REDES - Entregavel 2 - Grupo 4

Relatório de experimento: Contagem ao infinito em Vetor de distância

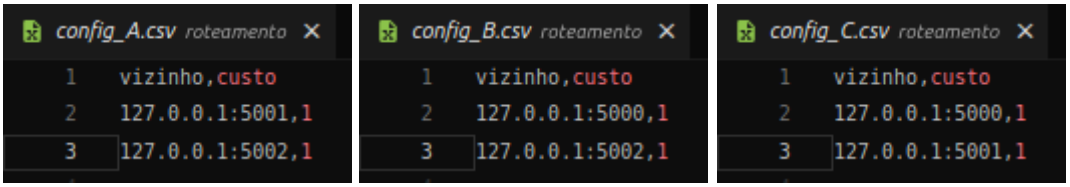
Metodologia

Configuração da Topologia

Foi implementada uma topologia triangular com três roteadores interconectados:

- **Router A**: Endereço 127.0.0.1:5000, rede rede_A
- **Router B**: Endereço 127.0.0.1:5001, rede rede_B
- **Router C**: Endereço 127.0.0.1:5002, rede rede_C

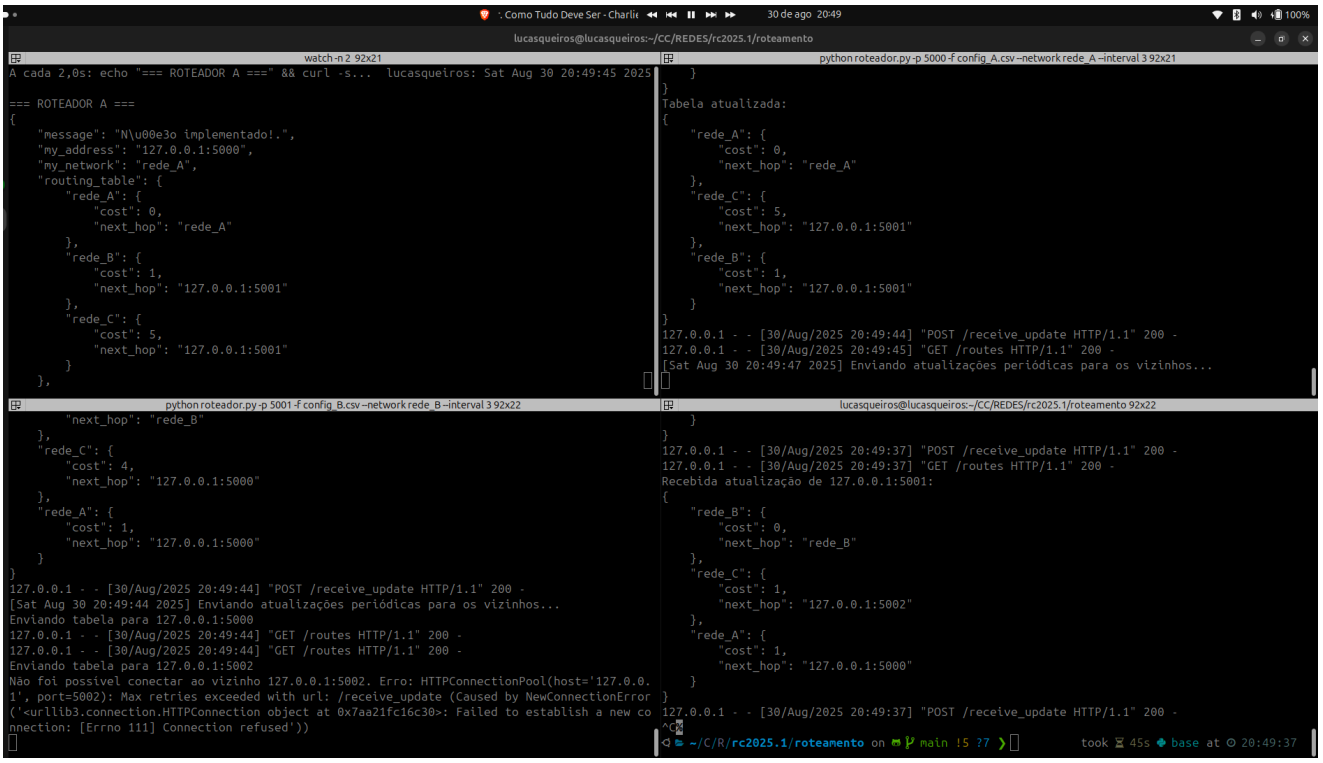
Todos os enlaces foram configurados com custo 1, formando uma malha triangular simétrica onde cada roteador possui conectividade direta com os outros dois.



Execução do Experimento

O experimento foi dividido em duas fases:

1. **Fase de Convergência**: Todos os três roteadores foram inicializados e permitiu-se que o protocolo de vetor de distância estabilizasse
2. **Fase de Falha**: O roteador C foi interrompido abruptamente para simular uma falha, observando-se o comportamento dos roteadores restantes



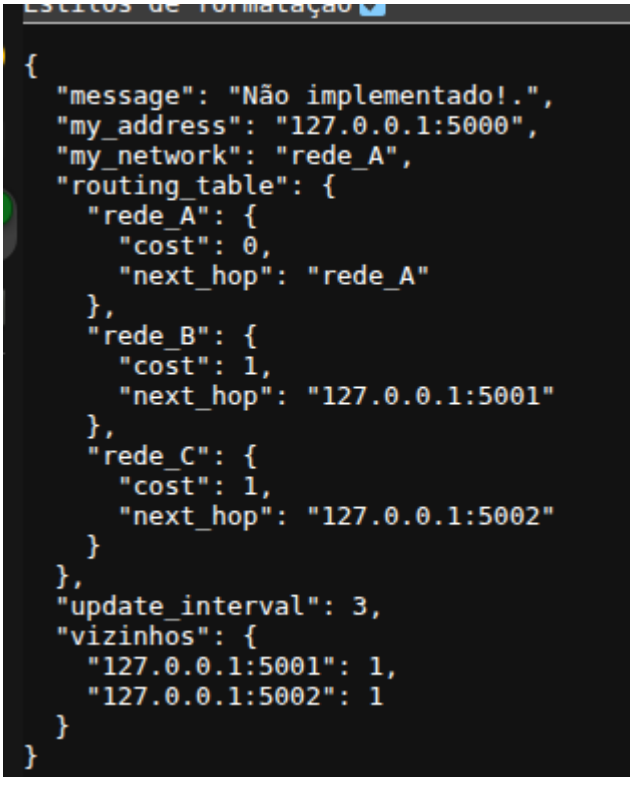
📊 RESULTADOS OBTIDOS

Estado Pré-Interrupção

Antes da falha, todas as tabelas de roteamento estavam corretamente convergidas:

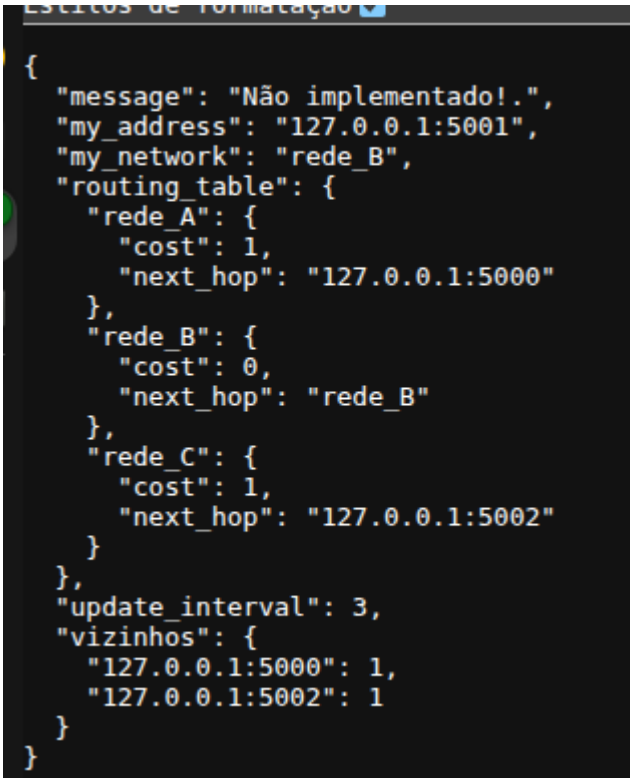
Router A:

- rede_B: custo 1 via 127.0.0.1:5001
- rede_C: custo 1 via 127.0.0.1:5002



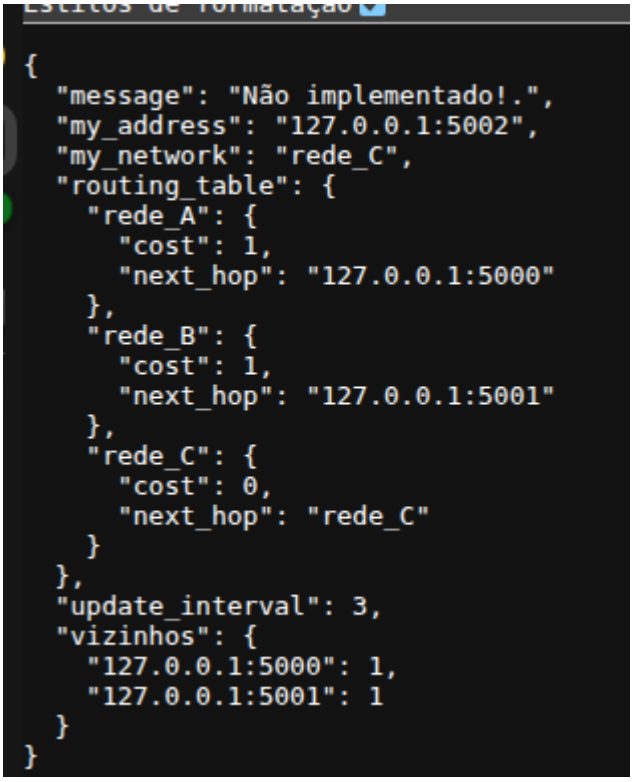
Router B:

- rede_A: custo 1 via 127.0.0.1:5000
- rede_C: custo 1 via 127.0.0.1:5002



Router C:

- rede_A: custo 1 via 127.0.0.1:5000
- rede_B: custo 1 via 127.0.0.1:5001



Estado Pós-Interrupção

Após a falha do roteador C, observou-se:

Router A:

- rede_C: custo 9 via 127.0.0.1:5001 (aumentou de 1 para 9) no momento da captura de tela e continuou aumentando conforme pode ser visto no arquivo de análise do wireshark `pcap`


```
{
  "message": "Não implementado!.",
  "my_address": "127.0.0.1:5000",
  "my_network": "rede_A",
  "routing_table": {
    "rede_A": {
      "cost": 0,
      "next_hop": "rede_A"
    },
    "rede_B": {
      "cost": 1,
      "next_hop": "127.0.0.1:5001"
    },
    "rede_C": {
      "cost": 9,
      "next_hop": "127.0.0.1:5001"
    }
  },
  "update_interval": 3,
  "vizinhos": {
    "127.0.0.1:5001": 1,
    "127.0.0.1:5002": 1
  }
}
```


Router B:

- rede_C: custo 24 via 127.0.0.1:5001 (aumentou de 1 para 24) nno momento da captura de tela e continuou aumentando conforme pode ser visto no arquivo de análise do wireshark `pcap`


```
{
  "message": "Não implementado!.",
  "my_address": "127.0.0.1:5001",
  "my_network": "rede_B",
  "routing_table": {
    "rede_A": {
      "cost": 1,
      "next_hop": "127.0.0.1:5000"
    },
    "rede_B": {
      "cost": 0,
      "next_hop": "rede_B"
    },
    "rede_C": {
      "cost": 24,
      "next_hop": "127.0.0.1:5000"
    }
  },
  "update_interval": 3,
  "vizinhos": {
    "127.0.0.1:5000": 1,
    "127.0.0.1:5002": 1
  }
}
```


Análise dos objetivos

1. Comportamento quando o destino é inacessível

Quando o roteador C tornou-se inacessível, os roteadores A e B continuaram anunciando rotas para rede_C baseadas em informações obsoletas. O roteador A passou a utilizar o caminho via roteador B para alcançar rede_C, incrementando progressivamente a métrica de custo.

2. Ciclos de Atualização e Reconhecimento de Falha

O experimento demonstrou que o protocolo **não reconhece automaticamente** o destino perdido. Mesmo após múltiplos ciclos de atualização (evidenciado pelo custo 9 e 24 para rede_C), a rota não foi removida da tabela, caracterizando o problema de contagem ao infinito.

3. Mecanismos de Proteção

O protocolo implementado **não possui mecanismos** para evitar crescimento indefinido das métricas. Não foi observado:

- Limite máximo de hop count
- Detecção automática de rotas inválidas
- Remoção de rotas inacessíveis

Explicação técnica do problema

Por que a Contagem ao Infinito Ocorre?

O problema de contagem ao infinito no algoritmo de vetor de distância ocorre devido à **natureza distribuída e à dependência de informação indireta**:

1. **Más Notícias Viajam Devagar**: Quando o roteador C falha, o roteador B inicialmente não detecta a falha imediatamente
2. **Dependência Circular**: O roteador A acredita que B pode alcançar C, e B acredita que A pode alcançar C
3. **Incremento Progressivo**: Cada roteador incrementa a métrica recebida do outro, criando um loop de aumento infinito
4. **Falta de Visão Global**: Como cada roteador toma decisões baseadas apenas em informações locais, não percebe o loop de roteamento formado

Técnicas de mitigação

Split Horizon

Impede que um roteador anuncie uma rota de volta para o vizinho do qual aprendeu essa rota. No experimento:

- **Como evitaria**: Se implementado, o roteador B não anunciaria para A que pode chegar em C (já que aprendeu essa rota originalmente de A)

Route Poisoning (Envenenamento de Rota)

Quando uma rota torna-se inacessível, o roteador anuncia essa rota com métrica infinita imediatamente. No experimento:

- **Como evitaria**: Ao detectar a falha de C, o roteador B anunciaria "rede_C: custo ∞" para A, quebrando o loop imediatamente

Hold-down Timers

Introduz um período de espera após a detecção de uma falha, durante o qual não são aceitas atualizações para aquela rota. No experimento:

- **Como evitaria**: Preveniria que o roteador A aceitasse rapidamente a rota inválida anunciada por B

Limite de Hop Count

Estabelece um valor máximo para a métrica (ex: 16 hops). No experimento:

- **Como evitaria**: Quando o custo atingisse o limite máximo (ex: 16), a rota seria considerada inalcançável e removida

Avaliação de impacto na convergencia

Eficácia das Técnicas

- **Split Horizon + Poison Reverse**: Reduziriam o tempo de convergência de minutos para segundos
- **Hold-down Timers**: Adicionariam pequeno atraso inicial mas preveniram loops transitórios
- **Hop Count Limit**: Estabeleceria um limite máximo para a divergência

Melhoria Esperada

Com as técnicas de mitigação implementadas, o tempo de convergência após uma falha seria reduzido drasticamente:

- **Sem mitigação**: Convergência nunca ocorre (contagem infinita)
- **Com mitigação**: Convergência em 2-3 ciclos de atualização (6-9 segundos)

Conclusão

O experimento demonstrou claramente a vulnerabilidade inerente ao algoritmo de vetor de distância frente a falhas de roteadores. A contagem ao infinito observada (custo 9 para rede_C) comprova a necessidade crítica de mecanismos de proteção em implementações reais de protocolos de roteamento.

As técnicas de mitigação discutidas - particularmente a combinação de Split Horizon com Route Poisoning - mostram-se essenciais para garantir convergência rápida e estável em redes dinâmicas, prevenindo os loops de roteamento que caracterizam o problema de contagem ao infinito.

Anexo: Arquivo de captura Wireshark (experimento.pcap) contendo a troca de pacotes durante o experimento.