

Relatório Trabalho 1 - Laboratório de redes de computadores

Deivid Santos, Henrique Andreatta, João Vitor Pioner

1. Topologia

Foi escolhida uma topologia com 16 endereços de sub-rede, todos foram utilizados, ficando com CIDR 130.10.0.0/20. Existe um total de 6 roteadores, 3 hubs, 5 switches 14 hosts e um ponto de conexão simulando uma conexão externa com a internet.

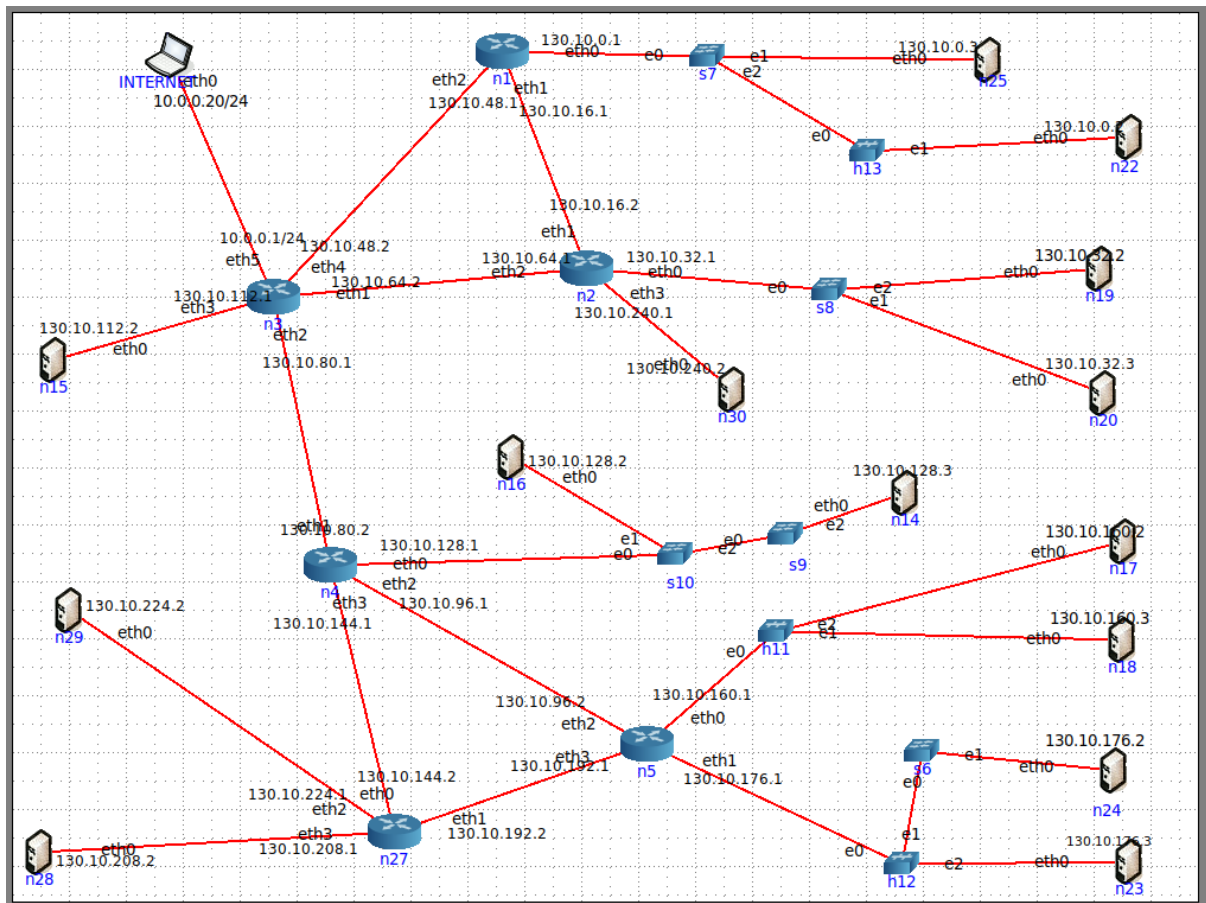


Figura 1 - Topologia

2. Ping

2.1. Nodo 25 -> Nodo 23

Na sequência de comandos a seguir nas figuras 2 e 3 é possível notar que a tabela arp está vazia no resultado do primeiro comando, depois é executado o comando de ping do nodo 25 para o nodo 23 e funciona corretamente, e então verificando a tabela arp novamente notamos que agora a tabela arp está preenchida com o endereço dos roteadores conectados em cada nodo.

```
vcmd (on corevm)
root@n25:/tmp/pycore.43113/n25.conf# ip neigh show dev eth0
root@n25:/tmp/pycore.43113/n25.conf# ping -c 3 130.10.176.3
PING 130.10.176.3 (130.10.176.3) 56(84) bytes of data.
64 bytes from 130.10.176.3: icmp_seq=1 ttl=60 time=0.647 ms
64 bytes from 130.10.176.3: icmp_seq=2 ttl=60 time=0.237 ms
64 bytes from 130.10.176.3: icmp_seq=3 ttl=60 time=0.347 ms

--- 130.10.176.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.237/0.410/0.647/0.173 ms
root@n25:/tmp/pycore.43113/n25.conf# ip neigh show dev eth0
130.10.0.1 lladdr 00:00:00:aa:00:0c REACHABLE
root@n25:/tmp/pycore.43113/n25.conf#
```

Figura 2 - Nodo origem do ping

```
vcmd (on corevm)
root@n23:/tmp/pycore.43113/n23.conf# ip neigh show dev eth0
root@n23:/tmp/pycore.43113/n23.conf# ip neigh show dev eth0
130.10.176.1 lladdr 00:00:00:aa:00:09 REACHABLE
root@n23:/tmp/pycore.43113/n23.conf#
```

Figura 3 - nodo destino do ping

Nas figuras 4 e 5 estão os pacotes ARP e ICMP transferidos durante a execução do ping anterior, como a tabela desses nodos está vazia, os dois executam um broadcast para descobrir qual é o endereço MAC da máquina que o pacote será enviado, como o default do nodo 25 é o ip 130.10.0.1 ele tenta descobrir qual o Mac responsável por esse IP.

Depois são executados os comandos de envio do ping com o ICMP echo request e reply, as últimas linhas representam o retorno do ping, onde o roteador está tentando descobrir qual o Mac do IP 130.10.0.3, que é o nodo atual.

O cenário se repete parecido no nodo 23, porém com a ordem ao contrário, primeiro o roteador descobrindo o nodo por ARP e depois o nodo conhecendo o roteador.

The image shows a Wireshark capture window titled '*veth19.0.c1 (on corevm)'. The packet list on the left shows a series of ICMP and ARP packets. The selected packet (No. 33) is an ARP request from 00:00:00:aa:00:0d to 00:00:00:aa:00:0c. The packet details pane shows the Ethernet II header, the destination and source MAC addresses, the ARP type, and the address resolution protocol reply. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
24	822.165029470	00:00:00:aa:00:0d	Broadcast	ARP	42	Who has 130.10.0.1? Tell 130.10.0.3
25	822.165211019	00:00:00:aa:00:0c	00:00:00:aa:00:0d	ARP	42	130.10.0.1 is at 00:00:00:aa:00:0c
26	822.165221273	130.10.0.3	130.10.176.3	ICMP	98	Echo (ping) request id=0x0034, seq=1/256, ttl=64 (reply in 2...)
27	822.165622720	130.10.176.3	130.10.0.3	ICMP	98	Echo (ping) reply id=0x0034, seq=1/256, ttl=64 (request in 2...)
28	823.168179575	130.10.0.3	130.10.176.3	ICMP	98	Echo (ping) request id=0x0034, seq=2/512, ttl=64 (reply in 2...)
29	823.168351241	130.10.176.3	130.10.0.3	ICMP	98	Echo (ping) reply id=0x0034, seq=2/512, ttl=64 (request in 2...)
30	824.194832983	130.10.0.3	130.10.176.3	ICMP	98	Echo (ping) request id=0x0034, seq=3/768, ttl=64 (reply in 3...)
31	824.195112191	130.10.176.3	130.10.0.3	ICMP	98	Echo (ping) reply id=0x0034, seq=3/768, ttl=64 (request in 3...)
32	827.394031170	00:00:00:aa:00:0d	00:00:00:aa:00:0d	ARP	42	Who has 130.10.0.3? Tell 130.10.0.1
33	827.394087631	00:00:00:aa:00:0d	00:00:00:aa:00:0c	ARP	42	130.10.0.3 is at 00:00:00:aa:00:0d

Frame 33: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth19.0.c1, id 0
Ethernet II, Src: 00:00:00:aa:00:0d (00:00:00:aa:00:0d), Dst: 00:00:00:aa:00:0c (00:00:00:aa:00:0c)
Destination: 00:00:00:aa:00:0c (00:00:00:aa:00:0c)
Source: 00:00:00:aa:00:0d (00:00:00:aa:00:0d)
Type: ARP (0x0806)
Address Resolution Protocol (reply)

0000 00 00 00 aa 00 0c 00 00 00 aa 00 0d 08 06 00 01
0010 08 00 06 04 00 02 00 00 aa 00 0d 82 0a 00 03
0020 00 00 00 aa 00 0c 82 0a 00 01

Figura 4 - Wireshark nodo origem

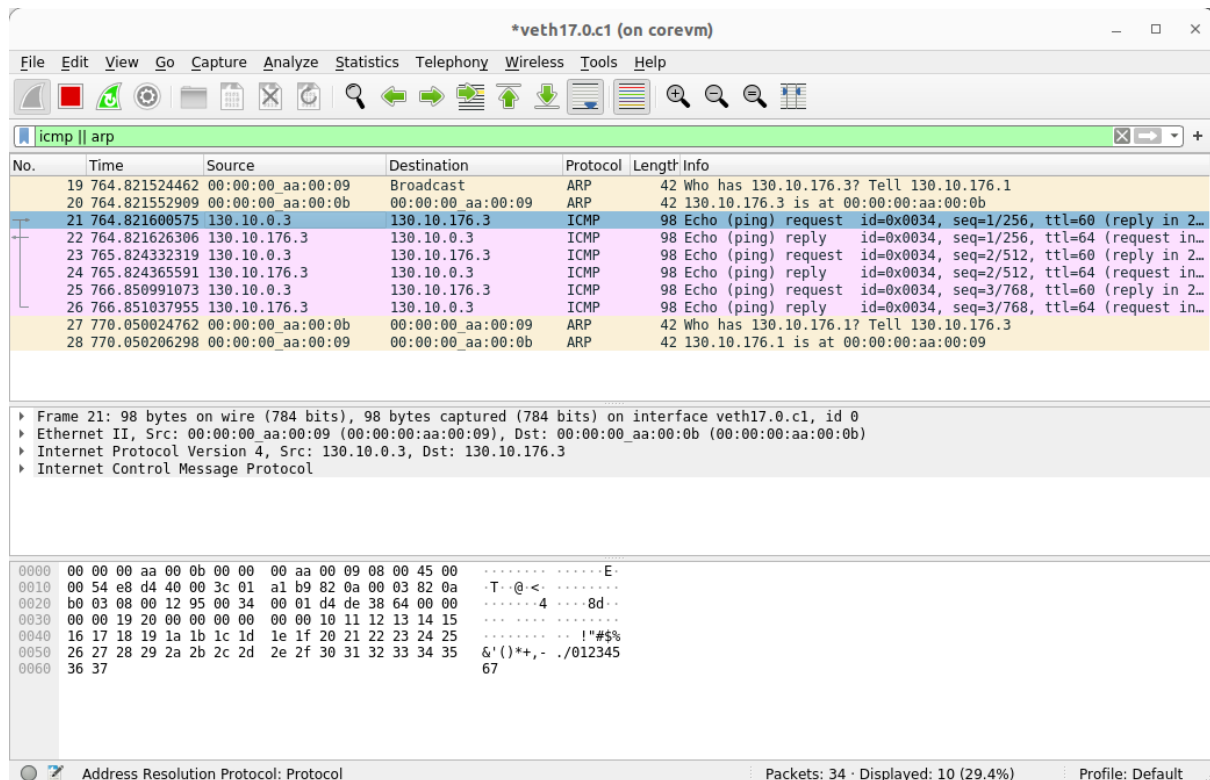


Figura 5 - Wireshark nodo destino

2.2. Nodo 15 -> Nodo 17

Para complementar o projeto, foi realizado um segundo ping do nodo 15 até o nodo 17 e o comportamento se manteve igual ao primeiro exemplo. Na imagem 6 abaixo, no lado esquerdo está o terminal e wireshark do nodo origem e na direita está o terminal e wireshark do nodo destino.

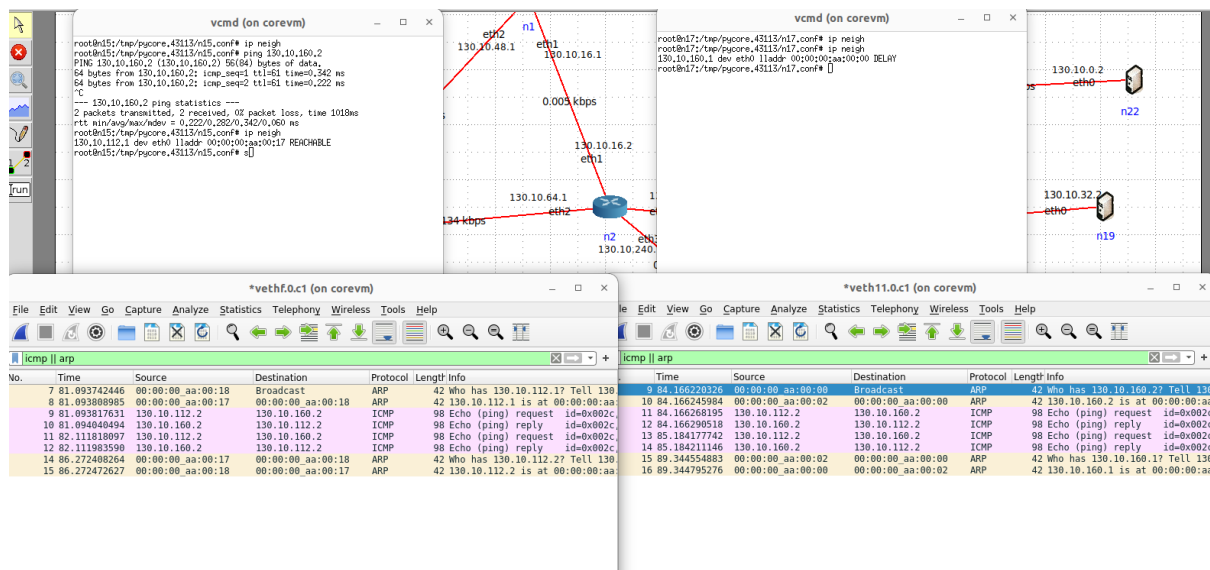


Figura 6 - Segundo ping

2.2.1. Flooding

Ao realizar o segundo ping do nodo 15 ao 17 utilizando o parâmetro de flooding conseguimos visualizar o caminho feito e notar o comportamento de um hub, que simplesmente espalha a request para todos os nodos conectados a ele, conforme é possível ver na figura 7.

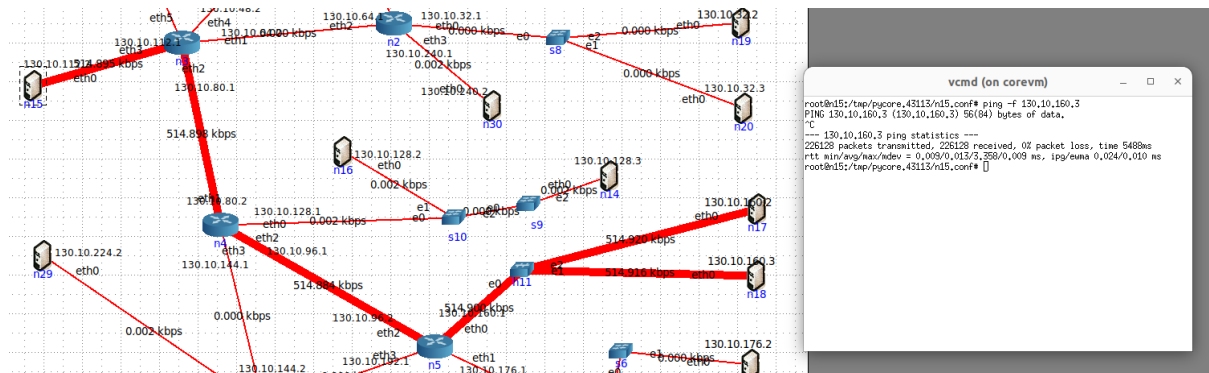


Figura 7 - Flooding

3. Traceroute

3.1. Nodo 29 -> nodo 20

Executando o comando traceroute do nodo 29 até o nodo 30 obtemos o resultado visto nas imagens 8 e 9, com 5 saltos de distância. Assim como no ping, inicialmente não temos nada na tabela arp e em seguida, após a execução do comando o roteador diretamente conectado foi adicionado a tabela arp do nodo.

```
vcmcd (on corevm)
root@n29:/tmp/pycore.43113/n29.conf# ip neigh
root@n29:/tmp/pycore.43113/n29.conf# traceroute 130.10.240.2
traceroute to 130.10.240.2 (130.10.240.2), 30 hops max, 60 byte packets
 1 130.10.224.1 (130.10.224.1) 0.255 ms 0.035 ms 0.041 ms
 2 130.10.144.1 (130.10.144.1) 0.142 ms 0.060 ms 0.197 ms
 3 130.10.80.1 (130.10.80.1) 0.217 ms 0.078 ms 0.104 ms
 4 130.10.64.1 (130.10.64.1) 0.168 ms 0.115 ms 0.060 ms
 5 130.10.240.2 (130.10.240.2) 0.136 ms 0.074 ms 0.074 ms
root@n29:/tmp/pycore.43113/n29.conf# ip neigh
130.10.224.1 dev eth0 lladdr 00:00:00:aa:00:21 REACHABLE
root@n29:/tmp/pycore.43113/n29.conf#
```

Figura 8 - Nodo origem do traceroute

```
vcmcd (on corevm)
root@n30:/tmp/pycore.43113/n30.conf# ip neigh
root@n30:/tmp/pycore.43113/n30.conf# ip neigh
130.10.240.1 dev eth0 lladdr 00:00:00:aa:00:25 DELAY
root@n30:/tmp/pycore.43113/n30.conf#
```

Figura 9 - Nodo destino do traceroute

Na figura 10 abaixo é possível notar o comportamento do traceroute em mais detalhes, onde temos vários timeouts devido ao comportamento onde é enviado primeiramente uma requisição com TTL 1 e assim incrementando a cada salto para obter a distância e todos os nodos do caminho.

Na figura 10 está destacado em azul na parte de baixo o time to live da primeira request e então obtemos um timeout, a partir disso, o TTL vai ser incrementado para 2 e uma request nova é feita para descobrir o próximo salto.

The image shows a Wireshark capture titled '*veth1d.0.c1 (on corevm)'. The filter is 'icmp || arp || udp'. The packet list shows a series of ICMP messages. The first message (Frame 8) is an ICMP Echo (ping) from 130.10.224.2 to 130.10.240.2 with TTL=1. The details pane for this packet shows 'Time to live: 1' highlighted in blue. Subsequent messages show 'Time-to-live exceeded' for TTL values 2 through 6. The packet details for the first message are expanded, showing the following fields:

- Frame 8: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface veth1d.0.c1, id 0
- Ethernet II, Src: 00:00:00:aa:00:22 (00:00:00:aa:00:22), Dst: 00:00:00:aa:00:21 (00:00:00:aa:00:21)
- Internet Protocol Version 4, Src: 130.10.224.2, Dst: 130.10.240.2
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
 - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 60
 - Identification: 0x26e2 (9954)
 - Flags: 0x0000
 - Fragment offset: 0
 - Time to live: 1
 - Protocol: UDP (17)
 - Header checksum: 0xbeb5 [validation disabled]
 - [Header checksum status: Unverified]

The packet bytes pane shows the raw data of the packet, with the first 20 bytes (header) highlighted in blue.

Figura 10 - Wireshark do nodo origem do traceroute

Na figura 11 abaixo está o TTL final da requisição, após os incrementos de cada tentativa de obter o resultado final, o ttl 6 chegou ao destino buscado e o comando foi finalizado.

27	25.830785738	130.10.64.1	130.10.224.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transi
28	25.830864627	130.10.224.2	130.10.240.2	UDP	74 47679 → 33445 Len=32
29	25.830913580	130.10.64.1	130.10.224.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transi
30	25.830970027	130.10.224.2	130.10.240.2	UDP	74 46924 → 33446 Len=32
31	25.831098416	130.10.240.2	130.10.224.2	ICMP	102 Destination unreachable (Port unreachable)
32	25.831154047	130.10.224.2	130.10.240.2	UDP	74 37672 → 33447 Len=32
33	25.831220592	130.10.240.2	130.10.224.2	ICMP	102 Destination unreachable (Port unreachable)
34	25.831311550	130.10.224.2	130.10.240.2	UDP	74 34789 → 33448 Len=32
35	25.831375929	130.10.240.2	130.10.224.2	ICMP	102 Destination unreachable (Port unreachable)
36	25.831430752	130.10.224.2	130.10.240.2	UDP	74 33599 → 33449 Len=32
37	25.831525074	130.10.240.2	130.10.224.2	ICMP	102 Destination unreachable (Port unreachable)
39	30.976568687	00:00:00_aa:00:21	00:00:00_aa:00:22	ARP	42 Who has 130.10.224.2? Tell 130.10.224.1
40	30.976647590	00:00:00_aa:00:22	00:00:00_aa:00:21	ARP	42 130.10.224.2 is at 00:00:00_aa:00:22

▶ Frame 36: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface veth1d.0.c1, id 0
 ▶ Ethernet II, Src: 00:00:00_aa:00:22 (00:00:00_aa:00:22), Dst: 00:00:00_aa:00:21 (00:00:00_aa:00:21)
 ▶ Internet Protocol Version 4, Src: 130.10.224.2, Dst: 130.10.240.2
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 60
 Identification: 0xdd96 (56726)
 ▶ Flags: 0x0000
 Fragment offset: 0
 Time to live: 6
 Protocol: UDP (17)
 Header checksum: 0x0301 [validation disabled]
 [Header checksum status: Unverified]

Figura 11 - Wireshark do nodo origem com ttl final.

*veth1e.0.c1 (on corevm)						
No.	Time	Source	Destination	Protocol	Length	Info
4	23.014984716	00:00:00_aa:00:25	Broadcast	ARP	42	Who has 130.10.240.2? Tell 130.10.240.1
5	23.015013559	00:00:00_aa:00:26	00:00:00_aa:00:25	ARP	42	130.10.240.2 is at 00:00:00_aa:00:26
6	23.015027650	130.10.224.2	130.10.240.2	UDP	74	46924 → 33446 Len=32
7	23.015048156	130.10.240.2	130.10.224.2	ICMP	102	Destination unreachable (Port unreachable)
8	23.015160891	130.10.224.2	130.10.240.2	UDP	74	37672 → 33447 Len=32
9	23.015173129	130.10.240.2	130.10.224.2	ICMP	102	Destination unreachable (Port unreachable)
10	23.015313649	130.10.224.2	130.10.240.2	UDP	74	34789 → 33448 Len=32
11	23.015327676	130.10.240.2	130.10.224.2	ICMP	102	Destination unreachable (Port unreachable)
12	23.015441691	130.10.224.2	130.10.240.2	UDP	74	33599 → 33449 Len=32
13	23.015464594	130.10.240.2	130.10.224.2	ICMP	102	Destination unreachable (Port unreachable)
14	28.160275243	00:00:00_aa:00:26	00:00:00_aa:00:25	ARP	42	Who has 130.10.240.1? Tell 130.10.240.2
15	28.160630864	00:00:00_aa:00:25	00:00:00_aa:00:26	ARP	42	130.10.240.1 is at 00:00:00_aa:00:25

▶ Frame 4: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth1e.0.c1, id 0
 ▶ Ethernet II, Src: 00:00:00_aa:00:25 (00:00:00_aa:00:25), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ▶ Address Resolution Protocol (request)

0000	ff ff ff ff ff ff 00 00	00 aa 00 25 08 06 00 01%.....
0010	08 00 06 04 00 01 00 00	00 aa 00 25 82 0a f0 01%.....
0020	00 00 00 00 00 00 82 0a	f0 02

Figura 12 - Wireshark do nodo destino do traceroute.

Utilizando o comando ping com flooding conseguimos obter um resultado visual do caminho feito nesse traceroute.

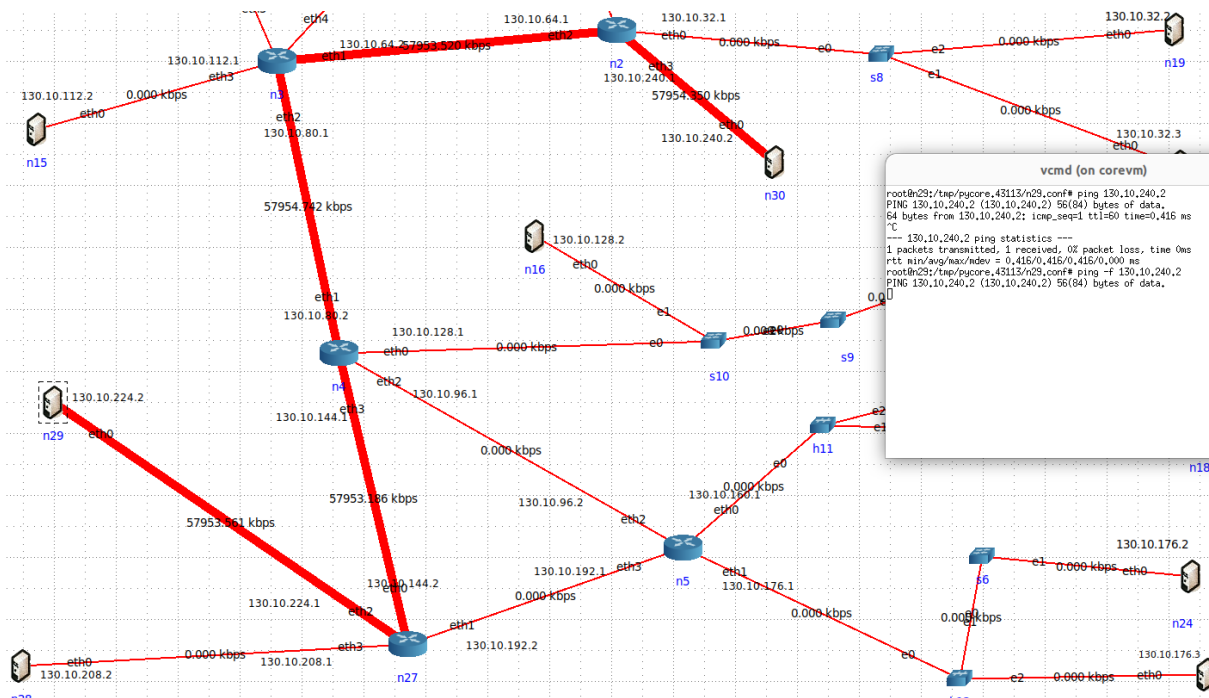


Figura 13 - Flooding do traceroute.

3.2. Nodo 14 -> nodo 28

No segundo traceroute do nodo 14 até o nodo 28 o comportamento se manteve igual ao primeiro exemplo. Na imagem 14 abaixo, no lado esquerdo está o terminal e wireshark do nodo origem e na direita está o terminal e wireshark do nodo destino.

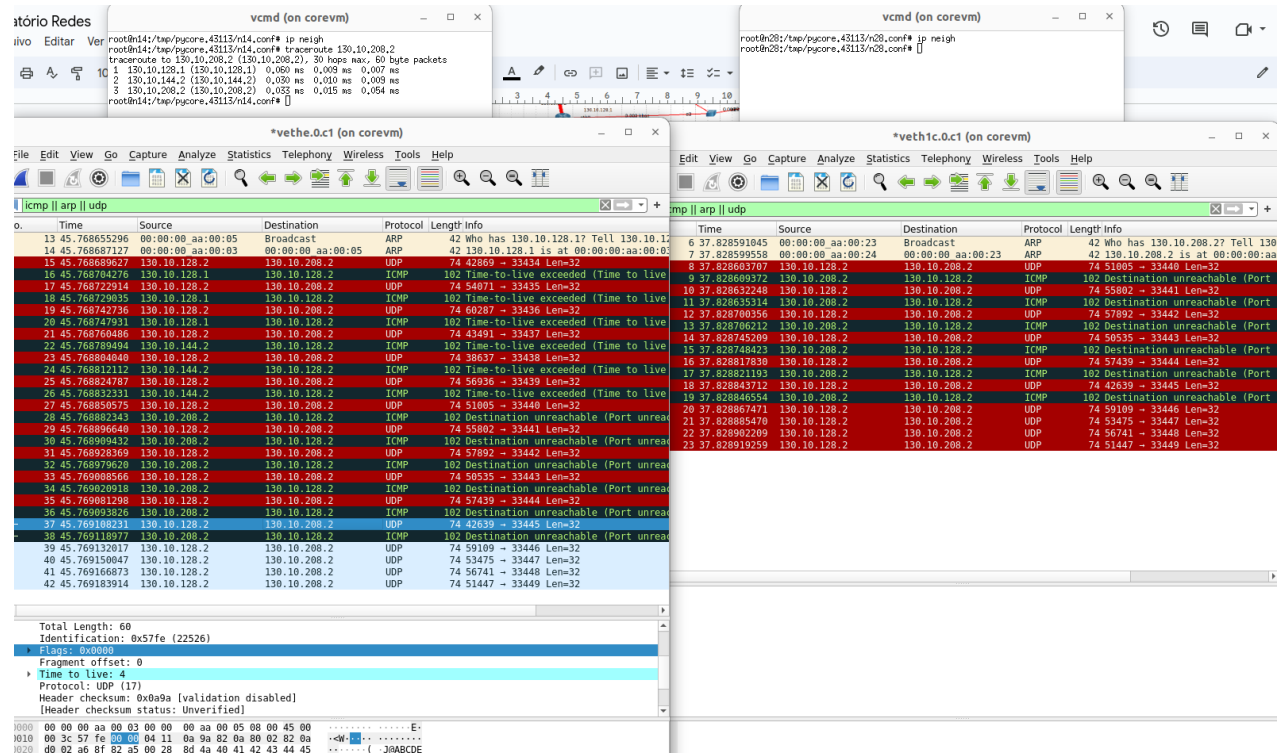


Figura 14 - Segundo traceroute.