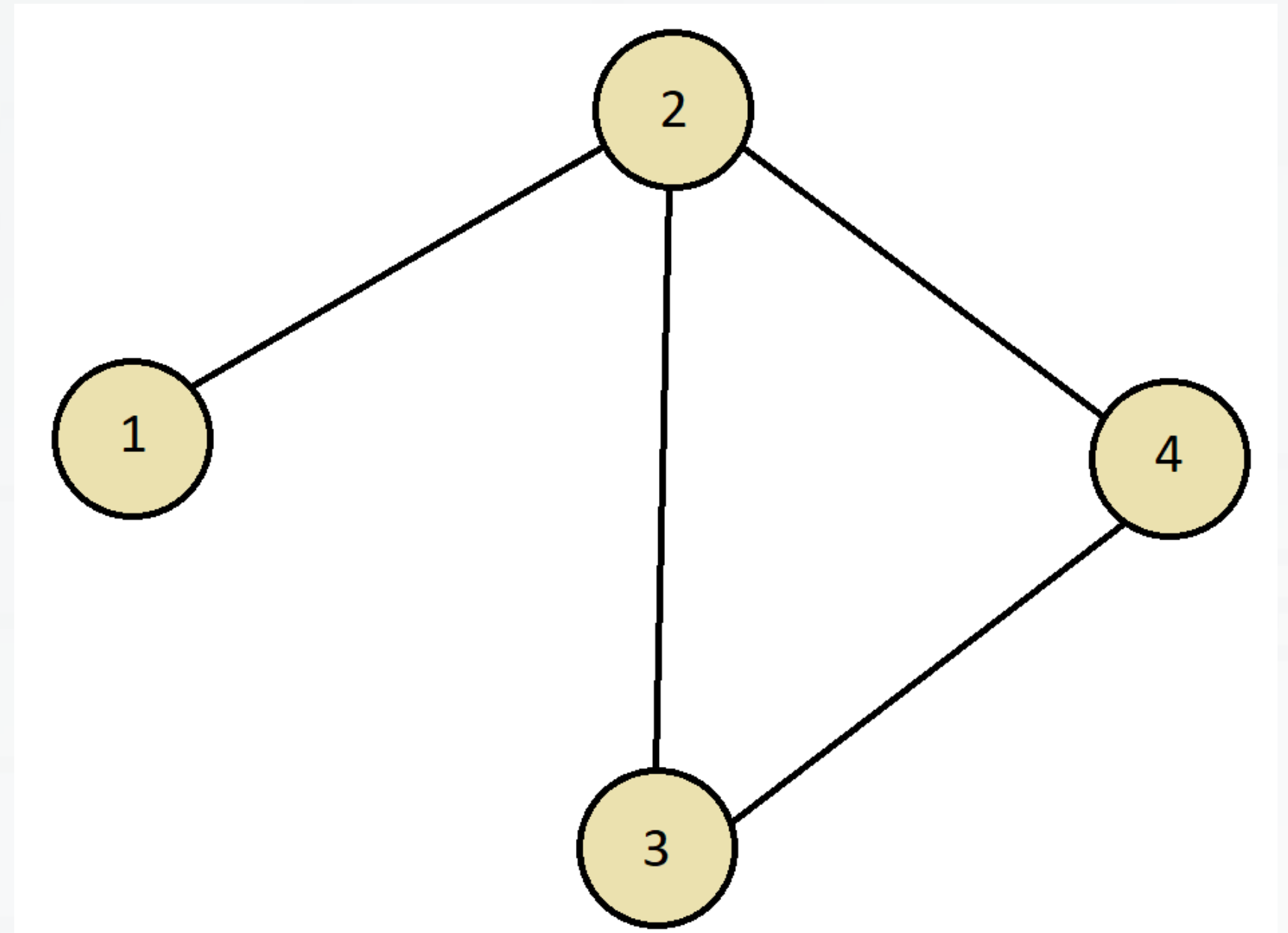
- Operações particionadas;
- Terminais autônomos;
- Roteamento de múltiplos saltos;
- Topologia de rede dinâmica;
- Capacidade de conexão flutuante;
- Terminais leves.

Fundamentação Teórica

Grafos

Grafos são um tipo de estrutura de dados usados para representar sistemas complexos (HAMILTON, 2020). De maneira simples, grafos podem ser descritos como uma coleção de objetos (nós) e uma coleção de interações (arestas).

Ao lado, está a representação gráfica de um grafo.



Fonte: Autor

Fundamentação Teórica

Grafos - Algoritmos de Busca

Para percorrer o grafo em busca de um nó, são utilizados os chamados algoritmos de busca. Os dois mais relevantes para este estudo são a Busca em Largura (BFS) e a Busca em Profundidade (DFS).

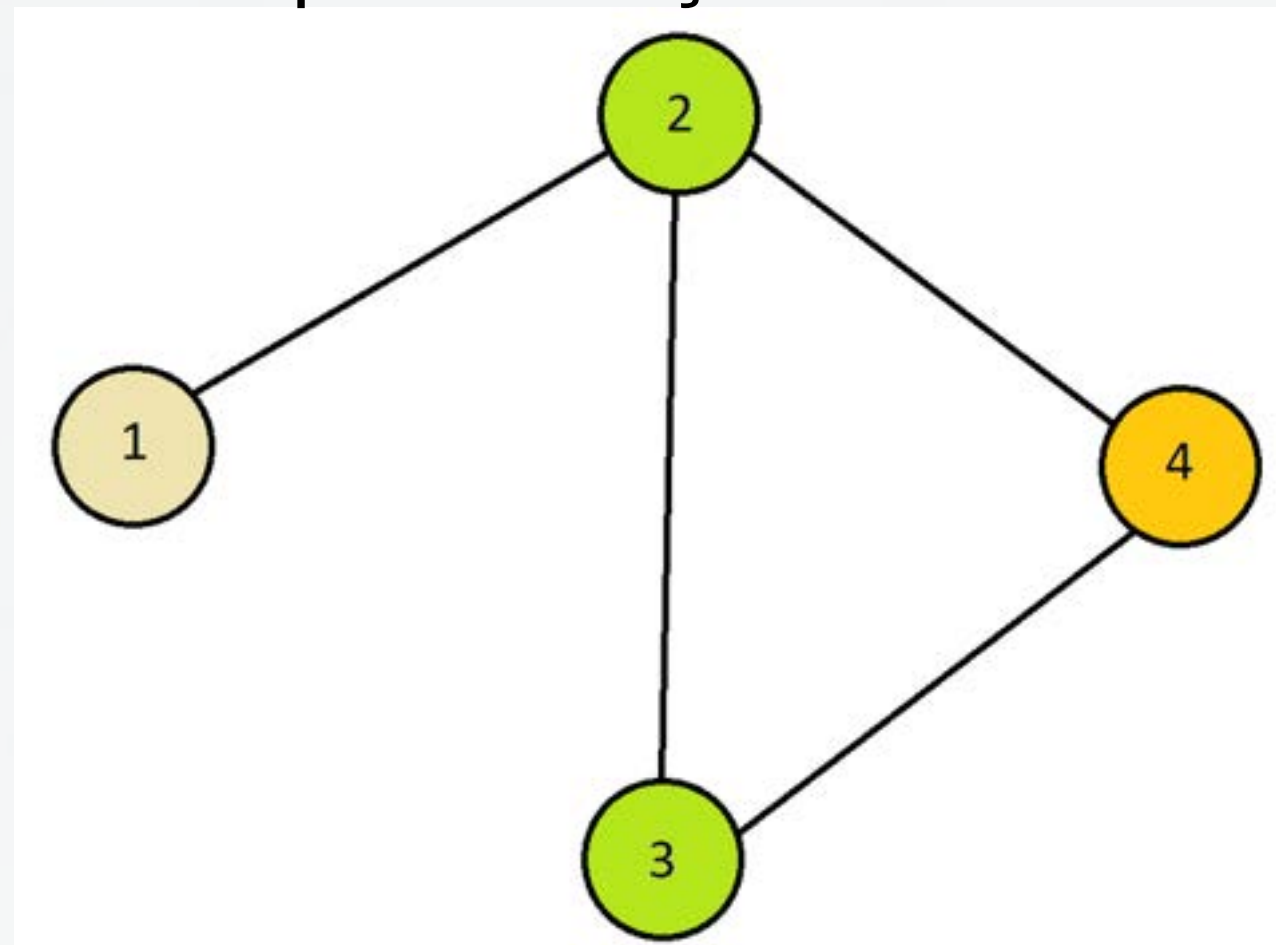
A BFS se baseia em explorar o grafo em “níveis”, analisando todos os nós vizinhos de uma vez só.

Já a DFS mergulha profundamente em um caminho, tentando atingir o destino, antes de explorar outro vizinho.

Fundamentação Teórica

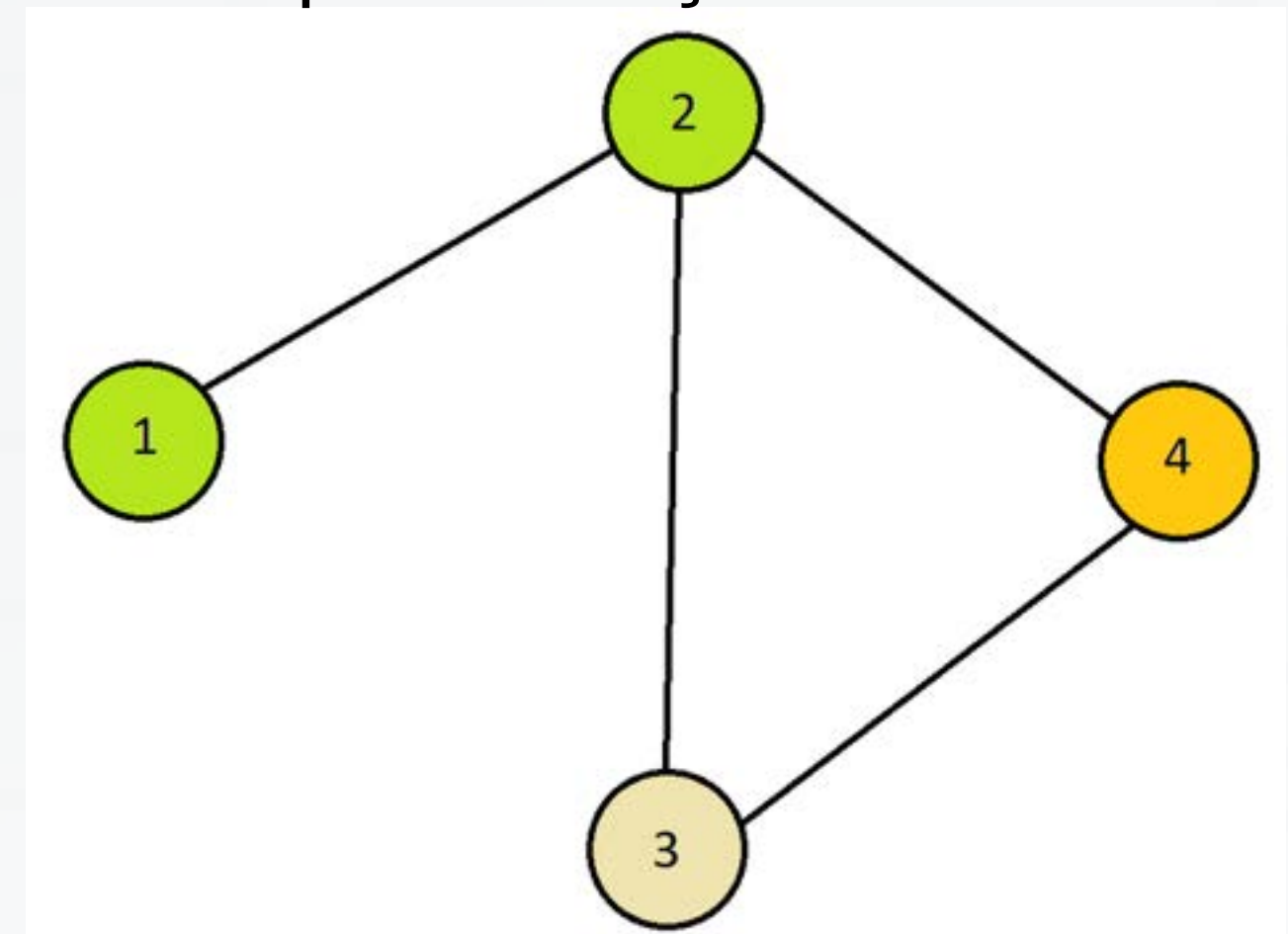
Grafos - Algoritmos de Busca

Representação da BFS



Fonte: Autor

Representação da DFS



Fonte: Autor

Fundamentação Teórica

Descobrimento de Caminhos

O descobrimento de caminhos de uma MANET é realizado pelos algoritmos de roteamento. Neste estudo três deles são relevantes: o algoritmo AODV, o algoritmo AOMDV e o algoritmo OWL.

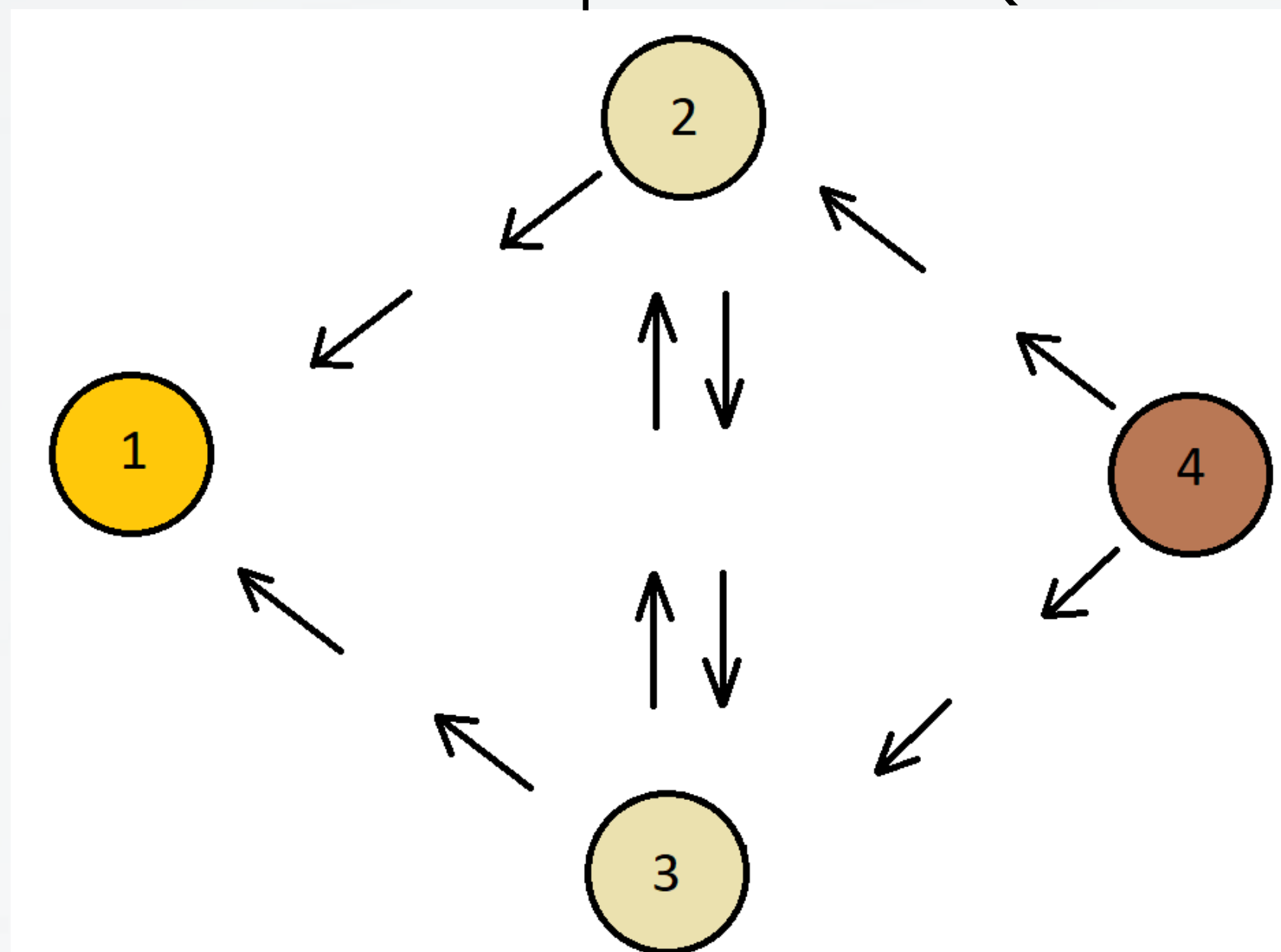
Todos os protocolos apresentados neste trabalho se utilizam do mesmo mecanismo básico: o envio de pacotes de pedido de rota (RREQ) e o recebimento de pacotes de resposta de rota (RREP). De maneira simplificada, o nó origem envia os RREQs e, quando o nó destino é atingido, este envia os RREPs, fechando o caminho.

Fundamentação Teórica

Descobrimento de Caminhos - Um Exemplo

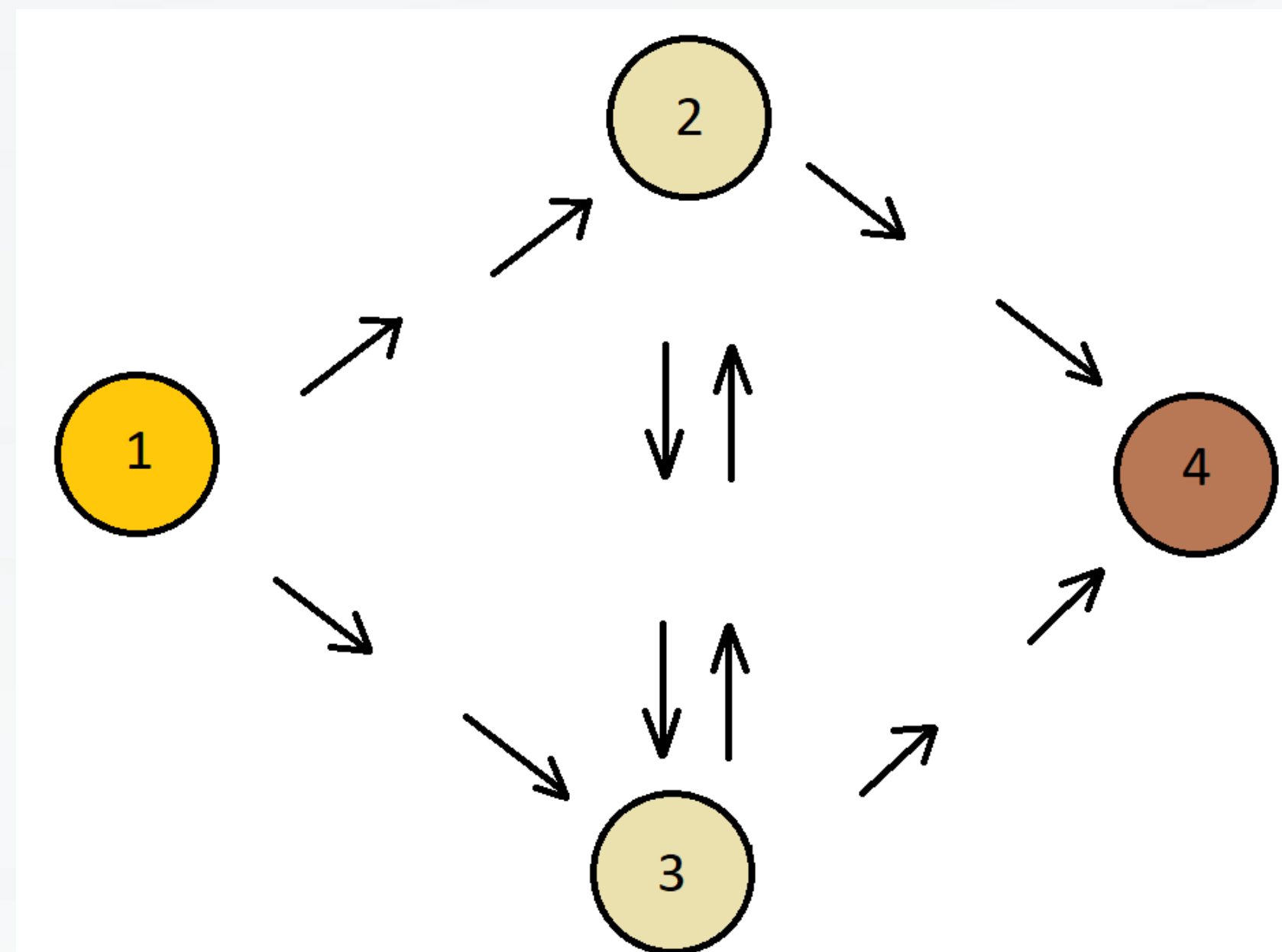
Exemplo: Considere o nó 4 como origem e o nó 1 como destino

Envio dos pacotes RREQs



Fonte: Autor

Envio dos pacotes RREPs



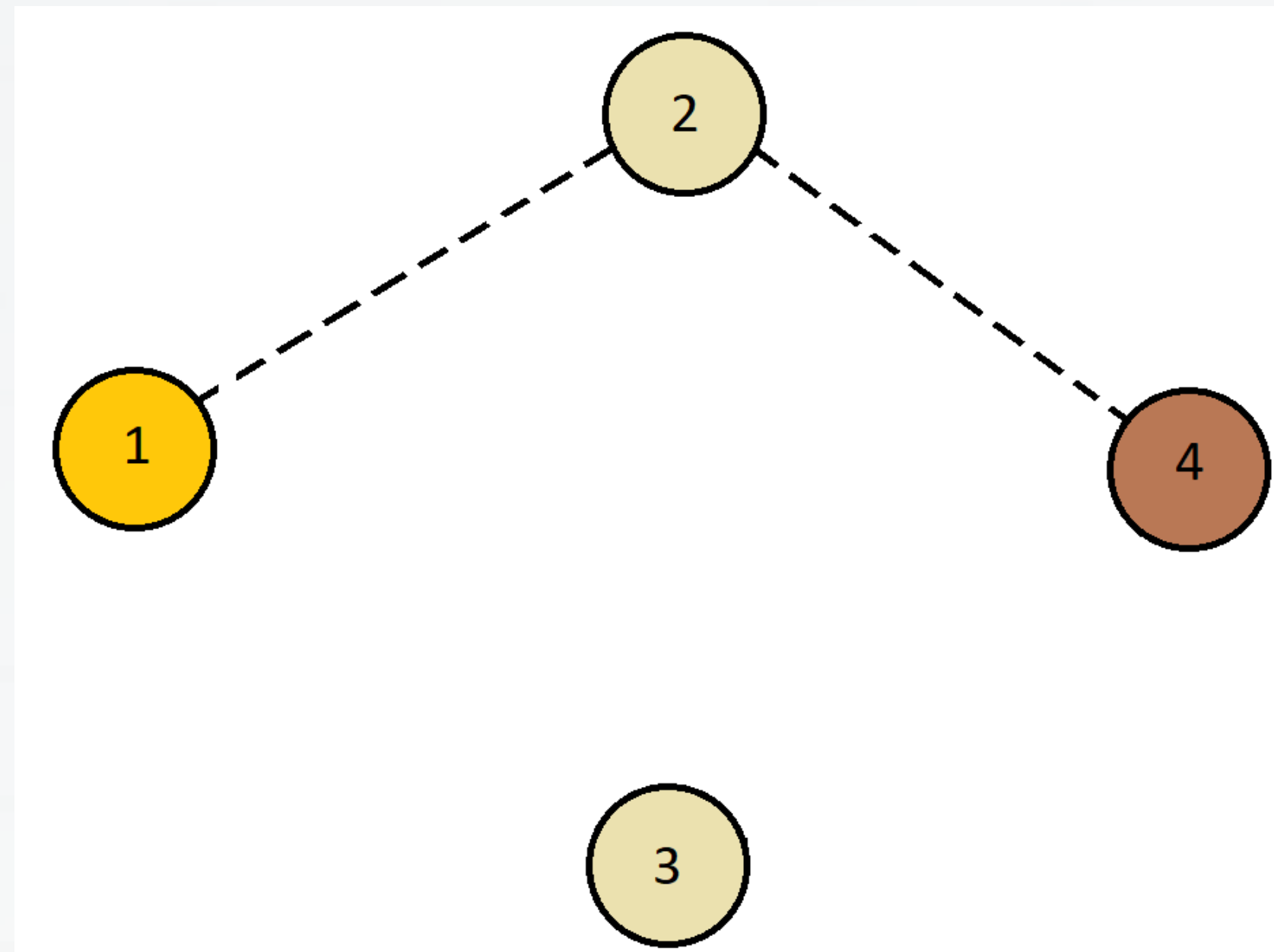
Fonte: Autor

Fundamentação Teórica

Descobrimento de Caminhos - Um Exemplo

Exemplo: Considere o nó 4 como origem e o nó 1 como destino

Estado final da rede



Fonte: Autor

Fundamentação Teórica

Descobrimento de Caminhos

As diferenças entre os algoritmos podem ser vistas quando são comparadas às implementações do envio dos pacotes RREQ e quanto à multiplicidade das rotas viáveis.

O algoritmo AODV se utiliza da BFS para encontrar apenas uma rota ótima. O algoritmo AOMDV faz uso da BFS para encontrar várias rotas possíveis. O algoritmo OWL faz uso da DFS para encontrar uma rota ótima.

Fundamentação Teórica

Descobrimento de Caminhos

Nota-se que cada algoritmo possui pontos positivos e negativos. Normalmente concentrando-se em tempo de envio.

Focando-se apenas nos algoritmos AODV e OWL, é possível fazer uma sintetização destes pontos fortes e fracos: o AODV ainda é mais eficiente, tendo em vista que funciona bem em redes de vários tamanhos, entretanto o OWL não fica muito atrás nos ambientes com menos nós, o que corrobora a DFS como opção viável para estes ambientes, especificamente.

Fundamentação Teórica

Ataques do Tipo Buraco Negro

Os Ataques Buraco Negro é um tipo ataque DDoS que alveja as MANETs, tendo em vista as características da rede, principalmente a mobilidade e facilidade de conexão.

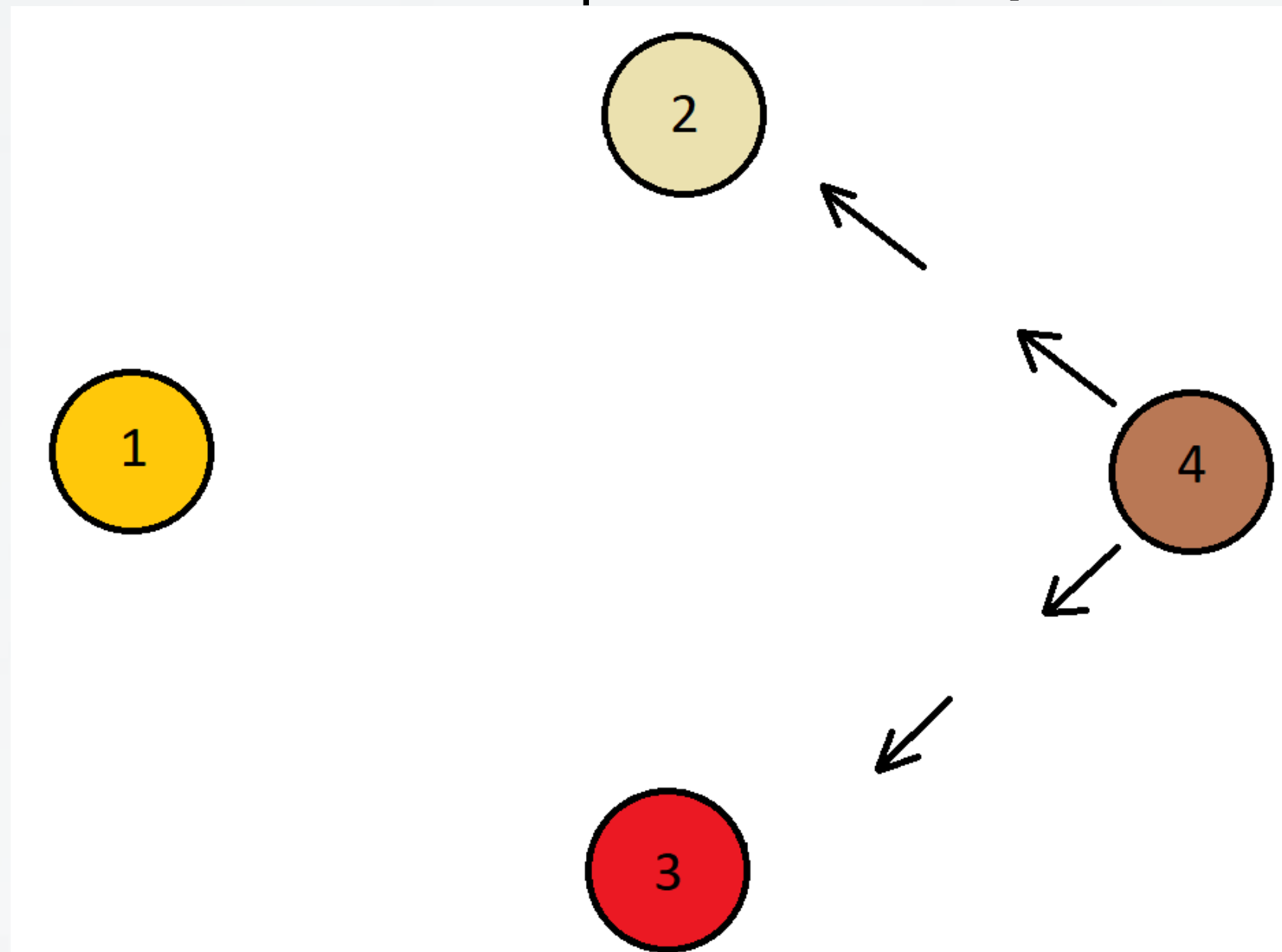
O Ataque Buraco Negro se utiliza da forma como o algoritmo de roteamento interpreta os pacotes RREQ e RREP para se passar por um nó no caminho para um nó destino legítimo: Ao receber um pacote RREQ, o nó malicioso envia o pacote RREP e, caso este pacote malicioso chegue ao nó origem antes do pacote legítimo, o nó invasor é tratado como estando no caminho para o destino.

Fundamentação Teórica

Ataques do Tipo Buraco Negro - Um Exemplo

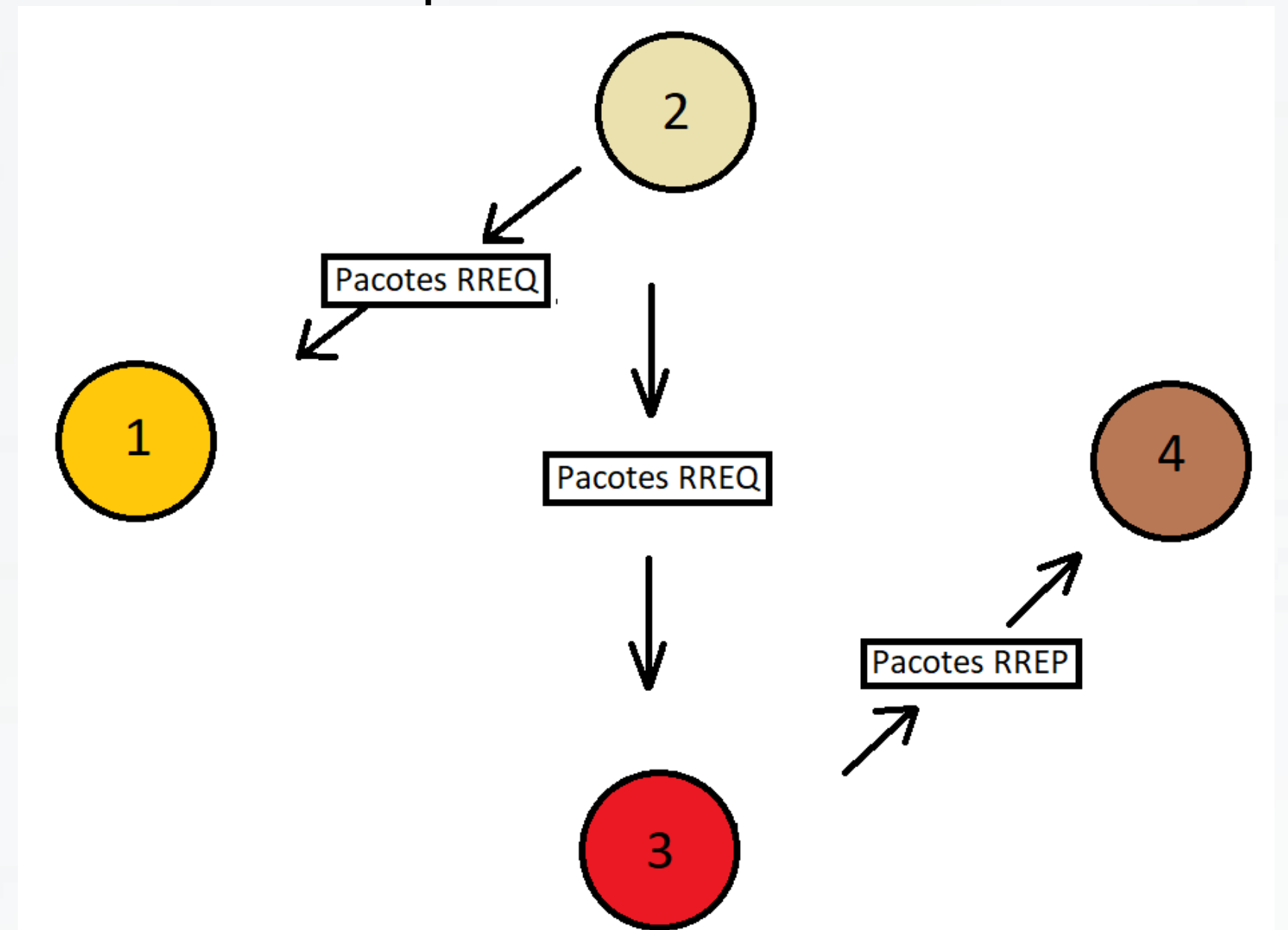
Exemplo: Considere o nó 4 como origem, o nó 1 como destino e o nó 3 como malicioso

Envio dos pacotes RREQs



Fonte: Autor

Envio do pacote RREP malicioso



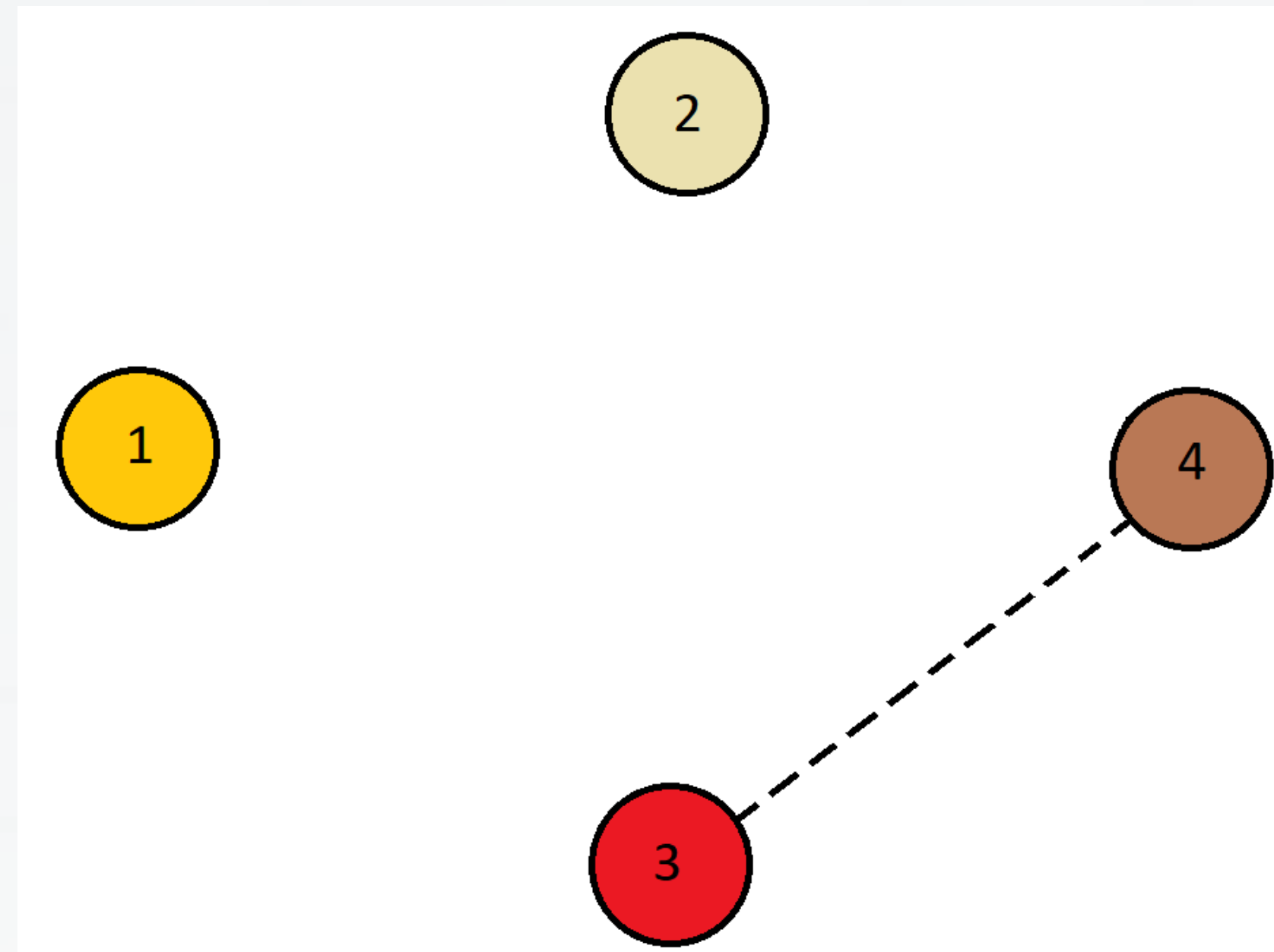
Fonte: Autor

Fundamentação Teórica

Ataques do Tipo Buraco Negro - Um Exemplo

Exemplo: Considere o nó 4 como origem, o nó 1 como destino e o nó 3 como malicioso

Estado final da rede



Fonte: Autor

Metodologia

Modificação do Algoritmo Original

O algoritmo que foi modificado é a implementação do protocolo AOMDV no NS 2. As modificações principais foram duas:

- Modificar o protocolo para utilizar a DFS, ao invés da BFS;
- Modificar o protocolo para enviar os pacotes por dois caminhos simultaneamente;

Qualquer outra modificação, que não foi citada, foi realizada para alcançar estas duas modificações principais.

Metodologia

Modificação do Algoritmo Original

Algoritmo 1: AOMDV Original Simplificado

Variáveis: pacote, rreq, rep, origem, destino, caminho_reverso(vetor n),

tabela_rotas(matriz $n \times n$), nó_atual

Objetivo: Envio de múltiplos caminhos para pacotes RREQ e resposta RREP

Inicializar tabela de rotas e vizinhos

Função: EnviaRequest(*rreq*, *destino*, *caminho_reverso*):

rreq.salto \leftarrow *ip_broadcast*

caminho_reverso.valor \leftarrow *nó_atual*

envia o pacote

Fim da função

Função: EnviaReply(*rrep*, *origem*, *caminho_reverso*):

rrep.salto \leftarrow *caminho_reverso*[1].*valor*

envia pacote

Fim da função

Função: Reverte(*caminho_reverso*):

Inverte a ordem dos elementos do vetor *caminho_reverso*

Fim da função

Função: Processar RREQ(*origem*, *destino*)

if $n = \textit{destino}$ then

| EnviaReply(RREP, nó)

else

| EnviaRequest(RREQ, destino)

end

Fim da função

Função: Processar RREP(*origem*, *destino*, *tabela*):

if *rota recebida é viável* then

| *tabela_rotas*[1][] \leftarrow *inverte*(*caminho_reverso*)

end

Fim da função

Função: EnviaPacote(*pacote*, *tabela_rotas*):

pacote.salto \leftarrow *tabela_rotas*[1][].*prximo*

envia o pacote

Fim da função

Algoritmo original

Metodologia

Modificação do Algoritmo Original

Algoritmo 2: AOMDV Modificado Simplificado

Variáveis: pacote, rreq, rrep, hello, origem, destino, cópia_pacote, nó_atual, caminho_reverso(vetor n), pilha_vizinhos(vetor n), tabela_rotas(matriz $n \times n$)

Objetivo: Envio de múltiplos caminhos para pacotes RREQ e resposta RREP

Inicializar tabela de rotas e vizinhos

Função: EnviaRequest(*rreq*, *destino*, *caminho_reverso*, *pilha_vizinhos*[]):

rreq.salto \leftarrow *pilha_vizinhos*[]
caminho_reverso.valor \leftarrow *nó_atual*
envia o pacote

Fim da função

Função: EnviaReply(*rrep*, *origem*, *caminho_reverso*):

rrep.salto \leftarrow *caminho_reverso*[1].valor
envia pacote

Fim da função

Função: Reverte(*caminho_reverso*):

Inverte a ordem dos elementos do vetor *caminho_reverso*

Fim da função

Função: Processar RREQ(*origem*, *destino*)

if $n = \textit{destino}$ then
| EnviaReply(RREP, nó)
else
| EnviaRequest(RREQ, destino)
end
Fim da função

Função: Processar RREP(*origem*, *destino*, *tabela*):

if *rota recebida é viável* then
| *tabela_rotas*[1][] \leftarrow *inverte(caminho_reverso)*
end

if *nó* = *origem* then
| EnviaPacote(pacote, *tabela_rotas*)
end

Fim da função

Função: Processar Hello(*hello*, *pilha_vizinhos*)

pilha_vizinhos[] \leftarrow *hello.id_origem*

Fim da função

Função: EnviaPacote(*pacote*, *tabela_rotas*):

cópia_pacote \leftarrow *pacote*
pacote.salto \leftarrow *tabela_rotas*[1][].próximo
cópia_pacote.salto \leftarrow *tabela_rotas*[2][].próximo
envia os pacotes

Fim da função

Algoritmo modificado

Metodologia

Preparação do Ambiente de Experimentação

O ambiente de experimentação foi dividido em dois:

- Um ambiente de controle;
- Um ambiente de testes;

No ambiente de controle, a rede foi configurada sem um nó atacante, enquanto que no ambiente de testes, a rede estava sob ataque. Os ambientes também tinham quantidades variáveis de nós, para testar o funcionamento dos algoritmos sob diferentes condições e tamanhos de rede.

Metodologia

Preparação do Ambiente de Experimentação

Os ambientes foram preparados para a execução no simulador, portanto, devem ser escritos em linguagem TCL, linguagem aceita pelo NS 2 para a execução das simulações.

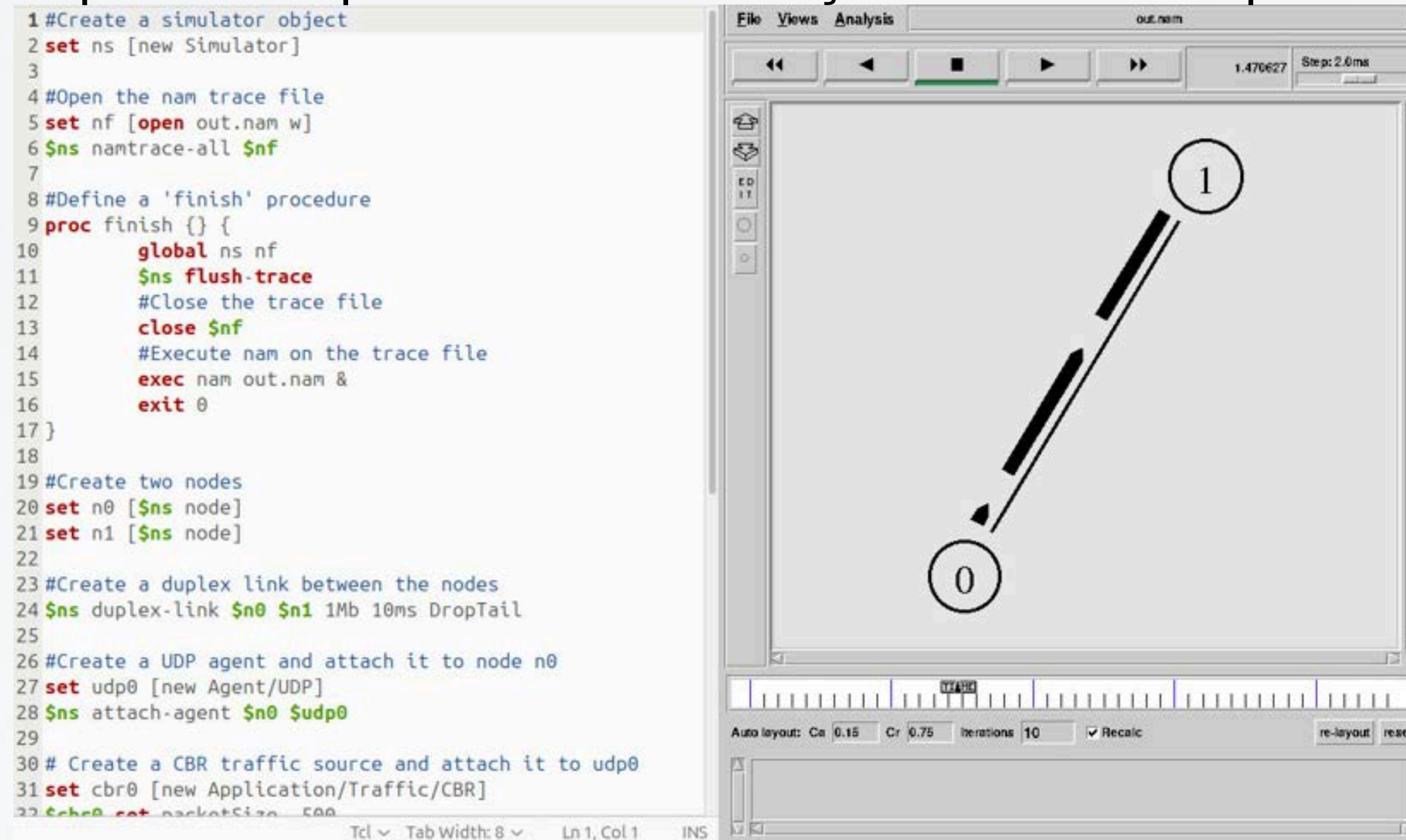
Com esta linguagem, é possível configurar praticamente tudo da rede manualmente.

Para auxiliar, foi utilizado o Network Animator (NAM), um software capaz de produzir animações da rede simulada.

Metodologia

Realização dos Experimentos

Exemplo de arquivo TCL e animação NAM correspondente



Fonte: Autor

Metodologia

Parâmetros de Medição

Os parâmetros de medição que foram considerados relevantes para este estudo são:

- Taxa de entrega de pacotes (PDR) de pedido de rota e de dados;
- Latência na entrega dos pacotes de dados;
- Tempo total de execução;
- Consumo de energia médio;

Além disso, especificamente para as redes sob ataque, foi considerada a efetividade do nó malicioso para descartar os pacotes de dados.

Metodologia

Parâmetros de Medição

Os parâmetros foram medidos a partir dos arquivos de rastreamento (*trace files*), gerados pelo próprio simulador. Estes arquivos são bem confusos em sua forma bruta, portanto, foram analisados por um programa escrito em linguagem AWK, que lia as informações dos arquivos de rastreamento e processavam-nas para melhor entendimento.

Metodologia

Parâmetros de Medição - Arquivo de Rastreamento

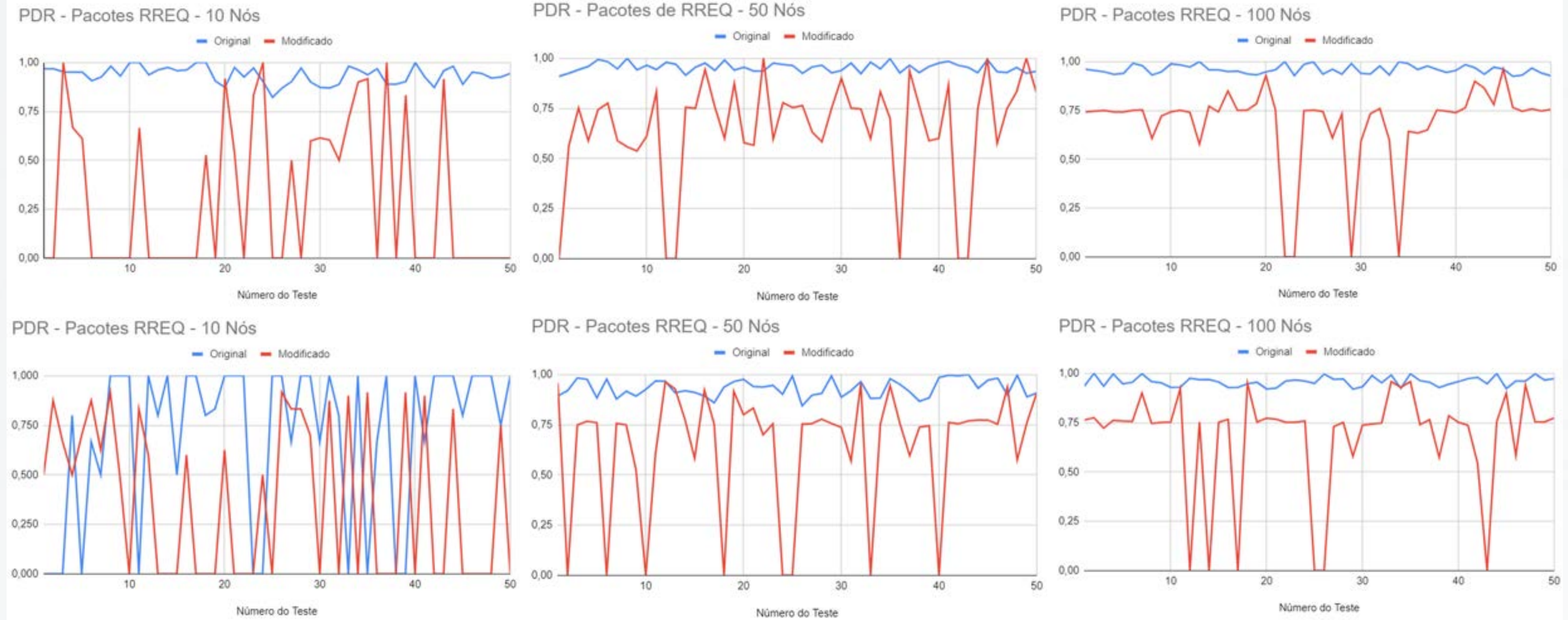
```
1 s 0.500000000 _4_ AGT --- 0 cbr 512 [0 0 0 0] [energy 100.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [4:0 2:0 32 0] [0] 0 0
2 r 0.500000000 _4_ RTR --- 0 cbr 512 [0 0 0 0] [energy 100.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [4:0 2:0 32 0] [0] 0 0
3 s 0.500000000 _4_ RTR --- 0 AOMDV 52 [0 0 0 0] [energy 100.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
4 s 0.500335000 _4_ MAC --- 0 AOMDV 110 [0 ffffffff 4 800] [energy 100.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
5 N -t 0.500335 -n 0 -e 99.949702
6 N -t 0.500335 -n 5 -e 99.949702
7 N -t 0.500335 -n 8 -e 99.949702
8 N -t 0.500335 -n 9 -e 99.949702
9 N -t 0.500335 -n 1 -e 99.949702
10 N -t 0.500335 -n 3 -e 99.949702
11 N -t 0.500335 -n 7 -e 99.949702
12 N -t 0.500336 -n 6 -e 99.949702
13 N -t 0.500336 -n 2 -e 99.949702
14 r 0.501215136 _0_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
15 r 0.501215279 _5_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
16 r 0.501215303 _8_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
17 r 0.501215335 _9_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
18 r 0.501215343 _1_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
19 r 0.501215483 _3_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
20 r 0.501215487 _7_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
21 r 0.501215579 _6_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
22 r 0.501215669 _2_ MAC --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
23 r 0.501240136 _0_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
24 r 0.501240279 _5_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
25 r 0.501240303 _8_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
26 r 0.501240335 _9_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
27 r 0.501240343 _1_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
28 r 0.501240483 _3_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
29 r 0.501240487 _7_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
30 r 0.501240579 _6_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
31 r 0.501240669 _2_ RTR --- 0 AOMDV 52 [0 ffffffff 4 800] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [4:255 -1:255 30 0] [0x2 0 1 [2 0] [4 4]] (REQUEST)
32 s 0.501240669 _2_ RTR --- 0 AOMDV 52 [0 0 0 0] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [2:255 4:255 30 4] [0x4 0 [2 2] 10.000000] (REPLY) [1 4]
33 s 0.501615669 _2_ MAC --- 0 ARP 86 [0 ffffffff 2 806] [energy 99.949702 ei 0.050 es 0.000 et 0.000 er 0.000] ----- [REQUEST 2/2 0/4]
34 N -t 0.501616 -n 6 -e 99.949456
35 N -t 0.501616 -n 7 -e 99.949456
36 N -t 0.501616 -n 9 -e 99.949456
37 N -t 0.501616 -n 3 -e 99.949456
38 N -t 0.501616 -n 0 -e 99.949456
39 N -t 0.501616 -n 8 -e 99.949456
```

Fonte: Autor

Experimentação e Resultados

PDR de Pacotes RREQ

Os gráficos de cima referem-se ao ambiente de controle, os de baixo, aos de teste

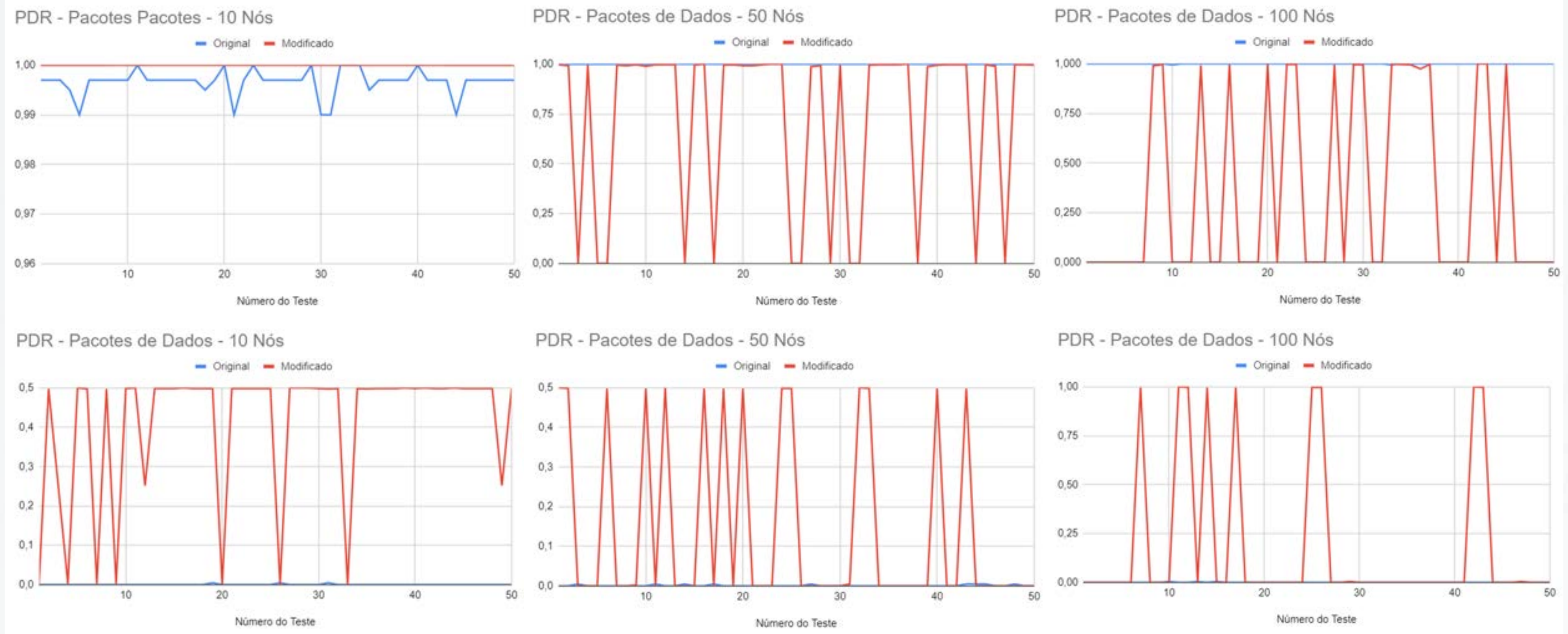


Fonte: Autor

Experimentação e Resultados

PDR de Pacotes Dados

Os gráficos de cima referem-se ao ambiente de controle, os de baixo, aos de teste

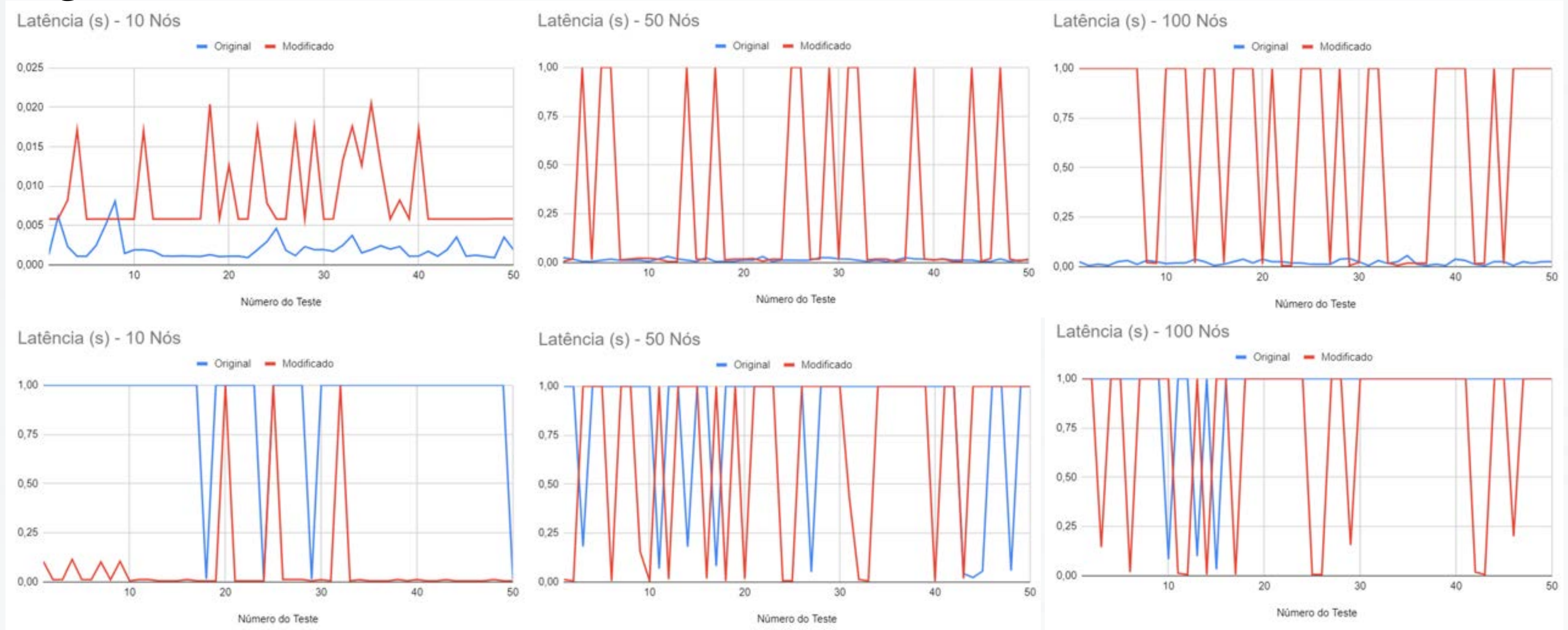


Fonte: Autor

Experimentação e Resultados

Latência na Entrega dos Pacotes de Dados

Os gráficos de cima referem-se ao ambiente de controle, os de baixo, aos de teste

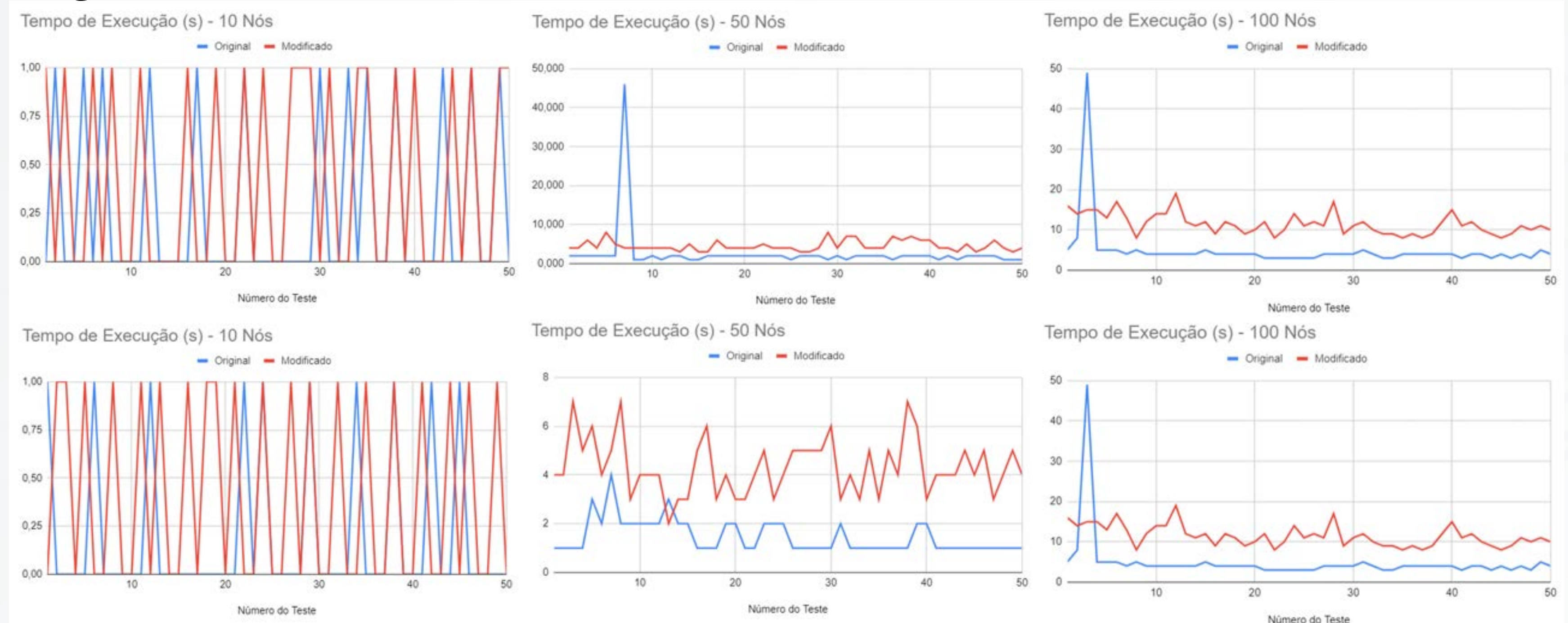


Fonte: Autor

Experimentação e Resultados

Tempo Total de Execução

Os gráficos de cima referem-se ao ambiente de controle, os de baixo, aos de teste

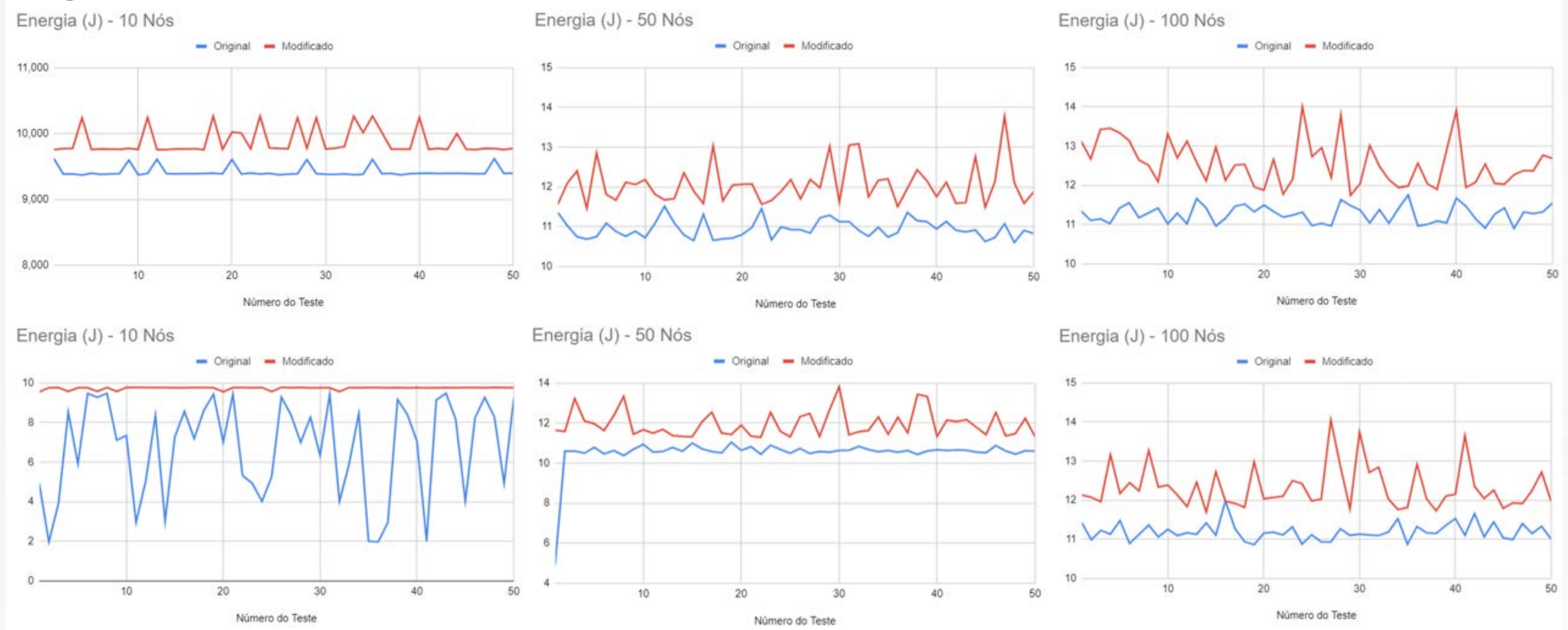


Fonte: Autor

Experimentação e Resultados

Consumo Médio de Energia por Nó

Os gráficos de cima referem-se ao ambiente de controle, os de baixo, aos de teste



Fonte: Autor

Experimentação e Resultados

Eficiência do Nó Malicioso



Fonte: Autor

Experimentação e Resultados

Análise dos Resultados

Os resultados indicados, além de outros, indicados no trabalho, mostram que, de maneira geral, o programa modificado se saiu pior que o algoritmo original, mostrando que existem pontos a serem melhorados nas modificações.

Entretanto, foi possível notar, nos experimentos da eficiência do nó malicioso, que o algoritmo modificado conseguiu entregar os pacotes de dados, mesmo que sua eficiência tenha diminuindo, mostrando certa promessa na ideia.

Conclusão

Conclusão Final

A ideia para o estudo é promissora, especialmente nos experimentos com poucos nós. Entretanto a implementação das modificações foi bem prototípica, tendo em vista os erros comentados e a forma simples em como o algoritmo foi mudado.

Portanto, conclui-se que o objetivo final foi atingido em partes, pois a ideia foi, mesmo que pouco, explorada, embora a execução dos testes necessite de mais refinamento, como modificações mais bem ajustadas e testes mais robustos.

Conclusão

Trabalhos Futuros

Para o futuro, é válido dizer que a ideia pode ser mais bem explorada, com modificações mais refinadas, ou, melhor ainda, um código construído com este objetivo em mente.

Além disso, é necessária maior robustez experimental, com os testes apresentando mais nós, mais conexões e mais atacantes, o que, tendo em vista as características das MANETs, é uma situação realista.

Referências


ABDEL-FATTAH, F.; FARHAN, K. A.; AL-TARAWNEH, F. H.; ALTAMIMI, F. Security challenges and attacks in dynamic mobile ad hoc networks manets. IEEE, p. 28–33, 2019.

MIRZA, S.; LÓPEZ BAKSHI, S. Z. introduction to manet. International Research Journal of Engineering and Technology (IRJET), IRJET, v. 5, n. 1, p. 17–20, 2018.

HAMILTON, W. L. Graph representation learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool, v. 14, n. 3, p. 1–159, 2020.



Alguma dúvida?



Sessão de Agradecimentos