TP2 - Camada de Ligação Lógica: Ethernet e Protocolo ARP

Questões e Respostas

- ## Parte I

HTTP

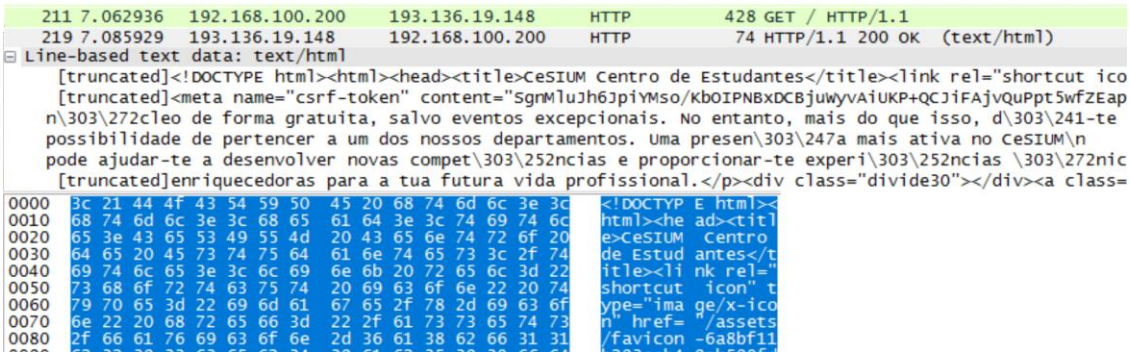


Imagem 0 – Número de ordem da sequência da mensagem HTTP GET é 211

1. **Qual é o endereço MAC da interface ativa do seu computador?**

   O endereço MAC é 00:e0:4c:60:3d:d0.

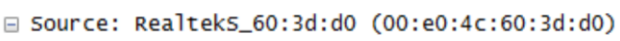


Imagem 1 – Endereço MAC da origem

2. **Qual é o endereço MAC destino da trama? A que sistema é destinada essa trama, será o endereço Ethernet do servidor *http* para *cesium.di.uminho.pt*? Justifique.**

   O endereço MAC é 00:0c:29:d2:19:f0. Não, a trama *Ethernet* é destinada ao router da rede local da sala de aula. Isto porque como as redes locais do nosso computador e do servidor onde está alojado o servidor do *Cesium* estão separadas por um ou mais routers, ao estabelecermos uma conexão, o nosso computador não comunica diretamente com o *site* mas sim com o router da sua rede local que depois se encarregará de comunicar com o servidor.

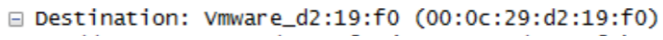


Imagem 2 – Endereço MAC do destino

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

   O valor hexadecimal é 0x0800. Diz-nos que o protocolo utilizado ao nível da rede é o IPv4.

Imagem 3 - Campo Type da trama Ethernet

4. **Quantos bytes são usados desde o início da trama até ao caractere ASCII "G" do método *HTTP GET*? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar no envio do *HTTP GET*.**

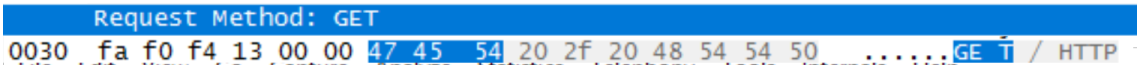


Imagem 4 - Localização do caractere "G" na trama Ethernet

São utilizados 54 bytes antes do caractere "G" (47 é o código ASCII da letra "G" em hexadecimal). *Overhead* = 54/428 = 12,6%, sendo 428 bytes o tamanho total da trama Ethernet e 54 o número de bytes de controlo.

5. **Em ligações com fios pouco suscetíveis a erros, nem sempre as *NICs* geram o código de deteção de erros. Verifique se o campo *FCS* está a ser utilizado. Aceda à opção *Edit/Preferences/Protocols/Ethernet* e indique que é assumido o uso do campo *FCS*. Verifique qual o valor hexadecimal desse campo na trama capturada. Que conclui? Reponha a configuração original.**

   Não, o campo *FCS* não está a ser utilizado pois a especificação da trama no *Wireshark* não faz referência a esse campo (apenas ao do tipo).

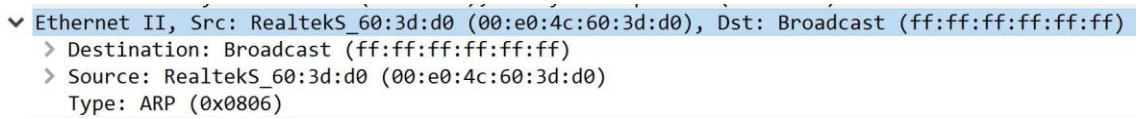
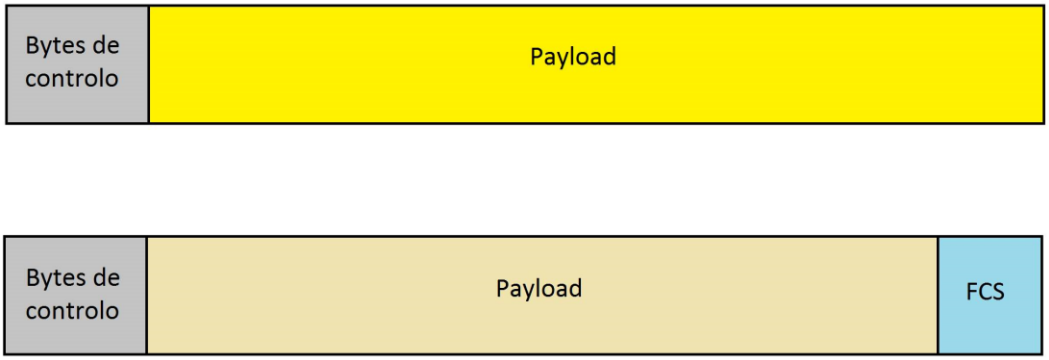


Imagem 5 - Especificação da trama Ethernet

Ao dizermos ao *Wireshark* para assumir o uso do campo *FCS* estamos, na verdade, a fazer com que ele utilize os últimos 4 bytes do *payload* para o campo *FCS* (a azul na imagem abaixo).

Assim – e apesar do *Wireshark* nos dizer que como estes 4 bytes diferem dos que resultam da aplicação do polinómio gerador nos dados recebidos deve ter havido um erro na transmissão da trama –, tal indicação é-nos irrelevante pois como os *bytes* utilizados pelo *Wireshark* para o campo *FCS* são do *payload*, não têm qualquer valor a nível da deteção de erros na transmissão da trama.



Trama original (em cima) e trama após alterar as definições do Wireshark (em baixo)

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

   O endereço *Ethernet* é 00:0c:29:d2:19:f0. Corresponde ao router da rede local da sala de aula.

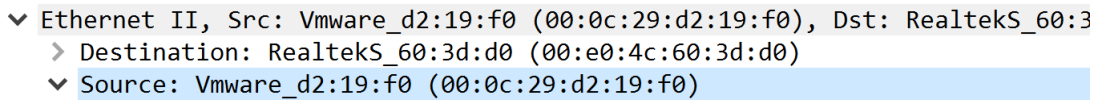


Imagem 6 - Endereço MAC da origem

7. **Qual é o endereço *MAC* do destino? A que sistema corresponde?**

   O endereço MAC é 00:e0:4c:60:3d:d0. Corresponde à placa de rede da nossa máquina.

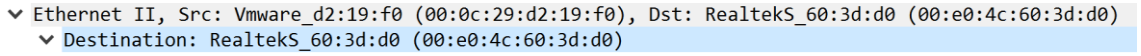


Imagem 7 - Endereço MAC do destino

8. **Qual é o valor hexadecimal do campo tipo (*Type*)?**

   O valor hexadecimal é 0x0800. Diz-nos que o protocolo utilizado ao nível da rede é o *IPv4*.

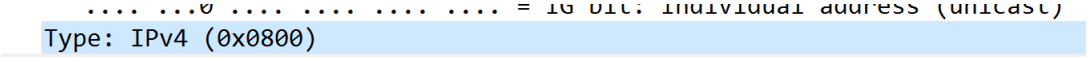


Imagem 8 - Campo Type da trama Ethernet

9. Que tipo de resposta foi enviada pelo servidor?

Foi uma resposta do tipo *text/html* (código da página inicial do site do *Cesium*).

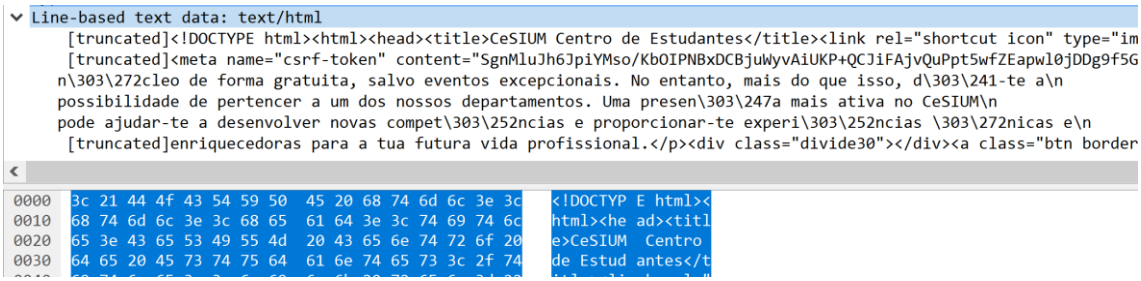


Imagem 9 - Resposta do servidor

Protocolo ARP

10. Observe o conteúdo da tabela *ARP*. Diga o que significa cada uma das colunas?

A primeira coluna faz referência ao endereço IP que é um endereço lógico e corresponde ao nível de rede. A segunda coluna faz referência ao endereço MAC que é um endereço físico e corresponde ao nível da ligação de dados. A terceira coluna corresponde à permanência da entrada da tabela ARP no sistema. Pode ser de dois tipos:

- *static*: endereços que são adicionados à tabela ARP permanentemente por um software específico e criado para o efeito;
- *dynamic*: endereços que são adicionados à tabela ARP depois de um ARP reply com sucesso e que permanecem no sistema temporariamente.

```
C:\WINDOWS\system32>arp -a

Interface: 192.168.100.200 --- 0xd
  Internet Address      Physical Address      Type
  192.168.100.202       50-b7-c3-77-a9-fe     dynamic
  192.168.100.254       00-0c-29-d2-19-f0     dynamic
  192.168.100.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Imagem 10 - Tabela ARP da máquina

11. Qual é o valor hexadecimal dos endereços origem e destino na trama Ethernet que contém a mensagem com o pedido *ARP* (*ARP Request*)? Como interpreta e justifica o endereço destino usado?

Os valores hexadecimais dos endereços de origem e destino são, respetivamente, *0x00 e0 4c 60 3d d0* (nossa máquina) e *0xff ff ff ff ff ff* (*broadcast*).

Como o site do MIEI está numa *subnet* diferente da rede local ao nosso computador (e ambas estão separadas por um ou mais *routers*), a comunicação terá obrigatoriamente de ser feita entre o primeiro router que liga ambas as redes e que se encarregará de todo o processo de encaminhamento.

No entanto, como a cache da tabela ARP da nossa máquina foi apagada, não existe a entrada correspondente ao *router* em questão. Assim, é necessário que o nosso computador envie um ARP *request* (com *ff:ff:ff:ff:ff:ff* como endereço de destino, denotando o *broadcast*) pedindo que lhe seja fornecido o endereço *MAC* correspondente ao IP do *router*.

Imagem 11 - Endereços de origem e destino da trama Ethernet

12. Qual o valor hexadecimal do campo tipo da trama *Ethernet*? O que indica?

O valor hexadecimal é *0x0806*. Indica que a trama *Ethernet* corresponde ao protocolo *ARP*.

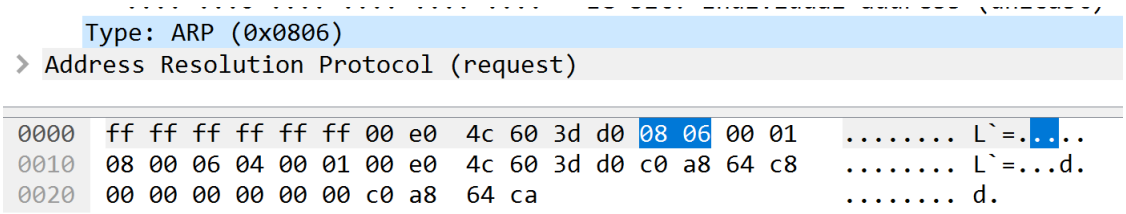


Imagem 12 - Campo Type da trama Ethernet

13. Qual o valor do campo *ARP opcode*? O que especifica? Se necessário, consulte a RFC do protocolo ARP http://tools.ietf.org/html/rfc826.html.

O *ARP opcode* é "*request* (1)". Especifica que o conteúdo que é do tipo relativo ao protocolo *ARP* é, mais concretamente, um *ARP request* (pedido para conhecer o endereço *MAC* a partir de um endereço IP).

```
Opcode: request (1)
Sender MAC address: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
Sender IP address: 192.168.100.200
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.100.202
```

```
0000  ff ff ff ff ff ff 00 e0  4c 60 3d d0 08 06 00 01   ........ L`=.....
0010  08 00 06 04 00 01 00 e0  4c 60 3d d0 c0 a8 64 c8   ........ L`=...d.
0020  00 00 00 00 00 00 c0 a8  64 ca                     ........ d.
```

Imagem 13 - Campo ARP opcode

14. A mensagem ARP contém o endereço IP de origem? Que tipo de pergunta é feita?

Sim, a mensagem *ARP* contém o endereço *IP* de origem (192.168.100.200), que corresponde à nossa máquina. É feito um pedido para conhecer o endereço *MAC* associado ao *IP* do *router*.

```
Sender MAC address: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
Sender IP address: 192.168.100.200
```

Imagem 14 - Endereço IP de origem

15. Localize a mensagem ARP que é a resposta ao pedido ARP efectuado.

a) **Qual o valor do campo *ARP opcode*? O que especifica?**
O valor do *ARP opcode* é "reply (2)". Indica que é a resposta ao *ARP request*, contendo a associação entre o endereço IP que lhe foi enviado e o endereço *MAC* que lhe era pedido (o do router).

```
Opcode: reply (2)
Sender MAC address: SamsungE_77:a9:fe (50:b7:c3:77:a9:fe)
Sender IP address: 192.168.100.202
Target MAC address: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
Target IP address: 192.168.100.200
```

```
0000  00 e0 4c 60 3d d0 50 b7  c3 77 a9 fe 08 06 00 01   ..L`=.P. .w......
0010  08 00 06 04 00 02 50 b7  c3 77 a9 fe c0 a8 64 ca   ......P. .w....d.
0020  00 e0 4c 60 3d d0 c0 a8  64 c8 00 00 00 00 00 00   ..L`=... d.......
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Imagem 15.1 - Campo ARP opcode

b) Em que posição da mensagem ARP está a resposta ao pedido ARP?

A resposta (endereço *MAC* 50:b7:c3:77:a9:fe) começa a partir do 23º byte.

```
Opcode: reply (2)
Sender MAC address: SamsungE_77:a9:fe (50:b7:c3:77:a9:fe)
Sender IP address: 192.168.100.202
Target MAC address: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
Target IP address: 192.168.100.200
```

```
0000  00 e0 4c 60 3d d0 50 b7  c3 77 a9 fe 08 06 00 01   ..L`=.P. .w......
0010  08 00 06 04 00 02 50 b7  c3 77 a9 fe c0 a8 64 ca   ......P. .w....d.
0020  00 e0 4c 60 3d d0 c0 a8  64 c8 00 00 00 00 00 00   ..L`=... d.......
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Imagem 15.2 - Resposta ao pedido ARP

16. Quais são os valores hexadecimais para os endereços origem e destino da trama que contém a resposta *ARP*? Que conclui?

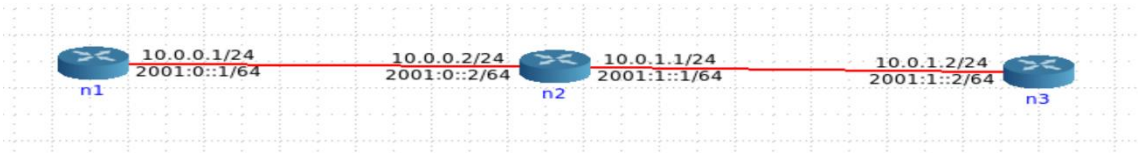
Os valores hexadecimais dos endereços de origem e destino são, respetivamente, *0x50 b7 c3 77 a9 fe* e *0x00 e0 4c 60 3d d0* (endereço *MAC*). Em primeiro lugar, podemos notar que o endereço de destino do *ARP reply* é o endereço de origem do *ARP request* e corresponde ao nosso computador. Em segundo, podemos dizer que o endereço de origem é o do *router*. Podemos ainda concluir que foi recebida a resposta ao pedido que foi feito ao *router* para conhecer o seu endereço *MAC*.

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: SamsungE_77:a9:fe (50:b7:c3:77:a9:fe)
    Sender IP address: 192.168.100.202
    Target MAC address: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
    Target IP address: 192.168.100.200
```

```
0000  00 e0 4c 60 3d d0 50 b7  c3 77 a9 fe 08 06 00 01   ..L`=.P. .w......
0010  08 00 06 04 00 02 50 b7  c3 77 a9 fe c0 a8 64 ca   ......P. .w....d.
0020  00 e0 4c 60 3d d0 c0 a8  64 c8 00 00 00 00 00 00   ..L`=... d.......
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Imagem 16 - Endereços origem e destino do ARP

ARP numa topologia CORE



Topologia com 3 *routers*

17. Com auxílio do comando *ifconfig* obtenha os endereços Ethernet das interfaces dos diversos routers.

```
> n1 > ifconfig:
eth0      Link encap:Ethernet  HWaddr 00:00:00:aa:00:00
          inet addr:10.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.0
```

Endereço Ethernet do router n1

```
> n2 > ifconfig:
eth0      Link encap:Ethernet  HWaddr 00:00:00:aa:00:01
          inet addr:10.0.0.2  Bcast:0.0.0.0  Mask:255.255.255.0

eth1      Link encap:Ethernet  HWaddr 00:00:00:aa:00:02
          inet addr:10.0.1.1  Bcast:0.0.0.0  Mask:255.255.255.0
```

Endereço Ethernet do router n2

```
> n3 > ifconfig:
eth0      Link encap:Ethernet  HWaddr 00:00:00:aa:00:03
          inet addr:10.0.1.2  Bcast:0.0.0.0  Mask:255.255.255.0
```

Endereço Ethernet do router n3

18. Usando o comando *arp* obtenha as caches *arp* dos diversos sistemas.

```
> n1 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.2                 ether   00:00:00:aa:00:01   C                     eth0

> n2 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.2                 ether   00:00:00:aa:00:03   C                     eth1
10.0.0.1                 ether   00:00:00:aa:00:00   C                     eth0

> n3 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.1                 ether   00:00:00:aa:00:02   C                     eth0
```

Imagem 18 – Tabelas arp de n1, n2 e n3

19. Faça *ping* de n1 para n2. Que modificações observa nas caches ARP desses sistemas? Faça *ping* de n1 para n3. Consulte as caches ARP. Que conclui?

Ao fazer um *ping* de N1 para N2, como já existia uma entrada na tabela *ARP* de N1 para os endereços de N2, não acontece nada. O mesmo ocorre para N3 pois o tráfego passa por N2 cujos endereços já eram conhecidos por N1.

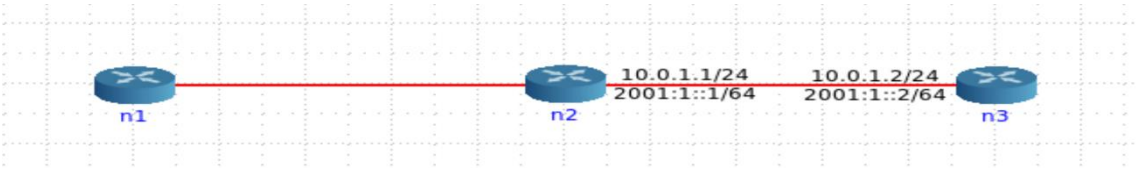
```
> n1 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.2                 ether   00:00:00:aa:00:01   C                     eth0

> n2 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.2                 ether   00:00:00:aa:00:03   C                     eth1
10.0.0.1                 ether   00:00:00:aa:00:00   C                     eth0

> n3 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.1                 ether   00:00:00:aa:00:02   C                     eth0
```

Imagem 19 – Tabelas ARP de n1, n2 e n3

20. Em n1 remova a entrada correspondente a n2. Coloque uma nova entrada para n2 com endereço *Ethernet* inexistente. O que acontece?



Desaparecem a entrada dos endereços de N2 da tabela ARP de N1 e a entrada dos endereços de N1 da tabela ARP de N2.

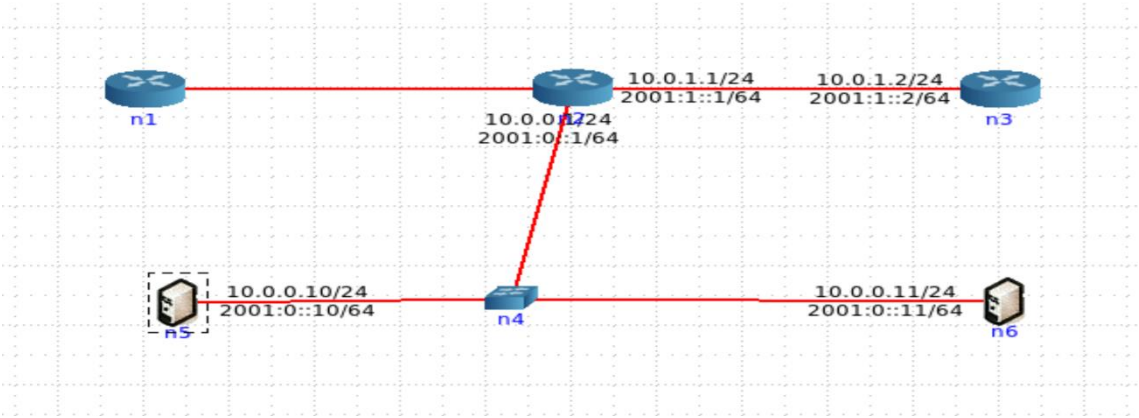
```
> n1 > arp:

> n2 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.2                 ether   00:00:00:aa:00:02   C                     eth1

> n3 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.1                 ether   00:00:00:aa:00:01   C                     eth0
```

Imagem 20 – Tabelas ARP de n1, n2 e n3

21. Faça *ping* de n5 para n6. Sem consultar a tabela *ARP* anote a entrada que, em sua opinião, é criada na tabela *ARP* de n5. Verifique, justificando, se a sua interpretação sobre a operação da rede *Ethernet* e protocolo *ARP* estava correto.



Na nossa opinião, como o que liga N5 a N6 é um *switch* que apenas se encarrega de reenviar os dados que recebe numa entrada (a de N5) para a saída correspondente (a de N6), a entrada que deve ser adicionada à tabela *ARP* de N5 é a entrada correspondente aos endereços de N6. No caso anterior, quando N1 faz *ping* a N3, tal já não acontece pois o que liga N1 a N2 é um outro *router* (e não um *switch*). Assim, N1 fica com a entrada correspondente aos endereços de N2 (e não de N3).

Numa abordagem mais concreta, se um computador fizer *ping* a um qualquer servidor da Google, por exemplo, a entrada na tabela *ARP* que vai ser adicionada ao computador não será a dos endereços (*Ethernet* e *MAC*) do servidor, mas sim a dos endereços do primeiro router que liga a rede local do computador à rede local do servidor.

```
> n1 > arp:

> n2 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.1.2                 ether   00:00:00:aa:00:02   C                     eth1

> n3 > arp:

> n5 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.11                ether   00:00:00:aa:00:05   C                     eth0

> n6 > arp:
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.10                ether   00:00:00:aa:00:04   C                     eth0
```

Imagem 21 – Tabelas ARP de n1, n2, n3, n5 e n6

- # PARTE II

ARP Gratuito

1. **Identifique um pacote de pedido ARP gratuito originado pelo seu sistema. Verifique quantos pacotes ARP gratuito foram enviados e com que intervalo temporal.**

   Foram enviados dois *ARP* gratuitos, um aos 9.57s e outro aos 68.67s. Registou-se, então, um intervalo temporal de 59.1s.

```
  261 9.575041  RealtekS_60:3d:d0  Broadcast  ARP  42 Gratuitous ARP for 192.168.100.200 (Request)
 3751 68.672232 AsustekC_06:79:c4  Broadcast  ARP  60 Gratuitous ARP for 192.168.100.222 (Request)
```

Imagem 22 – ARPs gratuitos

2. **Analise o conteúdo de um pedido ARP gratuito e identifique em que se distingue dos restantes pedidos ARP. Registe a trama Ethernet correspondente. Qual o resultado esperado face ao pedido ARP gratuito enviado?**

   No *ARP* gratuito – e em comparação com os restantes pedidos *ARP* – os endereços *IP* de origem e de destino são iguais (e correspondem ao do nosso computador), sendo o endereço *MAC* de destino do tipo *ff:ff:ff:ff:ff:ff* que denota o *broadcast*.

```
⊟ Ethernet II, Src: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊟ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  ⊟ Source: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
      Address: RealtekS_60:3d:d0 (00:e0:4c:60:3d:d0)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
```

Imagem 23 – Especificação do ARP gratuito

Domínios de colisão

1. **Faça *ping* de n1 para n2. Verifique com a opção *tcpdump* como flui o tráfego nas diversas interfaces dos vários dispositivos. Que conclui?**

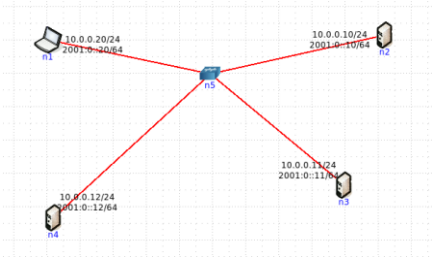


Imagem 24.1 – Sistema

```
root@n1:/tmp/pycore.46656/n1.conf# ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_req=1 ttl=64 time=0.048 ms
64 bytes from 10.0.0.10: icmp_req=2 ttl=64 time=0.064 ms
64 bytes from 10.0.0.10: icmp_req=3 ttl=64 time=0.060 ms
64 bytes from 10.0.0.10: icmp_req=4 ttl=64 time=0.055 ms
```

```
root@n2:/tmp/pycore.46656/n2.conf# tcpdump -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
06:49:33.984535 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1),
 length 84)
    10.0.0.20 > 10.0.0.10: ICMP echo request, id 39, seq 12, length 64
06:49:33.984556 IP (tos 0x0, ttl 64, id 56880, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.0.10 > 10.0.0.20: ICMP echo reply, id 39, seq 12, length 64
06:49:34.984633 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1)
```

```
root@n3:/tmp/pycore.46656/n3.conf# tcpdump -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
06:49:39.984535 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1),
 length 84)
    10.0.0.20 > 10.0.0.10: ICMP echo request, id 39, seq 18, length 64
06:49:39.984575 IP (tos 0x0, ttl 64, id 56886, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.0.10 > 10.0.0.20: ICMP echo reply, id 39, seq 18, length 64
06:49:40.985588 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1),
 length 84)
```

```
root@n4:/tmp/pycore.46656/n4.conf# tcpdump -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
06:49:53.985721 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1),
 length 84)
    10.0.0.20 > 10.0.0.10: ICMP echo request, id 39, seq 32, length 64
06:49:53.985757 IP (tos 0x0, ttl 64, id 56900, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.0.10 > 10.0.0.20: ICMP echo reply, id 39, seq 32, length 64
06:49:54.984729 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1),
 length 84)
```

Imagem 24.2 – Ping (n1 para n2) e tcpdump de n2, n3 e n4

Apesar de estarmos a fazer um *ping* de N1 para N2, N3 e N4 conseguem observar o tráfego gerado pois foi utilizado um *hub* no sistema.

2. **Na topologia de rede substitua o *hub* por um *switch*. Repita os procedimentos que realizou na pergunta anterior. 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.**

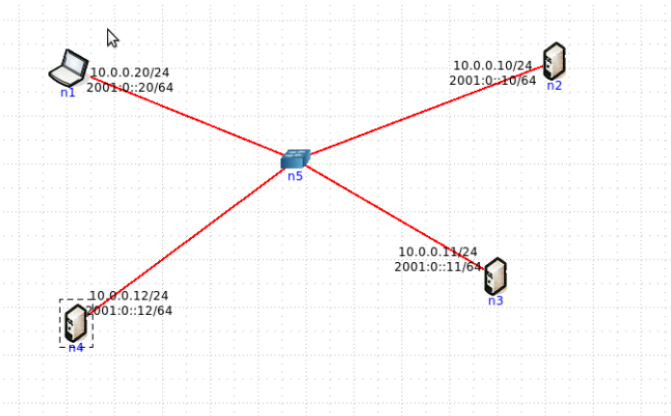


Imagem 25.1 - Sistema

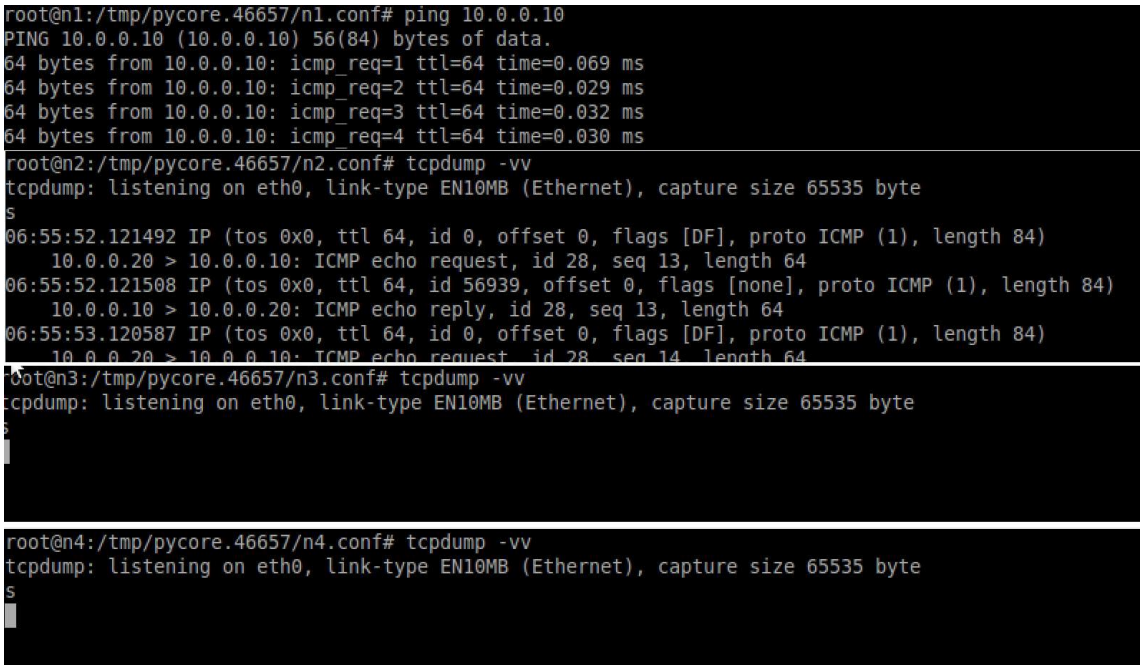


Imagem 25.2 – Ping (n1 para n2) e tcpdump de n2, n3 e n4

Ao contrário do sistema anterior, desta vez foi utilizado um *switch* e, assim, o tráfego gerado pelo ping de N1 para N2 apenas é observado por N2.

Num sistema que utilize um *hub*, quando uma máquina gera tráfego, todas as outras máquinas conseguem observar o tráfego gerado. Enquanto esta máquina estiver a gerar tráfego, nenhuma outra o consegue fazer, o que terá implicações a nível das colisões. Quanto maior o número de máquinas, maior é a probabilidade de existirem colisões.

Num sistema que utilize um *switch*, isto já não se verifica pois é “criado” um canal de comunicação quase exclusivo entre a origem e o destino, o que reduz a ocorrência de colisões.

- # Conclusões

Com este trabalho, pudemos pôr em prática os conhecimentos teóricos adquiridos nas aulas de Redes de Computadores e, assim, compreender melhor os mesmos de um ponto de vista mais real.

Na primeira parte, relativa aos protocolos *HTTP* e *ARP*, percebemos a distinção entre o endereço físico (*MAC*) e o endereço lógico (*IP*), bem como a interação do nosso computador com a rede local e a importância de ambos os endereços na comunicação com a rede local e também com outras *subnets*.

Inicialmente, tivemos algumas dificuldades em utilizar o *Wireshark*, nomeadamente em entender os resultados das capturas. Além disso, surgiram-nos algumas dúvidas em relação à comunicação entre *subnets* diferentes, mais concretamente na parte do *ARP Reply* e do papel dos routers na interação entre a nossa máquina e o site.

Na segunda parte, relativa à simulação de sistemas de rede físicos, percebemos a diferença entre os pedidos e respostas *ARP* da parte anterior e os pedidos *ARP* gratuitos. Percebemos também a diferença entre implementação de um *switch* e de um *hub* e do seu papel nas colisões de tráfego.