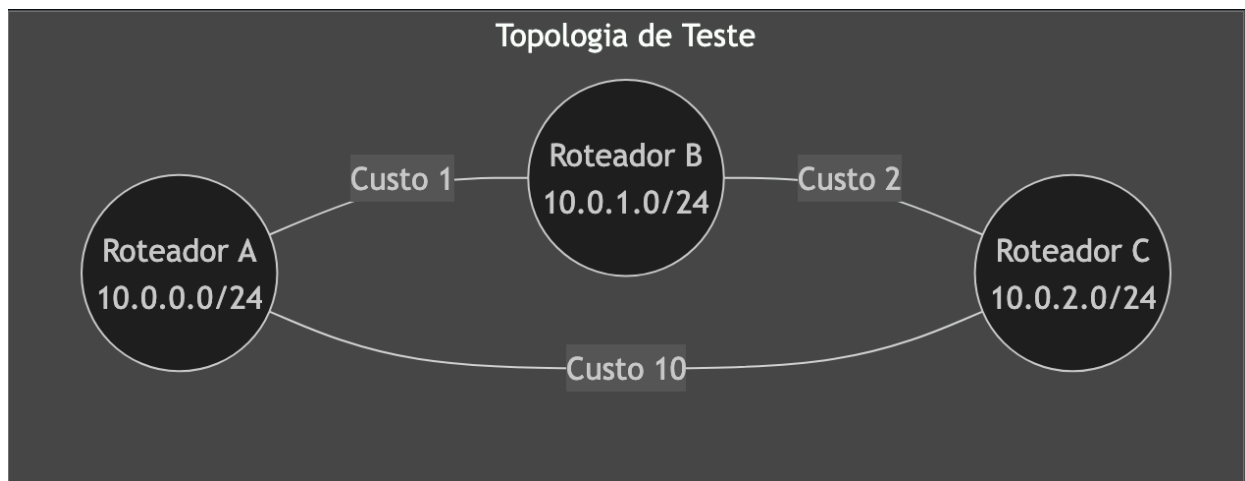


Topologia

A seguir temos a topologia de teste utilizada.

Para análise, iremos utilizar a tabela de roteamento do Roteador A. Podemos perceber que o custo direto ao Roteador C é 10. Porém há um custo menor, passando pelo Roteador B.

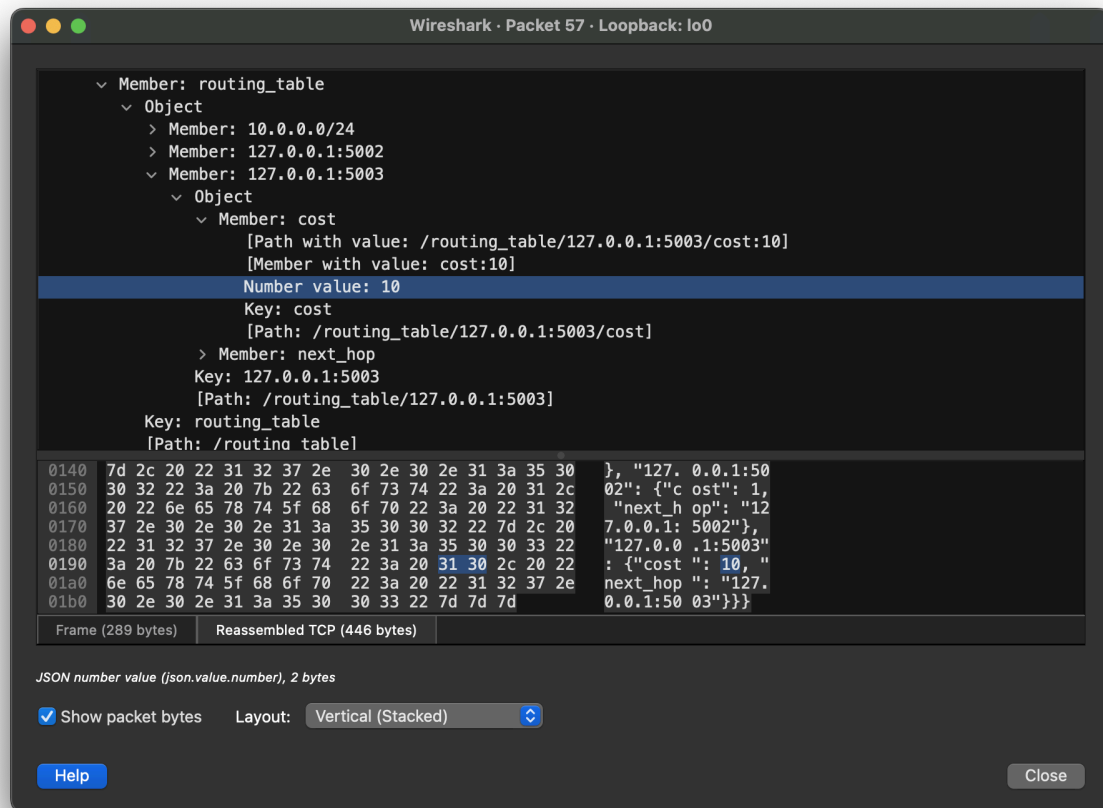


Endereços:

- **Roteador A:** 127.0.0.1:5001
- **Roteador B:** 127.0.0.1:5002
- **Roteador C:** 127.0.0.1:5003

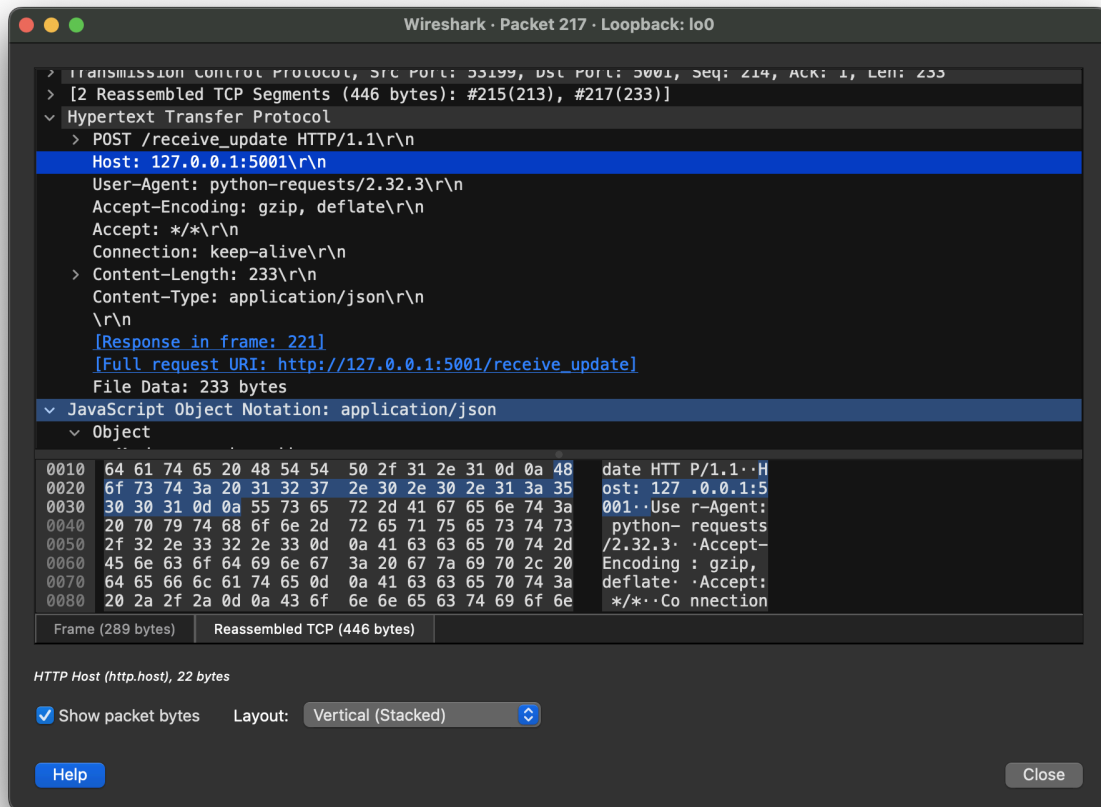
O objetivo dessa análise é mostrar como o algoritmo converge encontrando a melhor rota do Roteador A para o Roteador C.

Estado inicial



Na figura acima, temos o valor do custo para o Roteador C de 10.

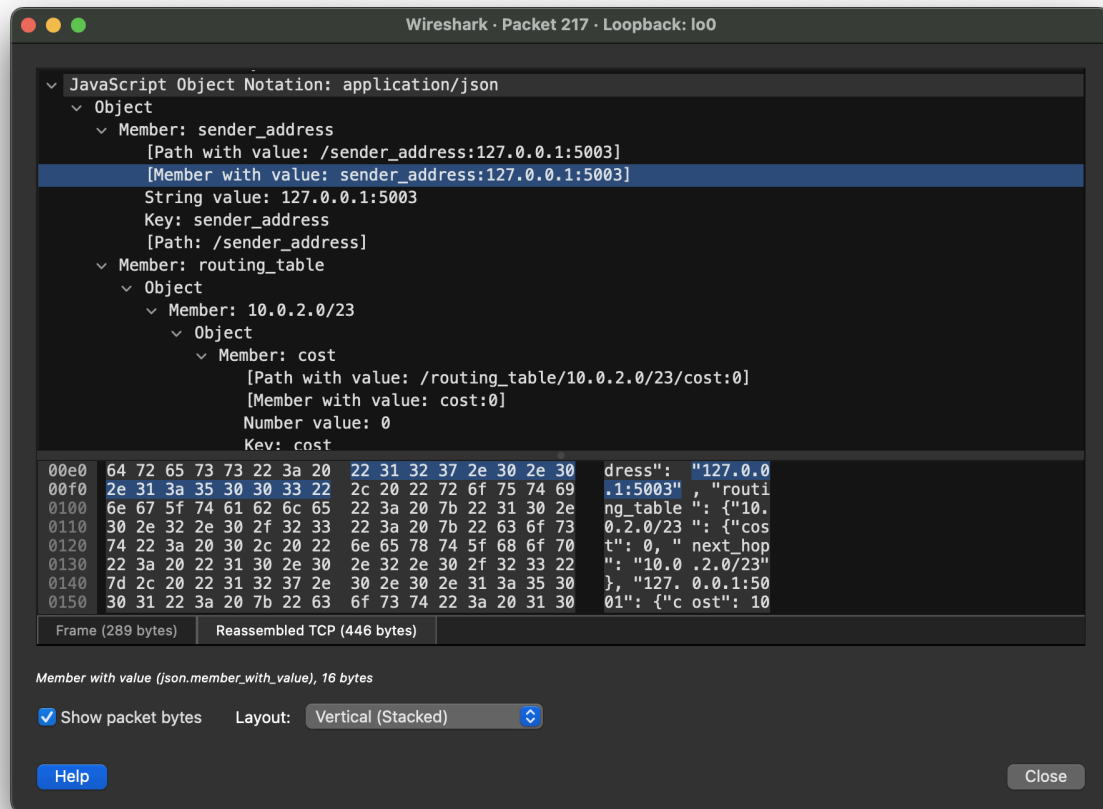
Mensagem de Atualização



Na figura acima identificamos o Host 127.0.0.1:5001 ou seja, o **Roteador A**.

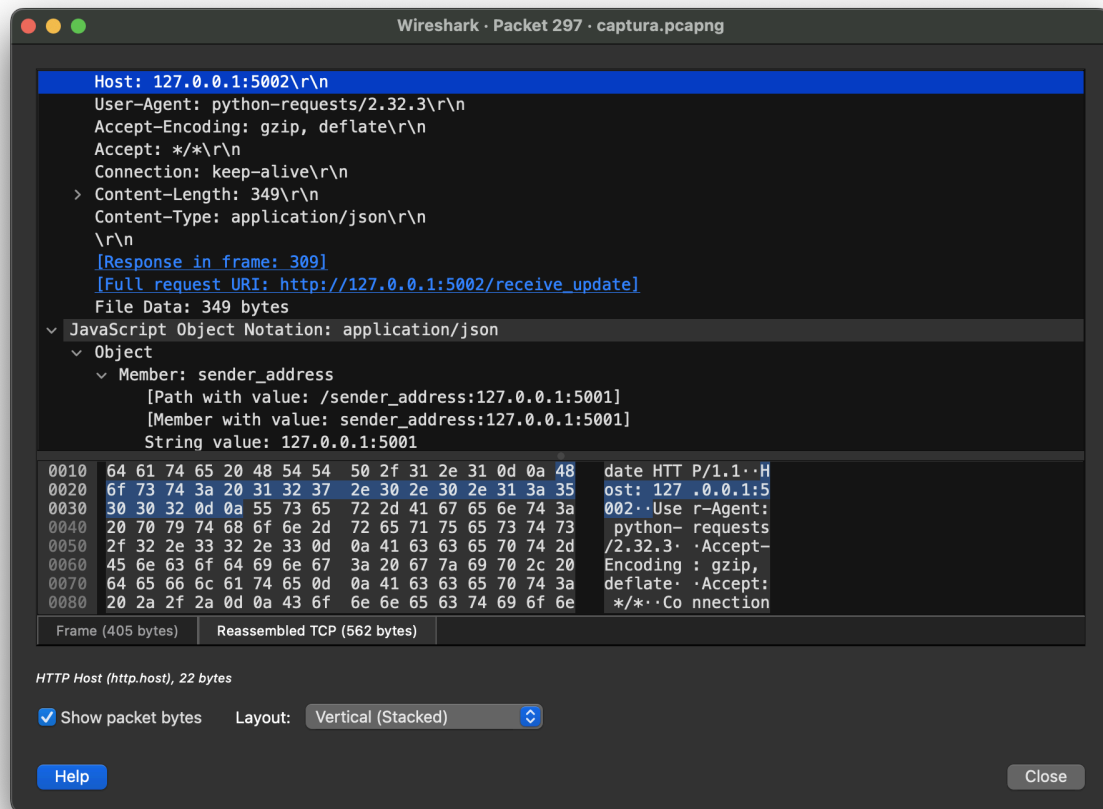
Para esse roteador, temos uma requisição **POST /receive_update** feita para ele. Ou seja, algum dos outros roteadores mandou sua tabela para o Roteador A.

A seguir, temos que o Roteador C enviou sua tabela de roteamento para o Roteador A.



Estado final

Depois do algoritmo Distance Vector convergir, temos o seguinte cenário. Na próxima figura temos o **/receive_update** para o Roteador B (primeira linha) vindo do Roteador A (linha de /sender_address).



Na próxima figura temos a tabela de roteamento do Roteador A. Vemos que nela, o custo para o Roteador C é 3, além de que seu next_hop é o Roteador B, como esperávamos.

Wireshark · Packet 297 · captura.pcapng

Member: 127.0.0.1:5003

Object

Member: cost

[Path with value: /routing_table/127.0.0.1:5003/cost:3]
[Member with value: cost:3]
Number value: 3
Key: cost
[Path: /routing_table/127.0.0.1:5003/cost]

Member: next_hop

[Path with value: /routing_table/127.0.0.1:5003/next_hop:127.0.0.1:5002]
[Member with value: next_hop:127.0.0.1:5002]
String value: 127.0.0.1:5002
Key: next_hop
[Path: /routing_table/127.0.0.1:5003/next_hop]

Key: 127.0.0.1:5003
[Path: /routing_table/127.0.0.1:5003]

Member: 10.0.1.0/24

0180	22 31 32 37 2e 30 2e 30 2e 31 3a 35 30 30 33 22	"127.0.0.1:5003"
0190	3a 20 7b 22 63 6f 73 74 22 3a 20 33 2c 20 22 6e	: {"cost ": 3, "n
01a0	65 78 74 5f 68 6f 70 22 3a 20 22 31 32 37 2e 30	ext_hop" : "127.0
01b0	2e 30 2e 31 3a 35 30 30 32 22 7d 2c 20 22 31 30	.0.1:500 2"}, "10
01c0	2e 30 2e 31 2e 30 2f 32 34 22 3a 20 7b 22 63 6f	.0.1.0/2 4": {"co
01d0	73 74 22 3a 20 31 2c 20 22 6e 65 78 74 5f 68 6f	st": 1, "next_ho
01e0	70 22 3a 20 22 31 32 37 2e 30 2e 30 2e 31 3a 35	p": "127 .0.0.1:5
01f0	30 30 32 22 7d 2c 20 22 31 30 2e 30 2e 32 2e 30	002"}, " 10.0.2.0

Frame (405 bytes) Reassembled TCP (562 bytes)

JSON object member (json.member), 59 bytes

☒ Show packet bytes

Layout: Vertical (Stacked)

Help

Close