Redes de Computadores

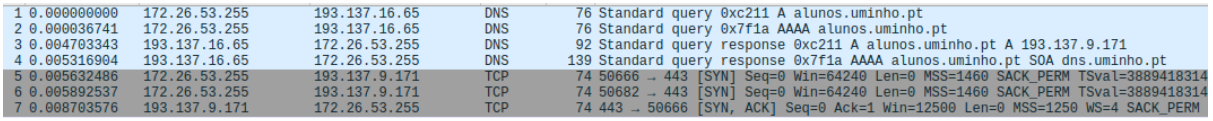
Trabalho Prático 3

Rodrigo Monteiro, Diogo Abreu, e Gustavo Barros
Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal
e-mail: {a100706,a100646,a100656}@alunos.uminho.pt

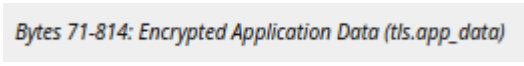
Parte 1

- Estabelecimento da conexão entre o cliente e o servidor

O primeiro passo para estabelecimento da conexão, é a transformação do link pesquisado (legível para humanos) num IP address (legível para máquinas). Este processo envolve o envio de queries, e a recepção de respostas do servidor.

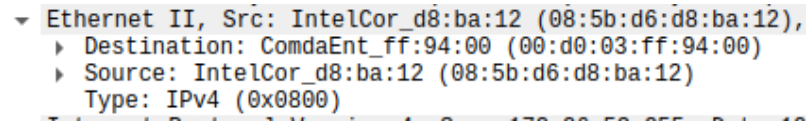


Dados aplicacionais enviados e recebidos



Nota: Como a ligação ao website é HTTPS (HTTP over TLS), o wireshark não tem a informação necessária para decifrar o TLS; como não consegue dissecar a informação encriptada, a camada mais acima que é reconhecida é TLS, e é, portanto, essa que é identificada pelo wireshark.

1. *Anote os endereços MAC de origem e de destino da trama capturada. Identifique a que sistemas se referem. Justifique.*



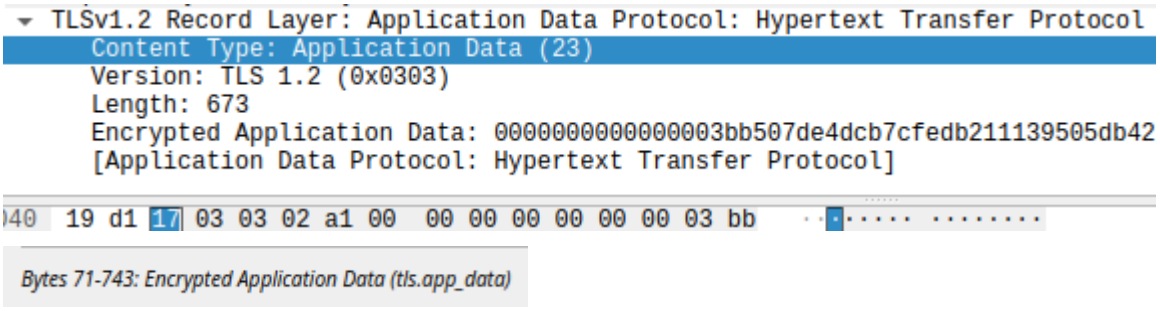

```
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group default qlen 1000
    link/ether 08:5b:d6:d8:ba:12 brd ff:ff:ff:ff:ff:ff
```

Portanto, o endereço MAC da origem é `08:5b:d6:d8:ba:12` (correspondente ao cliente) e o endereço MAC de destino é `00:d0:03:ff:94:00` (correspondente ao servidor).

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

O valor hexadecimal do campo Type da trama Ethernet é `0x0800` que corresponde ao protocolo da camada de rede que está a ser usado, sendo, neste caso, o protocolo IPv4 (existem outros protocolos como o ARP que será abordado posteriormente neste relatório). Deste modo, quando um dispositivo recebe uma trama com esse valor de Type, fica a saber que esta contém um pacote IPv4 e processa-o adequadamente.

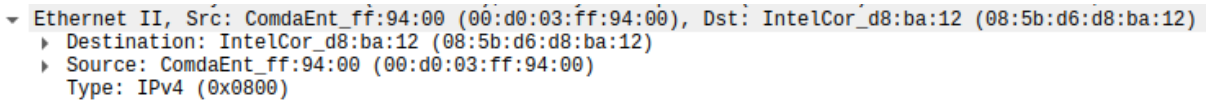
3. *Quantos bytes são usados no encapsulamento protocolar, i.e. desde o início da trama até ao início dos dados do nível aplicacional (`Application Data Protocol: http-over-tls, no caso de HTTPS`)? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar.*



Neste caso, são utilizados 67 bytes para overhead (devido à necessidade da existência da pilha protocolar). Depois, na Application Data Protocol, 1 byte para Content Type, 2 bytes para Version, 2 bytes para Length e 673 bytes para Encrypted Application Data.
Assim, existe uma sobrecarga de $\approx 10\%$.

(baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP proveniente do servidor.)

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



O endereço Ethernet, ou endereço MAC, da fonte é `00:d0:03:ff:94:00` (correspondente ao servidor). Cada endereço MAC tem os seus primeiros três bytes correspondentes a um certo "domínio" de rede de um dado fabricante de dispositivos (Organizationally Unique Identifier). Este domínio é atribuído a cada fabricante pela IEEE (Institute of Electrical and Electronics Engineers). Assim, como os três primeiros bytes são `00:d0:03`, significa que o fabricante do dispositivo da fonte é a Comda Enterprises Corp. Por outras palavras, o endereço Ethernet da fonte corresponde ao sistema de redes atribuído a essa fabricante.

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

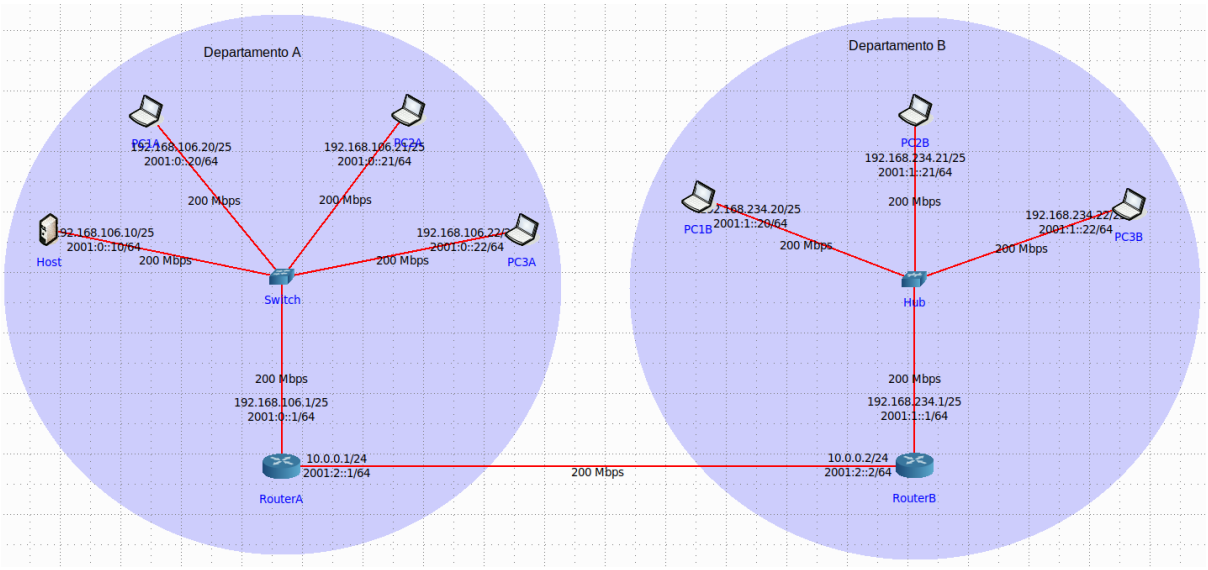
O endereço MAC, ou endereço Ethernet, do destino é `00:d0:03:ff:94:00`. O sistema a que este dispositivo corresponde é a Intel.

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.*

Foi recebido um pacote com protocolo IPv4 (network layer), o protocolo TCP (transport layer), o TLS, implementado entre a application layer e a transport layer (estabelece uma ligação segura), e o Application Data Protocol (application layer).

```
▾ Internet Protocol Version 4, Src: 172.26.53.255, Dst: 193.137.9.171
▾ Transmission Control Protocol, Src Port: 50682, Dst Port: 443, Seq: 2052, Ack: 7843, Len: 678
```

Parte 2



Nota: poderia ser usada uma máscara /24 (0 a 255), visto que não há necessidade para subnetting.

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

```
root@PC1A:/tmp/pycore.33753/PC1A.conf# ping -c 10 192.168.234.20
PING 192.168.234.20 (192.168.234.20) 56(84) bytes of data.
64 bytes from 192.168.234.20: icmp_seq=1 ttl=62 time=1.06 ms
64 bytes from 192.168.234.20: icmp_seq=2 ttl=62 time=0.349 ms
64 bytes from 192.168.234.20: icmp_seq=3 ttl=62 time=0.344 ms
64 bytes from 192.168.234.20: icmp_seq=4 ttl=62 time=0.290 ms
64 bytes from 192.168.234.20: icmp_seq=5 ttl=62 time=13.9 ms
^C
--- 192.168.234.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4082ms
rtt min/avg/max/mdev = 0.290/3.188/13.894/5.360 ms
root@PC1A:/tmp/pycore.33753/PC1A.conf#
```

> Ping do PC1A para o PC1B.

```
rtt min/avg/max/mdev = 0.290/3.188/13.894/5.360 ms
root@PC1A:/tmp/pycore.33753/PC1A.conf# ping 192.168.234.21
PING 192.168.234.21 (192.168.234.21) 56(84) bytes of data.
64 bytes from 192.168.234.21: icmp_seq=1 ttl=62 time=0.580 ms
64 bytes from 192.168.234.21: icmp_seq=2 ttl=62 time=0.178 ms
64 bytes from 192.168.234.21: icmp_seq=3 ttl=62 time=0.294 ms
64 bytes from 192.168.234.21: icmp_seq=4 ttl=62 time=0.271 ms
64 bytes from 192.168.234.21: icmp_seq=5 ttl=62 time=9.39 ms
64 bytes from 192.168.234.21: icmp_seq=6 ttl=62 time=0.187 ms
^C
--- 192.168.234.21 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5096ms
rtt min/avg/max/mdev = 0.178/1.816/9.386/3.388 ms
root@PC1A:/tmp/pycore.33753/PC1A.conf#
```

> Ping to PC1A para o PC2B

```
root@PC1A:/tmp/pycore.33753/PC1A.conf# arp -a
? (192.168.106.1) at 00:00:00:aa:00:00 [ether] on eth0
root@PC1A:/tmp/pycore.33753/PC1A.conf#
```

> Conteúdo da tabela arp do PC1A

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

O endereço 192.168.106.1 corresponde à interface de entrada do router.
Este endereço esta mapeado para o endereço MAC "00:00:00:aa:00:00".
Quando um dispositivo precisa de comunicar com outro dispositivo numa rede, este consulta a sua tabela ARP para encontrar o endereço MAC associado ao endereço IP do dispositivo de destino. Se o endereço MAC não estiver na tabela ARP, o dispositivo envia um pacote de solicitação ARP para a rede, perguntando qual é o endereço MAC associado ao endereço IP de destino. O dispositivo de destino responde com seu endereço MAC, que é então armazenado na tabela ARP do dispositivo que fez a solicitação.
Para além disso, este endereço está associado à conexão ethernet eth0.

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

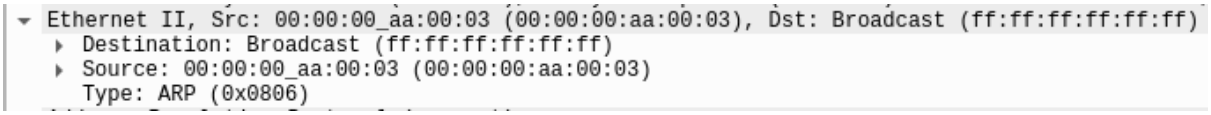
Neste caso, o equipamento que pode apresentar a maior tabela ARP em termos de número de entradas é o RouterB, uma vez que este tem os endereços e MAC addresses associados ao PC1B, ao PC2B e ao RouterA. (O Hub não gere MAC addresses).

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

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

No.	Time	Source	Destination	Protocol	Length	Info
43	37.243104129	192.168.106.20	192.168.234.20	ICMP	98	Echo (ping) request id=0x001b, seq=1/256, ttl=64 (reply in 4…
44	37.243594476	192.168.234.20	192.168.106.20	ICMP	98	Echo (ping) reply id=0x001b, seq=1/256, ttl=62 (request in…
53	44.909307108	192.168.106.20	192.168.234.21	ICMP	98	Echo (ping) request id=0x001c, seq=1/256, ttl=64 (reply in 5…
54	44.911883040	192.168.234.21	192.168.106.20	ICMP	98	Echo (ping) reply id=0x001c, seq=1/256, ttl=62 (request in…

No.	Time	Source	Destination	Protocol	Length	Info
41	37.242866776	00:00:00_aa:00:03	Broadcast	ARP	42	Who has 192.168.106.1? Tell 192.168.106.20
42	37.243099109	00:00:00_aa:00:00	00:00:00_aa:00:03	ARP	42	192.168.106.1 is at 00:00:00:aa:00:00
48	42.491966583	00:00:00_aa:00:00	00:00:00_aa:00:03	ARP	42	Who has 192.168.106.20? Tell 192.168.106.1
49	42.491978196	00:00:00_aa:00:03	00:00:00_aa:00:00	ARP	42	192.168.106.20 is at 00:00:00:aa:00:03



O valor do endereço MAC de origem é `00:00:00:aa:00:03`, o endereço correspondente ao dispositivo que manda a request, e o endereço MAC de destino é `ff:ff:ff:ff:ff:ff` que corresponde a Broadcast, ou seja, todos os dispositivos na rede irão receber esta request: "*Who has 192.168.106.1? Tell 192.168.106.20*".

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

O valor hexadecimal do campo Type da trama Ethernet é `0x0806` e indica que se trata de um pacote ARP (Address Resolution Protocol).

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

Uma forma de se saber que se trata de um pedido ARP é verificando se a trama possui o campo Type com valor `0x0806`, como se referiu anteriormente. Outra forma é, por exemplo, ver se o pacote possui o header Address Resolution Protocol, ou se os seus endereços de origem de origem e destino são endereços físicos / MAC addresses (apesar de que outros tipos de protocolos também poderiam utilizar endereços físicos como origem e destino).

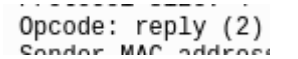
d. *Explicite, em linguagem comum, que tipo de pedido ou pergunta é feita pelo host de origem à rede?*

O host de origem (PC1A) pergunta quem possui o endereço IPv4 192.168.106.1 (RouterA) à rede, de modo a saber o seu endereço MAC (físico), e a resposta deve voltar para o endereço 192.168.106.1 (PC1A, a origem com endereço IPv4).

3. Localize a mensagem ARP que é a resposta ao pedido ARP efetuado.

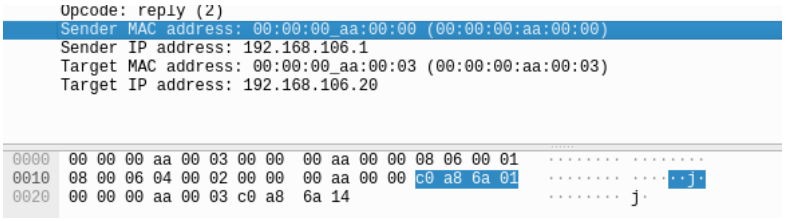
a. *Qual o valor do campo ARP opcode? O que especifica?*

O valor do campo opcode (bytes `00 02`) específica que se trata de uma reply.



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

O sender MAC address (a resposta) está no header do Address Resolution Protocol. "192.168.106.1 is at `00:00:00:00:aa:00:00`" -> Target: PC1A (198.162.106.20)



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.*

No PCA1, o endereço MAC associado é `00:00:00:aa:00:03`:

```
root@PC1A:/tmp/pycore.33753/PC1A.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.106.20  netmask 255.255.255.128  broadcast 0.0.0.0
        inet6 fe80::200:ff:feaa:3  prefixlen 64  scopeid 0x20<link>
        inet6 2001::20  prefixlen 64  scopeid 0x0<global>
        ether 00:00:00:aa:00:03  txqueuelen 1000  (Ethernet)
        RX packets 4626  bytes 372600 (372.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 1732 (1.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

A sua rota default é o RouterA (192.168.106.1):

```
root@PC1A:/tmp/pycore.33753/PC1A.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.106.1   0.0.0.0         UG        0 0          0 eth0
192.168.106.0   0.0.0.0         255.255.255.128 U         0 0          0 eth0
```

No RouterA, o endereço MAC associado é `00:00:00:aa:00:00`:

```
root@RouterA:/tmp/pycore.33753/RouterA.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.106.1  netmask 255.255.255.128  broadcast 0.0.0.0
        inet6 2001::1  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::200:ff:feaa:0  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 151  bytes 14354 (14.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4588  bytes 367336 (367.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Assim, a tabela ARP do PC1A possui uma entrada correspondente ao RouterA (a sua rota default):

```
root@PC1A:/tmp/pycore.33753/PC1A.conf# arp -a
? (192.168.106.1) at 00:00:00:aa:00:00 [ether] on eth0
root@PC1A:/tmp/pycore.33753/PC1A.conf#
```

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

O modo de comunicação usado no envio da resposta/ reply ARP é unicast.
Visto que o RouterA já sabe o endereço IPv4 e endereço MAC do dispositivo que fez a request, PC1A, pois está presente na trama ARP do pacote de request, não há necessidade de enviar a reply para todos os dispositivos na rede (broadcast).

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

O ping feito ao segundo PC (PC2B) originou novamente quatro pacotes ARP:

```
  57 71.060624690  00:00:00_aa:00:03    00:00:00_aa:00:00    ARP         42 Who has 192.168.106.1? Tell 192.168.106.20
  58 71.060716388  00:00:00_aa:00:00    00:00:00_aa:00:03    ARP         42 Who has 192.168.106.20? Tell 192.168.106.1
  59 71.060724143  00:00:00_aa:00:03    00:00:00_aa:00:00    ARP         42 192.168.106.20 is at 00:00:00:aa:00:03
  60 71.060768048  00:00:00_aa:00:00    00:00:00_aa:00:03    ARP         42 192.168.106.1 is at 00:00:00:aa:00:00
```

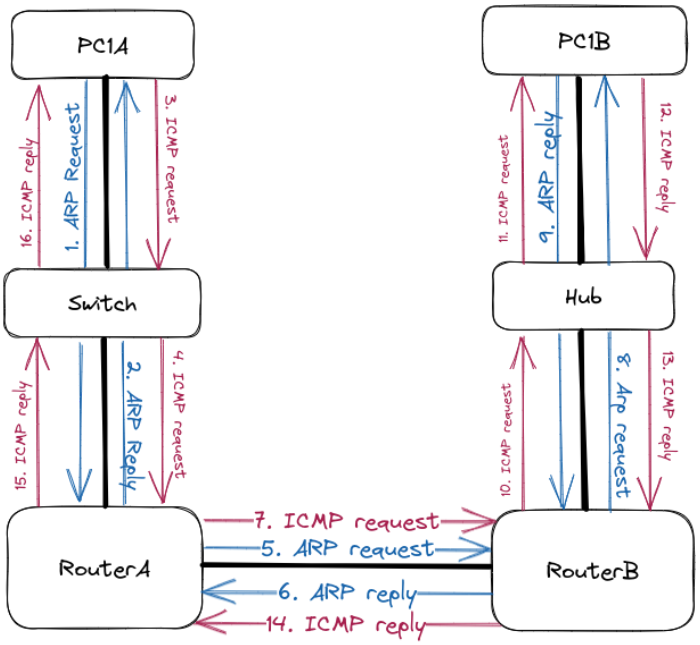
No entanto, como o RouterA já está na tabela ARP do PC1A, este não precisa de enviar uma request broadcast, para todos na rede, em vez disso manda diretamente para o endereço MAC que guardou. (últimos dois pacotes devido a [delay_first_probe_time])

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.

```
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
```

Neste caso, o tipo de hardware (ligação lógica) para comunicação usado é Ethernet, ao qual corresponde um MAC address de 6 bytes. O tipo de protocolo utilizado para comunicação (network layer) é o IPv4 que corresponde a um endereço IP de 4 bytes.

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.

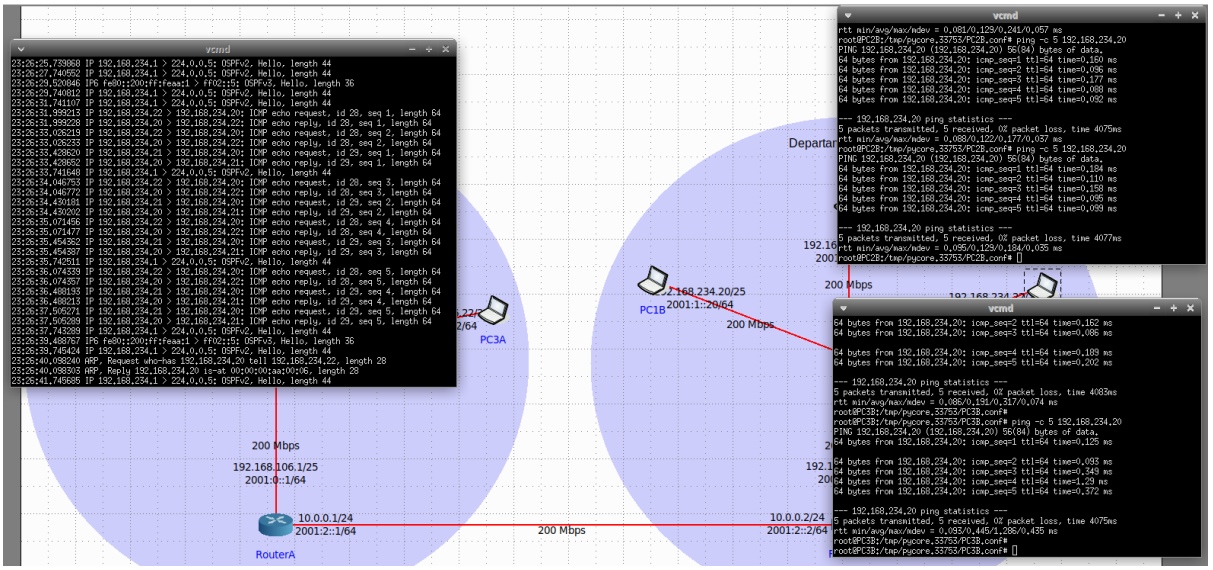


195	13.969183026	00:00:00_aa:00:03		ARP	44	Who has 192.168.106.1? Tell 192.168.106.20
196	13.969183026	00:00:00_aa:00:03		ARP	44	Who has 192.168.106.1? Tell 192.168.106.20
197	13.969265698	00:00:00_aa:00:03		ARP	44	Who has 192.168.106.1? Tell 192.168.106.20
198	13.969268273	00:00:00_aa:00:03		ARP	44	Who has 192.168.106.1? Tell 192.168.106.20
199	13.969270256	00:00:00_aa:00:03		ARP	44	Who has 192.168.106.1? Tell 192.168.106.20
200	13.969272210	00:00:00_aa:00:03		ARP	44	Who has 192.168.106.1? Tell 192.168.106.20
201	13.969307048	00:00:00_aa:00:00		ARP	44	192.168.106.1 is at 00:00:00:aa:00:00
202	13.969392676	00:00:00_aa:00:00		ARP	44	192.168.106.1 is at 00:00:00:aa:00:00
203	13.969412635	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=64 (no respons…
204	13.969509404	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=64 (no respons…
205	13.969536307	00:00:00_aa:00:09		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
206	13.969536307	00:00:00_aa:00:09		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
207	13.969627235	00:00:00_aa:00:09		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
208	13.969639699	00:00:00_aa:00:0a		ARP	44	10.0.0.2 is at 00:00:00:aa:00:0a
209	13.969685399	00:00:00_aa:00:0a		ARP	44	10.0.0.2 is at 00:00:00:aa:00:0a
210	13.969691030	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=63 (no respons…
211	13.969744444	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=63 (no respons…
212	13.969756217	00:00:00_aa:00:01		ARP	44	Who has 192.168.234.20? Tell 192.168.234.1
213	13.969756217	00:00:00_aa:00:01		ARP	44	Who has 192.168.234.20? Tell 192.168.234.1
214	13.969805113	00:00:00_aa:00:01		ARP	44	Who has 192.168.234.20? Tell 192.168.234.1
215	13.969807608	00:00:00_aa:00:01		ARP	44	Who has 192.168.234.20? Tell 192.168.234.1
216	13.969809451	00:00:00_aa:00:01		ARP	44	Who has 192.168.234.20? Tell 192.168.234.1
217	13.969838298	00:00:00_aa:00:06		ARP	44	192.168.234.20 is at 00:00:00:aa:00:06
218	13.969869780	00:00:00_aa:00:06		ARP	44	192.168.234.20 is at 00:00:00:aa:00:06
219	13.969871493	00:00:00_aa:00:06		ARP	44	192.168.234.20 is at 00:00:00:aa:00:06
220	13.969872675	00:00:00_aa:00:06		ARP	44	192.168.234.20 is at 00:00:00:aa:00:06
221	13.969876813	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=62 (no respons…
222	13.969941239	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=62 (no respons…
223	13.969943744	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=62 (no respons…
224	13.969945187	192.168.106.20	192.168.234.20	ICMP	100	Echo (ping) request id=0x001b, seq=1/256, ttl=62 (reply in 2…
225	13.969974314	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=64 (request in…
226	13.970060893	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=64
227	13.970063158	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=64
228	13.970064501	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=64
229	13.970084029	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=63
230	13.970103998	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=63
231	13.970110571	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=62
232	13.970178403	192.168.234.20	192.168.106.20	ICMP	100	Echo (ping) reply id=0x001b, seq=1/256, ttl=62

Parte 3

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.

`tcpdump` no PCB1; comando ping para PCB1 no PCB2 e PCB3.



(não foi possível verificar colisões)

Um hub não consegue impedir colisões, visto que não trabalha a nível de endereços MAC. Quando existem colisões, geralmente existe um período de *backoff* para evitar colisões contínuas.
Por outro lado, os switches conseguem evitar colisões, pois ao contrário dos hubs que enviam as transmissões recebidas para todas as portas (como foi possível verificar na última leitura do wireshark - imagem no fim da parte 2), os switches utilizam tabelas de endereços MAC para direcionar o tráfego apenas para a porta de destino correta.

Tabela de comutação do Switch do Departamento A

Dispositivo	Endereço Mac	Porta
RouterA	00:00:00:aa:00:00	1
Host	00:00:00:aa:00:02	2
PC1A	00:00:00:aa:00:03	3
PC2A	00:00:00:aa:00:04	4
PC3A	00:00:00:aa:00:05	5