

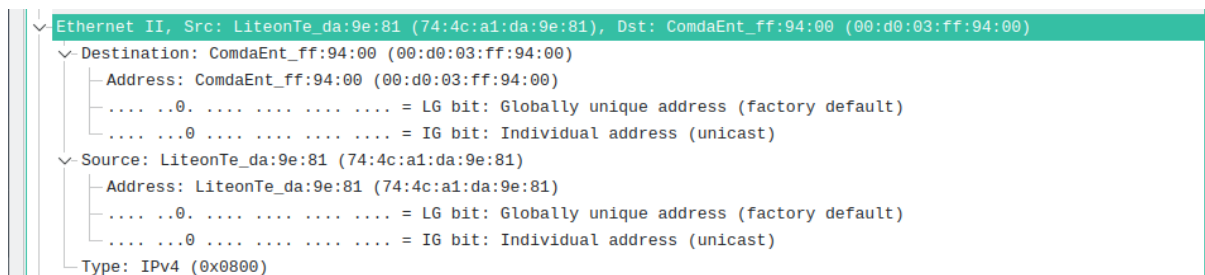
## Redes de Computadores

### Trabalho Prático 3

Pedro Afonso Moreira Lopes [A100759], Gonalo Machado Daniel Costa [A100824] e Jos  Eduardo Silva Monteiro Santos Oliveira [A100547]

### 3. Captura e an lise de Tramas Ethernet

**3.1. Anote os endereos MAC de origem e de destino da trama capturada. Identifique a que sistemas se referem. Justifique.**



**Figura 1.** Informa o da trama capturada

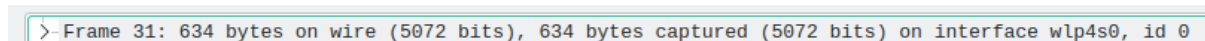
  poss vel concluir que o endereo MAC origem   **LiteonTe\_da:9e:81 (74:4c:a1:da:9e:81)** e o de destino   **ComdaEnt\_ff:94:00 (00:d0:03:ff:94:00)**.

O endereo MAC de origem refere-se ao nosso computador, pois refere-se   interface da nossa m quina nativa e o MAC de destino refere-se ao router da rede local ao qual o computador utilizado para capturar a trama estava conectado.

**3.2. Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?**

A partir da informa o que temos dispon vel na figura 1, conseguimos identificar que o valor hexadecimal do campo Type da trama   igual a **0x0800**. Este campo serve para identificar o protocolo encapsulado no campo de dados da trama, ou seja, IPv4.

**3.3. Quantos bytes s o usados no encapsulamento protocolar, i.e. desde o in cio da trama at  ao in cio dos dados do n vel aplicacional (Application Data Protocol: http-over-tls, no caso de HTTPS)? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar.**



**Figura 2.** Tamanho da trama



**Figura 3.** Tamanho do payload TCP

Nós conseguimos calcular quantos bytes são usados no encapsulamento protocolar através da subtração do tamanho do payload TCP ao tamanho total da trama, com os valores nas figuras acima. Sabendo isto, foram usados **634 - 568 = 66 bytes** no encapsulamento protocolar. Com este valor calculado, conseguimos calcular o valor de percentagem de overhead, que vai corresponder a **(66/634)\*100 = 10.41%**.

Agora verifiquemos o conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP proveniente do servidor:

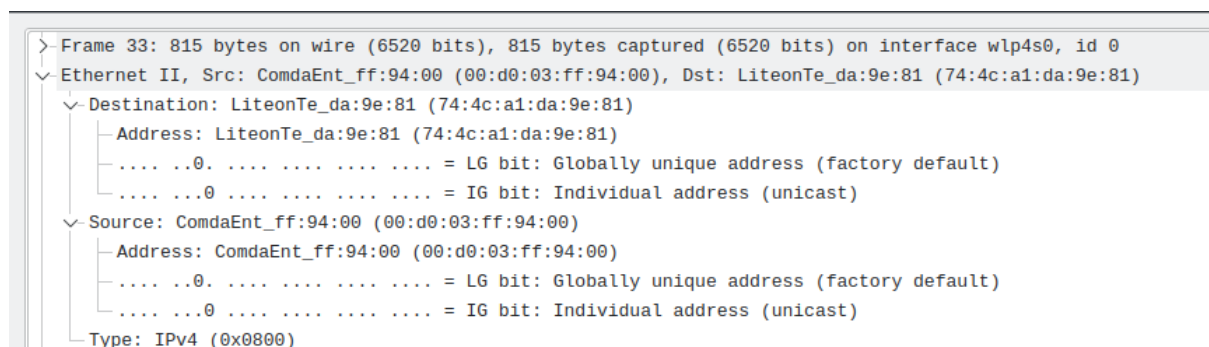


Figura 4. Tabela ARP

### 3.4. Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique.

O endereço Ethernet da fonte é **ComdaEnt\_ff:94:00 (00:d0:03:ff:94:00)**, que vai corresponder ao router da rede local ao qual estamos ligados.

### 3.5. Qual é o endereço MAC do destino? A que sistema (host) corresponde?

Como se pode ver pela figura, o endereço MAC do destino é **LiteonTe\_da:9e:81 (74:4c:a1:da:9e:81)**. O endereço MAC é usado para identificar os dispositivos físicos de origem e destino no segmento da rede local. Sabendo isto e o sentido de envio da trama, este endereço refere-se ao nosso computador, no qual foi realizada esta questão.

### 3.6. Atendendo ao conceito de encapsulamento protocolar, identifique os vários protocolos contidos na trama recebida. Justifique, indicando em que campos dos cabeçalhos capturados se baseou.

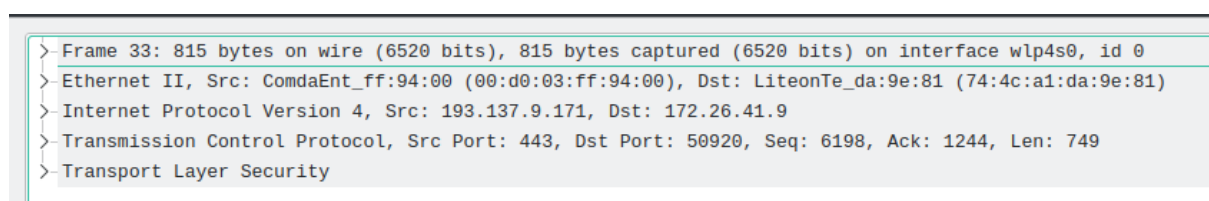


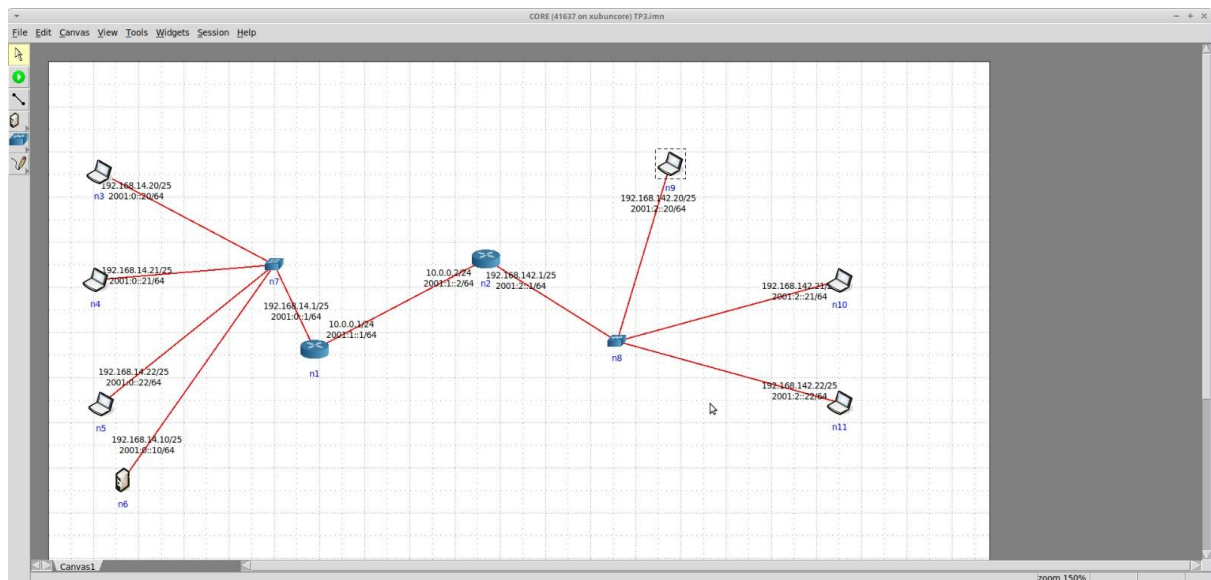
Figura 5. Protocolos contidos na trama recebida

Como se pode ver pela figura, os protocolos contidos na trama são:

- Ethernet II
- Internet Protocol Version 4 (IPv4)
- Transmission Control Protocol (TCP)
- Transport Layer Security (TLS)

#### 4. Protocolo ARP

**4.1. Abra uma consola no PC onde efetuou o ping. Observe o conteúdo da tabela ARP com o comando `arp -a`.**



**Figura.6.** Topologia utilizada na resolução deste exercício

**4.1.a).** Com a ajuda do manual ARP (man arp), interprete o significado de cada uma das colunas da tabela.

A primeira coluna representa os IPs dos Hosts, a segunda o tipo de conexão feita, a terceira o endereço MAC do destino, a quarta representa as flags( O C representa que a conexão foi bem sucedida), e a última coluna revela qual foi a interface do dispositivo de destino.

**4.1.b).** Indique, justificando, qual o equipamento da intranet em causa que poderá apresentar a maior tabela ARP em termos de número de entradas.

Será o router B, já que vai armazenar os endereços relativos ao router A e aos dois PCs, já que foi o que teve mais ligações diretas, logo vai ter a tabela ARP mais cheia.

**4.2. Observe a trama Ethernet que contém a mensagem com o pedido ARP (ARP Request).**

**4.2.a). Qual é o valor hexadecimal dos endereços MAC origem e destino? Como interpreta e justifica o endereço destino usado?**

origem -> 00:00:00\_aa:00:03

destino-> 00:00:00:00:00:00

Como está a ser uma mensagem broadcasted, não está a enviar para um endereço MAC específico, logo não tem nenhum endereço MAC no endereço destino.

**4.2.b). Qual o valor hexadecimal do campo Tipo da trama Ethernet? O que indica?**

0x0806, o que indica que é um pedido ARP.

**4.2.c). Observando a mensagem ARP, como pode saber que se trata efetivamente de um pedido ARP? Refira duas formas distintas de obter essa informação.**

Para obtermos essa informação podemos olhar para o opcode, e se tiver um 1 então é um pedido ARP(se tivesse um 2 seria uma resposta ARP). Outra maneira seria olhar para o MAC de destino, e verificar se ele possuía um endereço específico ou não. Como não possui, podemos identificar que é um broadcast, que é uma característica dos pedidos ARP.

**4.2.d). Explícite, em linguagem comum, que tipo de pedido ou pergunta é feita pelo host de origem à rede?**

Pergunta à rede se existe alguma máquina que está a utilizar aquele endereço.

**4.3. Localize a mensagem ARP que é a resposta ao pedido ARP efetuado.**

**4.3.a). Qual o valor do campo ARP opcode? O que especifica?**

É uma reply, ou seja, simboliza que é uma resposta ao pedido ARP efetuado.

**4.3.b). Em que posição da mensagem ARP está a resposta ao pedido ARP efetuado?**

Está nos bytes 26 a 30.

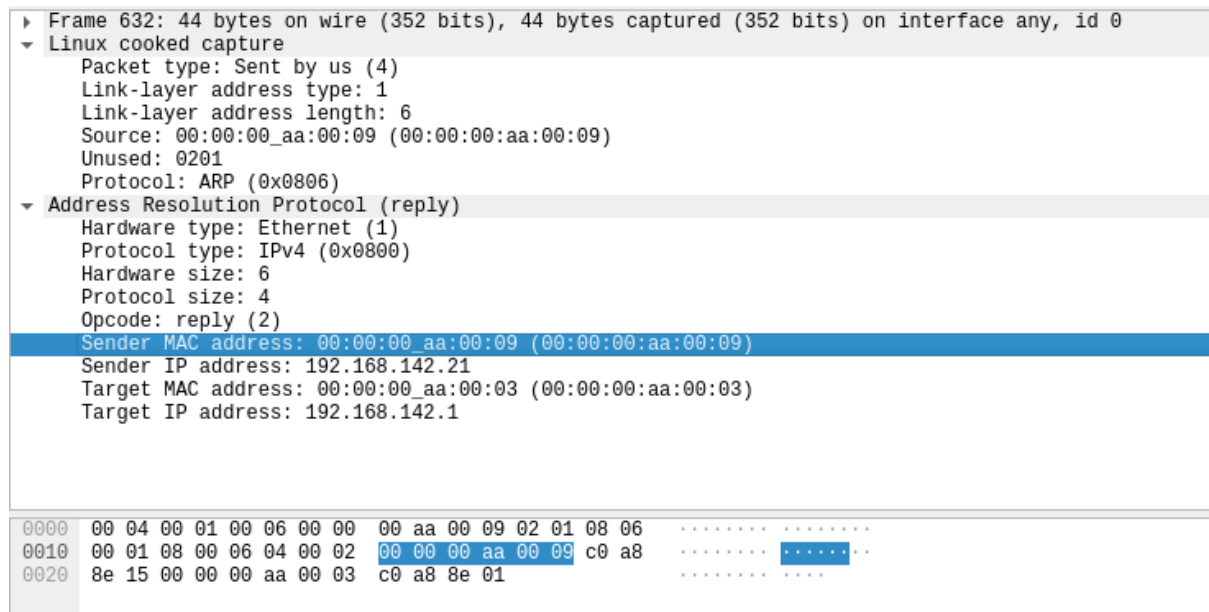


Figura 7. Localização dos bytes com a resposta

4.3.c). Identifique a que sistemas correspondem os endereços MAC de origem e de destino da trama em causa, recorrendo aos comandos `ifconfig`, `netstat -rn` e `arp` executados no PC selecionado.

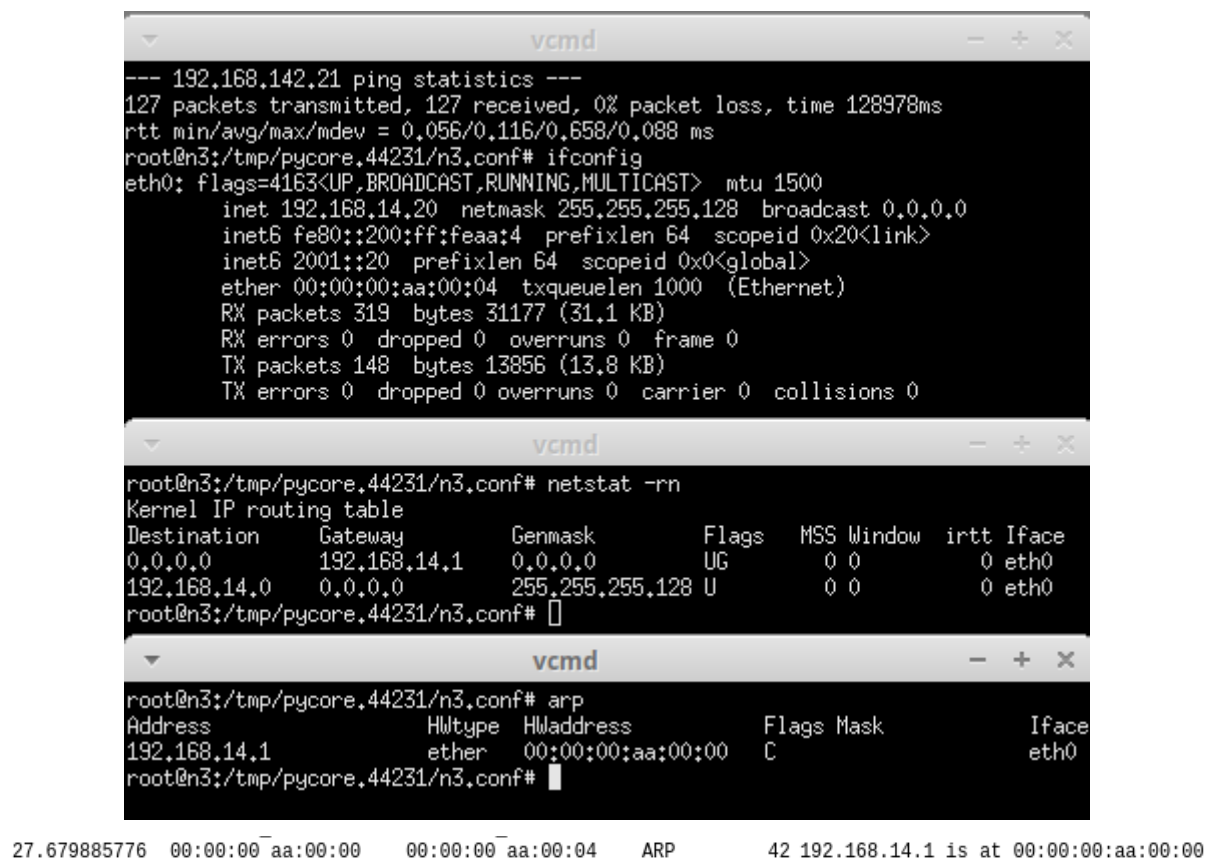


Figura 8. Ipconfig, netstat -rn, tabela arp e wireshark no computador n3

O endereço do transmissor é o do router RA e o endereço de destino é o n3.

**4.3.d). Justifique o modo de comunicação (unicast vs broadcast) usado no envio da resposta ARP (ARP Reply).**

Utiliza o Unicast pois a source conhece o endereço MAC de quem fez o pedido ARP.

**4.4. Verifique se o ping feito ao segundo PC originou pacotes ARP. Justifique a situação observada.**

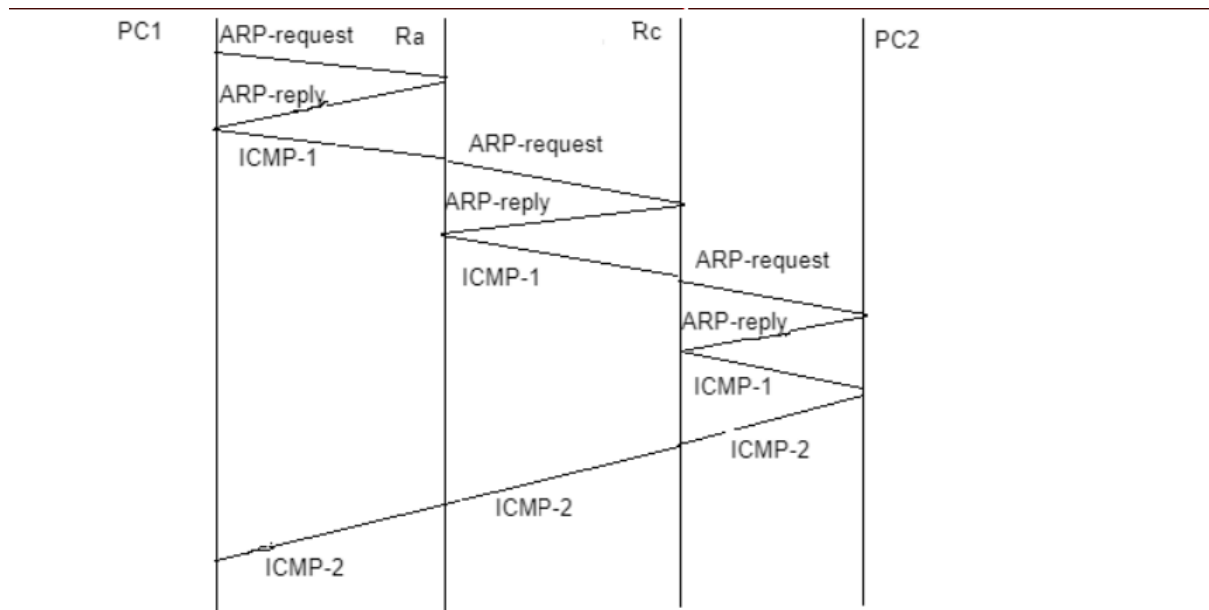
O segundo ping também gera pacotes ARP. Como o endereço MAC já tinha sido guardado na cache, desta vez não foi necessário utilizar Broadcast.

**4.5. Identifique na mensagem ARP os campos que permitem definir o tipo e o tamanho dos endereços das camadas de rede e de ligação lógica que se pretendem mapear. Justifique os valores apresentados nesses campos.**

A mensagem ARP possui os campos necessários para definir o tipo e o tamanho dos endereços das camadas de rede e de ligação lógica que se pretendem mapear. Esses campos são o Hardware type, o Protocol type, o Hardware size e o Protocol size.

- Hardware type -> Camada de ligação lógica que se vai mapear (2 bytes e o 1 significa que o tipo do endereço é Ethernet).
- Protocol type -> Tipo de endereço da camada de rede (2 bytes). Como é 0x800 isto significa que o endereço é IPv4.
- Hardware size -> Tamanho do endereço da camada de ligação lógica (1 byte, para Ethernet tem o valor 6).
- Protocol size -> Tamanho da camada de rede (1 byte, que para endereços IPv4 tem valor 4).

**4.6. Na situação em que efetua um ping a um PC não local à sua sub-rede, esboce um diagrama em que indique claramente, e de forma cronológica, todas as mensagens ARP e ICMP trocadas, até à recepção da resposta ICMP do sistema destino (represente apenas os nós intervenientes). Assuma que todas as tabelas ARP se encontram inicialmente vazias.**



**Diagrama 1.**

Diagrama com os pedidos ARP e ICMP desde o computador de origem, um computador da subrede de A e um computador de outra subrede, conectado a um router, que por sua vez está conectado ao router A.

## 5. Domínios de colisão

**5.1. Através da opção tcpdump, verifique e compare como flui o tráfego nas diversas interfaces dos vários dispositivos no departamento A (LAN comutada) e no departamento B (LAN partilhada) quando é gerado tráfego intra-departamento (por exemplo, através do comando ping). Que conclui?**

**Comente os resultados obtidos quanto à utilização de hubs e switches no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.**

```

vcmd
23:32:14.426839 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 44
23:32:14.979834 IP 192.168.14.20 > 192.168.14.22: ICMP echo request, id 35, seq 456, length 64
23:32:14.979902 IP 192.168.14.22 > 192.168.14.20: ICMP echo reply, id 35, seq 456, length 64
23:32:16.004419 IP 192.168.14.20 > 192.168.14.22: ICMP echo request, id 35, seq 457, length 64
23:32:16.004500 IP 192.168.14.22 > 192.168.14.20: ICMP echo reply, id 35, seq 457, length 64
23:32:16.430608 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 44
23:32:17.038526 IP 192.168.14.20 > 192.168.14.22: ICMP echo request, id 35, seq 458, length 64
23:32:17.038756 IP 192.168.14.22 > 192.168.14.20: ICMP echo reply, id 35, seq 458, length 64
23:32:17.597961 IP6 fe80::200:ff:feaa:0 > ff02::5: OSPFv3, Hello, length 36
23:32:18.060548 IP 192.168.14.20 > 192.168.14.22: ICMP echo request, id 35, seq 459, length 64
23:32:18.060612 IP 192.168.14.22 > 192.168.14.20: ICMP echo reply, id 35, seq 459, length 64
25 packets captured
25 packets received by filter
0 packets dropped by kernel
root@n5:/tmp/pycore.44231/n5.conf#

vcmd
1, length 64
23:29:16.183486 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 44
23:29:17.031634 IP 192.168.14.20 > 192.168.14.21: ICMP echo request, id 27, seq 292, length 64
23:29:17.031639 IP 192.168.14.21 > 192.168.14.20: ICMP echo reply, id 27, seq 292, length 64
23:29:17.494780 IP6 fe80::200:ff:feaa:0 > ff02::5: OSPFv3, Hello, length 36
23:29:18.024859 ARP, Request who-has 192.168.14.21 tell 192.168.14.20, length 28
23:29:18.024888 ARP, Reply 192.168.14.21 is-at 00:00:00:aa:00:05 (oui Ethernet), length 28
23:29:18.052198 IP 192.168.14.20 > 192.168.14.21: ICMP echo request, id 27, seq 293, length 64
23:29:18.052294 IP 192.168.14.21 > 192.168.14.20: ICMP echo reply, id 27, seq 293, length 64
23:29:18.184584 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 44
23:29:19.077627 IP 192.168.14.20 > 192.168.14.21: ICMP echo request, id 27, seq 294, length 64
23:29:19.077634 IP 192.168.14.21 > 192.168.14.20: ICMP echo reply, id 27, seq 294, length 64
20 packets captured
20 packets received by filter
0 packets dropped by kernel
root@n4:/tmp/pycore.44231/n4.conf#

```

Figura 9 . Outputs do comando tcpdump nos pcs n4 e n5 no Departamento A

```

vcmd
23:42:37.892783 IP 192.168.142.22 > 192.168.142.20: ICMP echo reply, id 35, seq 29, length 64
23:42:38.628305 IP 192.168.142.20 > 192.168.142.21: ICMP echo request, id 43, seq 23, length 64
23:42:38.628616 IP 192.168.142.21 > 192.168.142.20: ICMP echo reply, id 43, seq 23, length 64
23:42:38.915755 IP 192.168.142.20 > 192.168.142.22: ICMP echo request, id 35, seq 30, length 64
23:42:38.915855 IP 192.168.142.22 > 192.168.142.20: ICMP echo reply, id 35, seq 30, length 64
23:42:39.404960 IP 192.168.142.1 > 224.0.0.5: OSPFv2, Hello, length 44
23:42:39.651603 IP 192.168.142.20 > 192.168.142.21: ICMP echo request, id 43, seq 24, length 64
23:42:39.651685 IP 192.168.142.21 > 192.168.142.20: ICMP echo reply, id 43, seq 24, length 64
23:42:39.939517 IP 192.168.142.20 > 192.168.142.22: ICMP echo request, id 35, seq 31, length 64
23:42:39.939587 IP 192.168.142.22 > 192.168.142.20: ICMP echo reply, id 35, seq 31, length 64
42 packets captured
42 packets received by filter
0 packets dropped by kernel
root@n10:/tmp/pycore.44231/n10.conf#

vcmd
23:42:39.651842 IP 192.168.142.21 > 192.168.142.20: ICMP echo reply, id 43, seq 24, length 64
23:42:39.939516 IP 192.168.142.20 > 192.168.142.22: ICMP echo request, id 35, seq 31, length 64
23:42:39.939580 IP 192.168.142.22 > 192.168.142.20: ICMP echo reply, id 35, seq 31, length 64
23:42:40.675567 IP 192.168.142.20 > 192.168.142.21: ICMP echo request, id 43, seq 25, length 64
23:42:40.675630 IP 192.168.142.21 > 192.168.142.20: ICMP echo reply, id 43, seq 25, length 64
23:42:40.964502 IP 192.168.142.20 > 192.168.142.22: ICMP echo request, id 35, seq 32, length 64
23:42:40.964521 IP 192.168.142.22 > 192.168.142.20: ICMP echo reply, id 35, seq 32, length 64
23:42:41.405183 IP 192.168.142.1 > 224.0.0.5: OSPFv2, Hello, length 44
23:42:41.699709 IP 192.168.142.20 > 192.168.142.21: ICMP echo request, id 43, seq 26, length 64
23:42:41.699731 IP 192.168.142.21 > 192.168.142.20: ICMP echo reply, id 43, seq 26, length 64
16 packets captured
16 packets received by filter
0 packets dropped by kernel
root@n11:/tmp/pycore.44231/n11.conf#

```

Figura 10 . Outputs do comando tcpdump nos pcs n10 e n11 no Departamento B

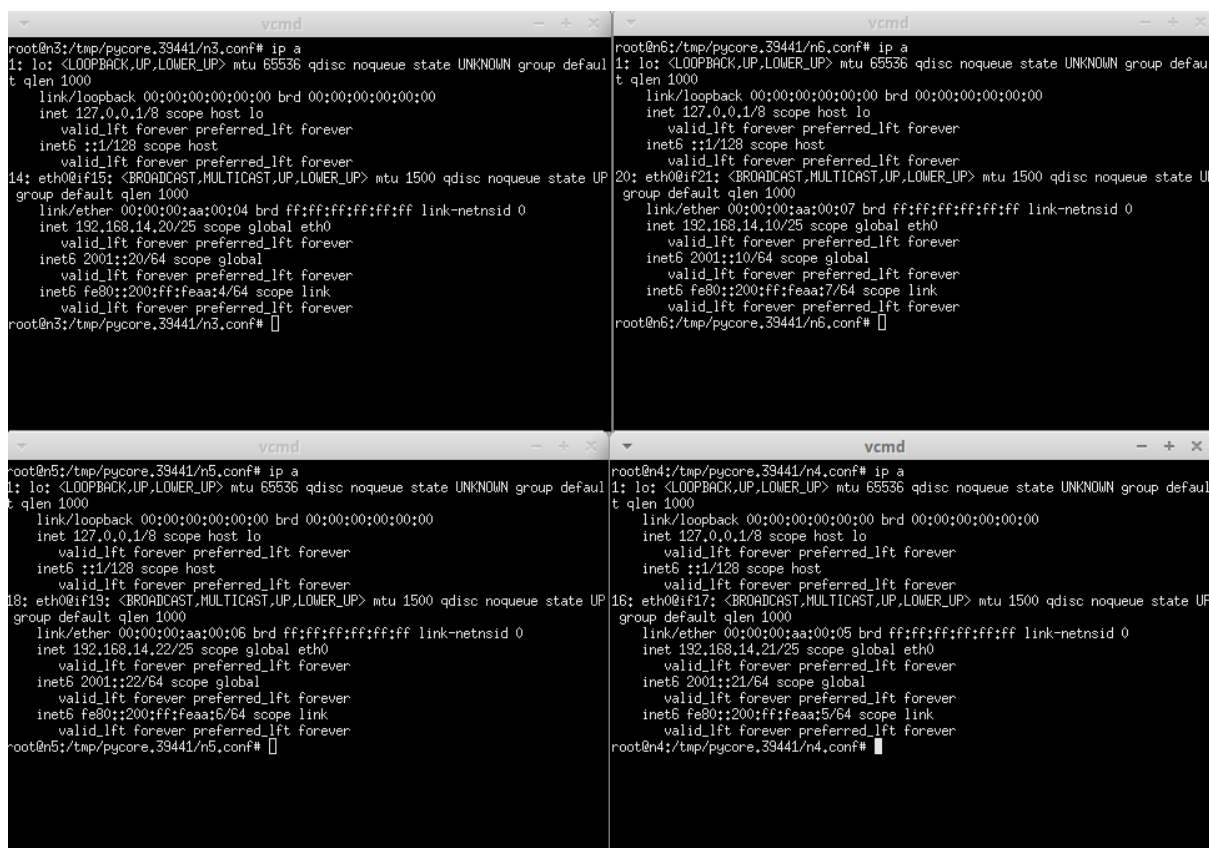


No departamento A, verificamos que cada PC só recebe Echo Request do seu ping pois este departamento funciona com um switch, logo cada computador tem o seu domínio de colisão individual.

No departamento B, por outro lado, verificamos uma repetição dos Echo Request no PC um do outro. Isso acontece pois o departamento B funciona com um Hub, logo todos os computadores partilham o domínio de colisão, resultando neste acontecimento.

## 5.2. Construa manualmente a tabela de comutação do switch do Departamento A, atribuindo números de porta à sua escolha.

```
vcmd
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
t qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP g
roup default qlen 1000
  link/ether 00:00:00:aa:00:00 brd ff:ff:ff:ff:ff:ff link-netnsid 0
  inet 192.168.14.1/25 scope global eth0
    valid_lft forever preferred_lft forever
  inet6 2001::1/64 scope global
    valid_lft forever preferred_lft forever
  inet6 fe80::200:ff:feaa:0/64 scope link
    valid_lft forever preferred_lft forever
8: eth1@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP g
roup default qlen 1000
  link/ether 00:00:00:aa:00:01 brd ff:ff:ff:ff:ff:ff link-netnsid 0
  inet 10.0.0.1/24 scope global eth1
    valid_lft forever preferred_lft forever
  inet6 2001:1::1/64 scope global
    valid_lft forever preferred_lft forever
  inet6 fe80::200:ff:feaa:1/64 scope link
    valid_lft forever preferred_lft forever
root@n1:/tmp/pycore.39441/n1.conf#
```



**Figuras 11 e 12.** Outputs do comando “ip a” para diferentes portas

Através dos outputs do comando “ip a” realizado nas diversas portas do switch do departamento A, foi capaz de se chegar à seguinte tabela de comutação:

Endereço MAC	Porta	TTL
00:00:00:aa:00:04(n3)	1	20
00:00:00:aa:00:05(n4)	2	20
00:00:00:aa:00:06(n5)	3	20
00:00:00:aa:00:03(HostA)	4	20
00:00:00:aa:00:02 (RA)	5	20