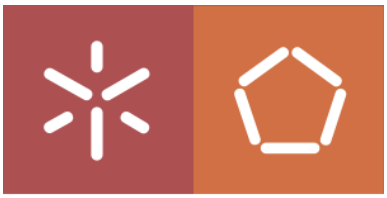


Universidade do Minho
Escola de Engenharia

DEZEMBRO 2020

TP3

REDES DE COMPUTADORES

Grupo 54

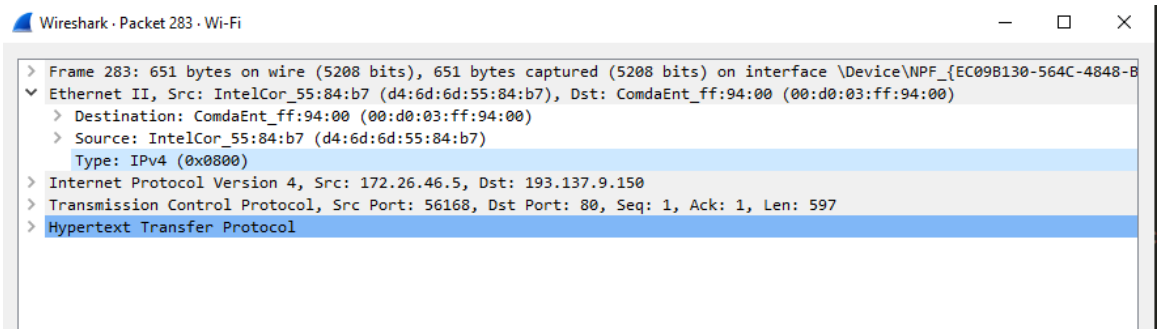
Adriano Maior, a89483

Joel Martins, a89575

Manuel Moreira, a89471

3. Captura e análise de Tramas Ethernet

1. Anote os endereços MAC de origem e de destino da trama capturada.



Endereço de Origem: d4:6d:6d:55:84:b7

Endereço de Destino: 00:d0:03:ff:94:00

2. Identifique a que sistemas se referem. Justifique.

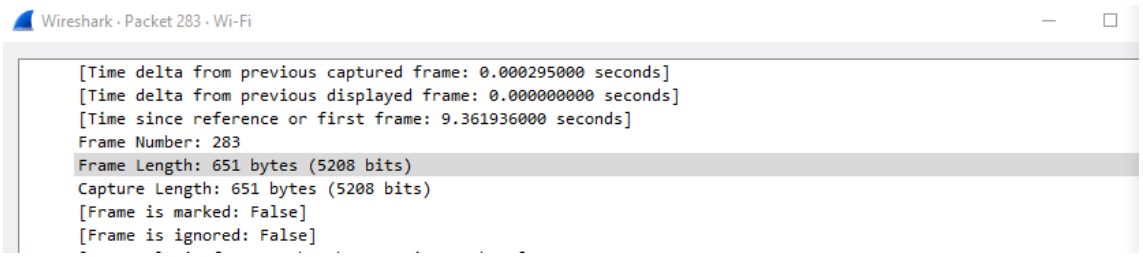
O endereço de origem é o endereço da interface da nossa máquina e o endereço de destino é o endereço da interface do router local.

O endereço de origem identifica o local de onde é enviada a trama o que significa que esse endereço vai representar a interface da nossa máquina. Como a nossa máquina não reconhece endereços fora da rede local é definido como endereço destino a interface do router da rele local, que posteriormente irá tratar a trama recebida.

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

Como se verifica na figura 1, 0x0800 simboliza que a trama encapsula um pacote encapsulado em IPv4.

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.



```
Internet Protocol Version 4, Src: 172.26.46.5, Dst: 193.137.9.150
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
```

O TCP header tem um tamanho de 20 bytes. O IPv4 header tem um tamanho de 20 bytes. O endereço MAC origem tem um tamanho de 6 bytes, tal como o endereço MAC destino. O Type tem um tamanho de 2 bytes.

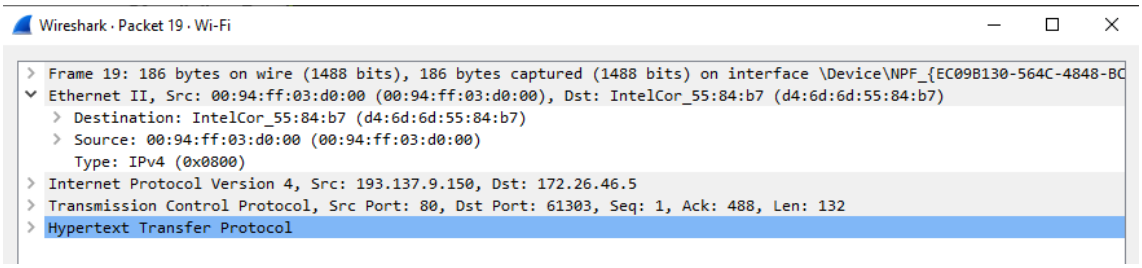
Ora se o tamanho do frame é 651 bytes, temos um overload de 54 bytes.

Logo 54/651 * 100 = 8,29%.

5. Através de visualização direta ou construindo um filtro específico, verifique se foram detetadas tramas com erros (por verificação do campo FCS (Frame Check Sequence)).

O campo FCS (Frame Check Sequence) não aparece na trama capturada porque as redes wired (como a ethernet) são muito robustas e suscetíveis a erros. Tal não acontece com as redes Wireless, pois pelo contrário, são muito suscetíveis a erros.

A seguir responda às seguintes perguntas, baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP.



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

O MAC origem: **00:d0:03:ff:94:00** refere-se à interface do router rede local.

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

O MAC destino: **d4:6d:6d:55:84:b7** refere-se à interface da nossa máquina.

8. Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida.

As camadas protocolares são: Ethernet,IPv4,TCP,HTTP

4. Protocolo ARP

9. Observe o conteúdo da tabela ARP. Diga o que significa cada uma das colunas.

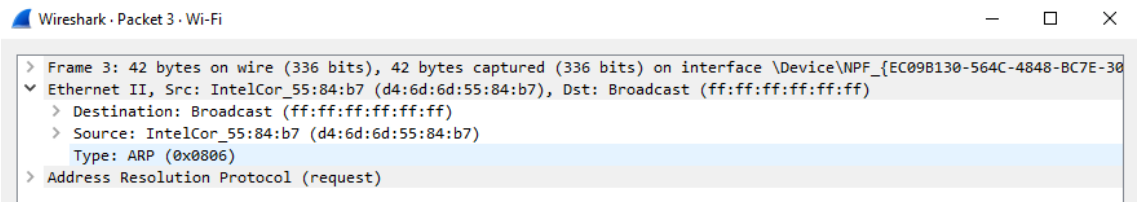
A primeira coluna representa o endereço IP do host. A segunda coluna representa o endereço físico, MAC address, e a terceira coluna representa o seu tipo, ou seja, se é estático ou dinâmico.

```
C:\Users\Adriano>arp -a

Interface: 192.168.56.1 --- 0xf
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  239.255.255.253       01-00-5e-7f-ff-fd     static

Interface: 172.26.46.5 --- 0x13
  Internet Address      Physical Address      Type
  172.26.254.254        00-d0-03-ff-94-00     dynamic
  172.26.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  239.255.255.253       01-00-5e-7f-ff-fd     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

10. 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?

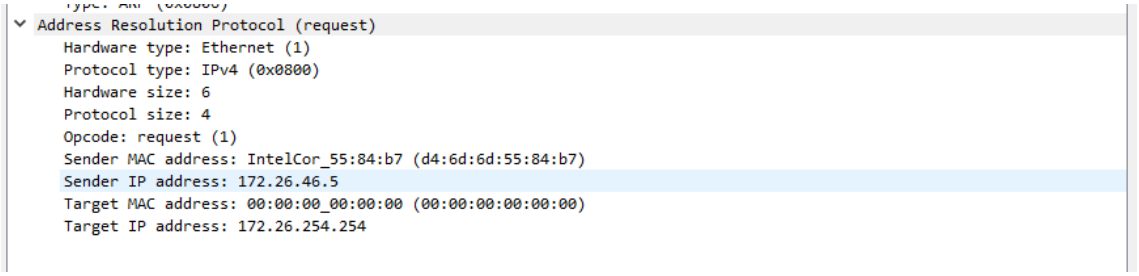


A origem é d4:6d:6d:55:84:b7 e o destino é ff:ff:ff:ff:ff:ff. Como é necessário conhecer o MAC associado ao IP do próximo salto, temos que enviar um pedido ARP a todos os dispositivos da rede local para saber quem tem esse IP.

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

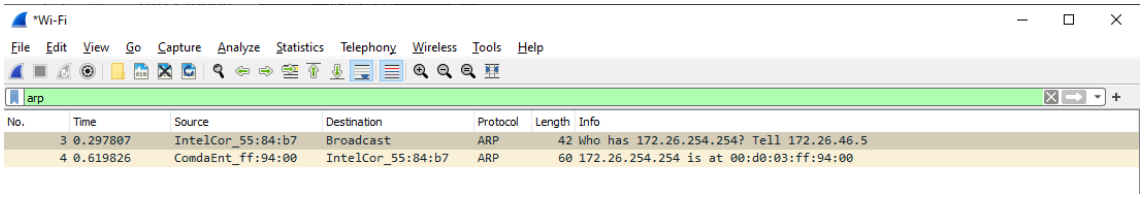
O valor é 0x0806 e indica o ARP.

12. Como pode confirmar que se trata efetivamente de um pedido ARP? Identifique que tipo de endereços estão contidos na mensagem ARP? Que conclui? (Se necessário, consulte a RFC do protocolo ARP http://tools.ietf.org/html/rfc826.html.).



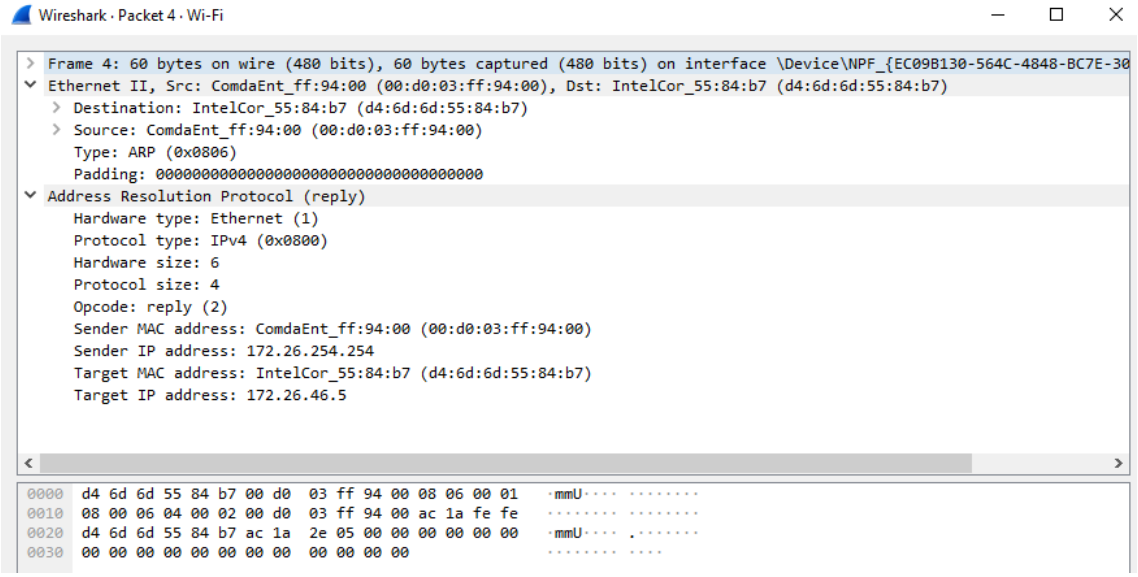
Como se pode ver na figura, esta indica que se trata de um pedido ARP. Os tipos de endereços presentes na mensagem ARP são os endereços MAC (Sender MAC address e Target MAC address) e IP (Sender IP Address e Target IP address). Podemos concluir que se utilizam os endereços MAC para descobrir que dispositivo tem um determinado IP.

13. Explicite que tipo de pedido ou pergunta é feita pelo host de origem?



O host pergunta quem tem o IP do gateway (172.26.254.254) e pede para enviar a resposta para 172.26.46.5.

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



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

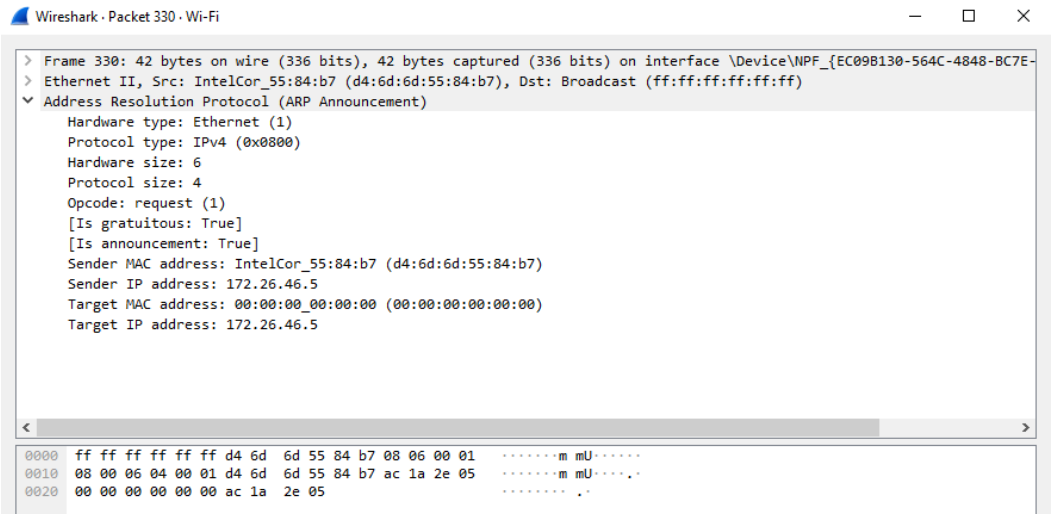
Opcode: reply (2) e especifica que é uma resposta ao *request* anterior.

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

A resposta ao pedido ARP encontra-se no Sender MAC address.

5. ARP Gratuito

15. Identifique um pacote de pedido ARP gratuito originado pelo seu sistema. 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?



Um pedido ARP gratuito pode ser identificado quando o Sender IP address é igual ao Target IP address.

6. Domínios de colisão

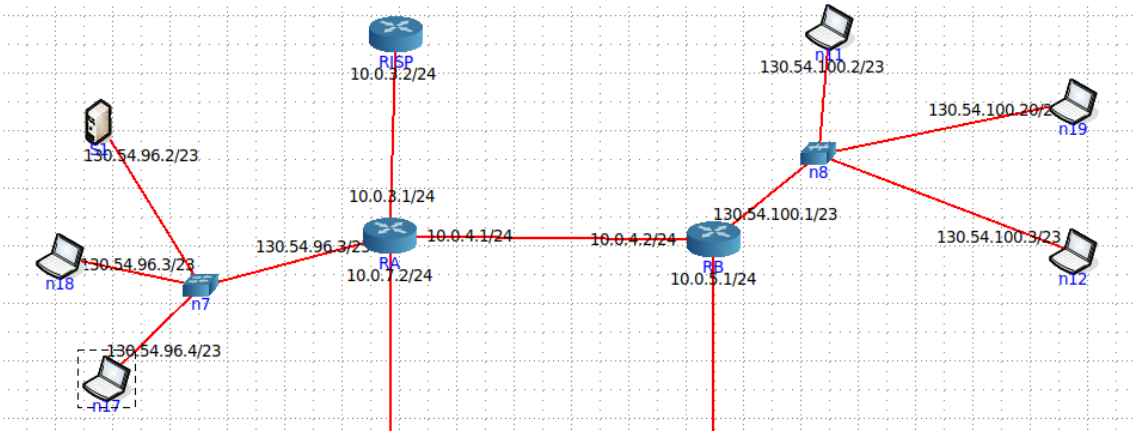
16. Através da opção tcpdump verifique e compare como flui o tráfego nas diversas interfaces dos vários dispositivos no departamento A (LAN comutada) e no departamento B (LAN partilhada) quando gera tráfego intra-departamento (por exemplo, através do comando *ping*). Que conclui?

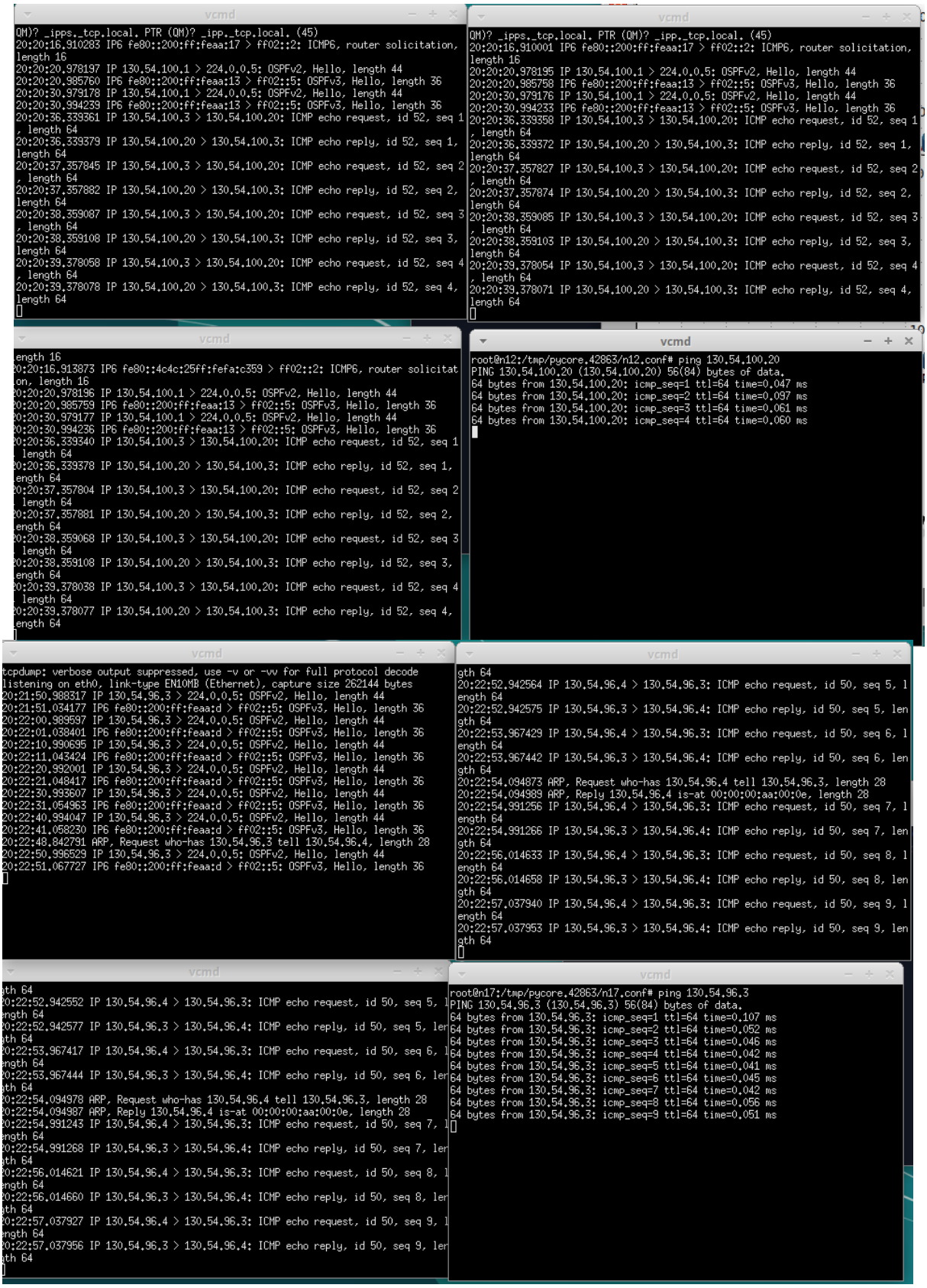
Comente os resultados obtidos quanto à utilização de *hubs* e *switches* no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.

Como podemos ver, no caso de um hub o tráfego que circula na rede local é recebido por todos os dispositivos que nela se encontram, enquanto que num switch apenas os dispositivos que estão envolvidos na comunicação recebem os dados.

Assim sendo num contexto geral, um switch pode ser utilizado para enviar encaminhar os dados que se pretende para utilizadores específicos dentro de uma rede local enquanto que hubs inundam a rede com a informação que lhes chega.

(Nota: Foi necessário adicionar um pc extra no departamento B de modo a que fosse




vcmd
QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
20:20:16.910283 IP6 fe80::200:ff:feaa:17 > ff02::2: ICMP6, router solicitation, length 16
20:20:20.978197 IP 130.54.100.1 > 224.0.0.5: OSPFv2, Hello, length 44
20:20:20.985760 IP6 fe80::200:ff:feaa:13 > ff02::5: OSPFv3, Hello, length 36
20:20:30.979178 IP 130.54.100.1 > 224.0.0.5: OSPFv2, Hello, length 44
20:20:30.994239 IP6 fe80::200:ff:feaa:13 > ff02::5: OSPFv3, Hello, length 36
20:20:36.339361 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 1, length 64
20:20:36.339379 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 1, length 64
20:20:37.357845 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 2, length 64
20:20:37.357882 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 2, length 64
20:20:38.359087 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 3, length 64
20:20:38.359108 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 3, length 64
20:20:39.378058 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 4, length 64
20:20:39.378078 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 4, length 64
vcmd
QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
20:20:16.910001 IP6 fe80::200:ff:feaa:17 > ff02::2: ICMP6, router solicitation, length 16
20:20:20.978195 IP 130.54.100.1 > 224.0.0.5: OSPFv2, Hello, length 44
20:20:20.985758 IP6 fe80::200:ff:feaa:13 > ff02::5: OSPFv3, Hello, length 36
20:20:30.979176 IP 130.54.100.1 > 224.0.0.5: OSPFv2, Hello, length 44
20:20:30.994233 IP6 fe80::200:ff:feaa:13 > ff02::5: OSPFv3, Hello, length 36
20:20:36.339358 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 1, length 64
20:20:36.339372 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 1, length 64
20:20:37.357827 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 2, length 64
20:20:37.357874 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 2, length 64
20:20:38.359085 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 3, length 64
20:20:38.359103 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 3, length 64
20:20:39.378054 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 4, length 64
20:20:39.378071 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 4, length 64
vcmd
ength 16
20:20:16.913873 IP6 fe80::4c4c:25ff:fefa:c359 > ff02::2: ICMP6, router solicitation, length 16
20:20:20.978196 IP 130.54.100.1 > 224.0.0.5: OSPFv2, Hello, length 44
20:20:20.985759 IP6 fe80::200:ff:feaa:13 > ff02::5: OSPFv3, Hello, length 36
20:20:30.979177 IP 130.54.100.1 > 224.0.0.5: OSPFv2, Hello, length 44
20:20:30.994236 IP6 fe80::200:ff:feaa:13 > ff02::5: OSPFv3, Hello, length 36
20:20:36.339340 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 1, length 64
20:20:36.339378 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 1, length 64
20:20:37.357804 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 2, length 64
20:20:37.357881 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 2, length 64
20:20:38.359068 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 3, length 64
20:20:38.359108 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 3, length 64
20:20:39.378038 IP 130.54.100.3 > 130.54.100.20: ICMP echo request, id 52, seq 4, length 64
20:20:39.378077 IP 130.54.100.20 > 130.54.100.3: ICMP echo reply, id 52, seq 4, length 64
vcmd
root@n12:/tmp/pycore.42863/n12.conf# ping 130.54.100.20
PING 130.54.100.20 (130.54.100.20) 56(84) bytes of data.
64 bytes from 130.54.100.20: icmp_seq=1 ttl=64 time=0.047 ms
64 bytes from 130.54.100.20: icmp_seq=2 ttl=64 time=0.097 ms
64 bytes from 130.54.100.20: icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from 130.54.100.20: icmp_seq=4 ttl=64 time=0.060 ms
vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:21:50.988317 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:21:51.034177 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
20:22:00.989597 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:22:01.038401 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
20:22:10.990695 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:22:11.043424 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
20:22:20.992001 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:22:21.048417 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
20:22:30.993607 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:22:31.054963 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
20:22:40.994047 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:22:41.058230 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
20:22:48.842791 ARP, Request who-has 130.54.96.3 tell 130.54.96.4, length 28
20:22:50.996529 IP 130.54.96.3 > 224.0.0.5: OSPFv2, Hello, length 44
20:22:51.067727 IP6 fe80::200:ff:feaa:d > ff02::5: OSPFv3, Hello, length 36
vcmd
gth 64
20:22:52.942564 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 5, length 64
20:22:52.942575 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 5, length 64
20:22:53.967429 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 6, length 64
20:22:53.967442 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 6, length 64
20:22:54.094873 ARP, Request who-has 130.54.96.4 tell 130.54.96.3, length 28
20:22:54.094989 ARP, Reply 130.54.96.4 is-at 00:00:00:aa:00:0e, length 28
20:22:54.991256 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 7, length 64
20:22:54.991266 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 7, length 64
20:22:56.014633 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 8, length 64
20:22:56.014658 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 8, length 64
20:22:57.037940 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 9, length 64
20:22:57.037953 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 9, length 64
vcmd
th 64
20:22:52.942552 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 5, length 64
20:22:52.942577 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 5, length 64
20:22:53.967417 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 6, length 64
20:22:53.967444 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 6, length 64
20:22:54.094978 ARP, Request who-has 130.54.96.4 tell 130.54.96.3, length 28
20:22:54.094987 ARP, Reply 130.54.96.4 is-at 00:00:00:aa:00:0e, length 28
20:22:54.991243 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 7, length 64
20:22:54.991268 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 7, length 64
20:22:56.014621 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 8, length 64
20:22:56.014660 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 8, length 64
20:22:57.037927 IP 130.54.96.4 > 130.54.96.3: ICMP echo request, id 50, seq 9, length 64
20:22:57.037956 IP 130.54.96.3 > 130.54.96.4: ICMP echo reply, id 50, seq 9, length 64
vcmd
root@n17:/tmp/pycore.42863/n17.conf# ping 130.54.96.3
PING 130.54.96.3 (130.54.96.3) 56(84) bytes of data.
64 bytes from 130.54.96.3: icmp_seq=1 ttl=64 time=0.107 ms
64 bytes from 130.54.96.3: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 130.54.96.3: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 130.54.96.3: icmp_seq=4 ttl=64 time=0.042 ms
64 bytes from 130.54.96.3: icmp_seq=5 ttl=64 time=0.041 ms
64 bytes from 130.54.96.3: icmp_seq=6 ttl=64 time=0.045 ms
64 bytes from 130.54.96.3: icmp_seq=7 ttl=64 time=0.042 ms
64 bytes from 130.54.96.3: icmp_seq=8 ttl=64 time=0.056 ms
64 bytes from 130.54.96.3: icmp_seq=9 ttl=64 time=0.051 ms

Conclusão

Este trabalho permitiu consolidar os conhecimentos que foram obtidos nas aulas teóricas.

Efetivamente, a análise de tramas Ethernet proporcionou uma oportunidade para que conseguíssemos compreender melhor ao funcionamento destas redes e da camada de ligação em si. Um ponto que foi particularmente interessante, foi ver pacotes que tinham mensagens ARP Gratuitas, uma vez que percebemos a maneira como as interfaces lidam com endereços IPs repetidos. Por fim, a comparação entre hubs e switches levou a que compreendêssemos a diferença entre estes dispositivos que ainda não era evidente.

Assim sendo, achamos que este trabalho foi produtivo e que alcançamos os objetivos pretendidos.