Universidade do Minho
Braga, Portugal

Redes de Computadores TP3:

20th November
Grupo 4

3)

```
No.     Time           Source                Destination           Protocol Length Info
     85 8.513399085    192.168.100.163       193.136.19.40         HTTP     846    GET / HTTP/1.1
Frame 85: 846 bytes on wire (6768 bits), 846 bytes captured (6768 bits) on interface 0
Ethernet II, Src: AsustekC_02:28:66 (2c:56:dc:02:28:66), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    Destination: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    Source: AsustekC_02:28:66 (2c:56:dc:02:28:66)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.100.163, Dst: 193.136.19.40
Transmission Control Protocol, Src Port: 40902, Dst Port: 80, Seq: 1, Ack: 1, Len: 780
Hypertext Transfer Protocol
```

.1

MAC Origem: 2c-56-dc-02-28-66 MAC Destino: 00-0c-29-d2-19-f0

.2

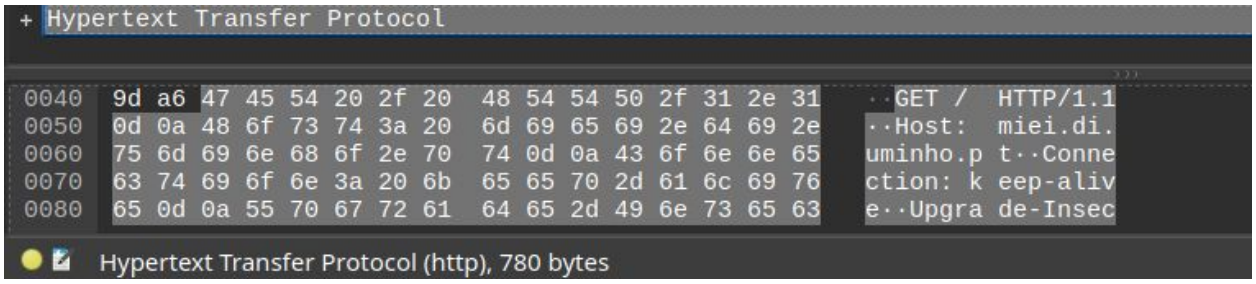
2c-56-dc-02-28-66 refere-se ao endereço estático da NIC do meu portátil.

00-0c-29-d2-19-f0 refere-se ao endereço estático da NIC do servidor miei.

Pois para percorrer os nós precisa do endereço MAC destino para se guiar e origem para responder.

.3

0x0800 indica o protocolo da camada superior que deve ser utilizado, neste caso, IPV4.



.4

São usados 66 bytes até ao character ASCII “G”.

packet = 846 bytes

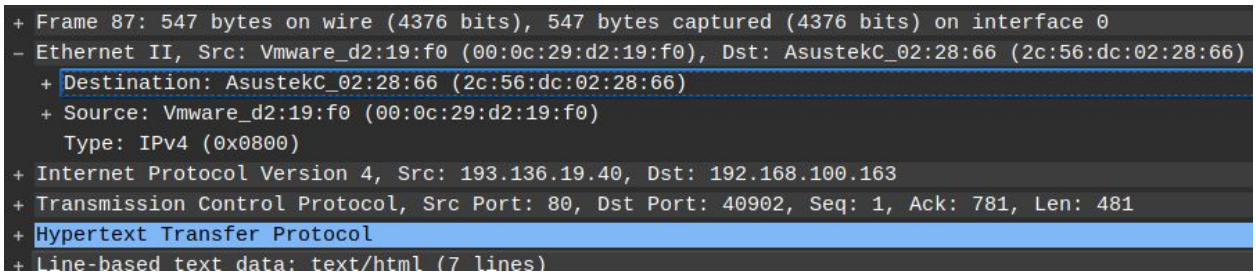
payload = 780 bytes

Header size = packet - payload + 4 = 70 bytes

Overhead% = (70*100)/780 = 8.97%

.5

Devido a estar conectado por cabo de rede e o pacote Get ser de tamanho reduzido, a probabilidade de erro é extremamente pequena.



.6

Source: Vmware_d2:19:f0. Corresponde ao servidor de http://miei.di.uminho.pt pois é uma resposta ao Get enviado pelo nosso PC.

.7

Destination:(2c:56:dc:02:28:66). Corresponde ao nosso PC

.8

Protocolos: IPv4, TCP e HTTP

.9

```
λ  ~  arp -a
gw.sa.di.uminho.pt (192.168.100.254) at 00:0c:29:d2:19:f0 [ether] on enp0s20f0u3u3
_gateway (172.26.254.254) at 00:d0:03:ff:94:00 [ether] on wlp2s0
λ  ~  ping 192.168.100197
ping: 192.168.100197: Name or service not known
λ  ~  ping 192.168.100.197
PING 192.168.100.197 (192.168.100.197) 56(84) bytes of data.
64 bytes from 192.168.100.197: icmp_seq=1 ttl=64 time=3.45 ms
64 bytes from 192.168.100.197: icmp_seq=2 ttl=64 time=0.473 ms
64 bytes from 192.168.100.197: icmp_seq=3 ttl=64 time=1.73 ms
^C
--- 192.168.100.197 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.473/1.882/3.445/1.218 ms
λ  ~  arp -a
pedro.local (192.168.100.197) at 88:d7:f6:2c:42:59 [ether] on enp0s20f0u3u3
gw.sa.di.uminho.pt (192.168.100.254) at 00:0c:29:d2:19:f0 [ether] on enp0s20f0u3u3
_gateway (172.26.254.254) at 00:d0:03:ff:94:00 [ether] on wlp2s0
λ  ~  arp
Address                  HWtype  HWaddress           Flags Mask            Iface
pedro.local              ether   88:d7:f6:2c:42:59   C                     enp0s20f0u3u3
gw.sa.di.uminho.pt       ether   00:0c:29:d2:19:f0   C                     enp0s20f0u3u3
_gateway                 ether   00:d0:03:ff:94:00   C                     wlp2s0
λ  ~
```

Podemos observar 5 colunas e uma linha

- “Address” refere se ao endereço da entry.

- “HWtype” (Hardware Type) indica o tipo de hardware que está a ser identificado pelo ARP, neste caso como é ether significa que está a ser usado Ethernet .
- Flags com C significa que é uma complete entry no ARP.
- “HWaddress” significa “Hardware Address” e é o endereço físico de quem está conectado.
- “Iface” é a interface do endereço.

.10

Destination:(ff:ff:ff:ff:ff:ff), Source:(9c:eb:e8:07:8b:e5). Este destino significa que todos os devices conectados nesta Lan vão receber esta mensagem

.11

(0x0806, identifica o protocolo de encapsulamento, nest caso, ARP.

.12

Opcode = 00 01, indica que é um request.

.13

Endereços MAC e IP.

.14

ARP Gratuito.

.15

a.

Opcode = 00 02, indica que é uma reply.

b.

Posiçao byte 20-21.

.16

O pacote de pedido ARP apresenta apenas 42 bytes, em vez dos 60 bytes esperados.

.17

Faz vários router e neighbor solicitations para identificar a rota.

```
15:35:12.239524 IP6 fe80::48b0:b5ff:fe80:1fc4 > ff02::1:ff00:0: ICMP6, neighbor solicitati
on, who has ::, length 32
15:35:12.239575 IP6 fe80::200:ff:feaa:2 > ff02::1:ff00:0: ICMP6, neighbor solicitation, wh
o has ::, length 32
15:35:12.239612 IP6 fe80::200:ff:feaa:1 > ff02::1:ff00:0: ICMP6, neighbor solicitation, wh
o has ::, length 32
15:35:13.262545 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 197, seq 1, length 64
15:35:13.262659 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 197, seq 1, length 64
15:35:14.265643 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 197, seq 2, length 64
15:35:14.265746 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 197, seq 2, length 64
15:35:15.279331 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 197, seq 3, length 64
15:35:15.279372 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 197, seq 3, length 64
```

.18

Usa ARP para mapear a rede.

```
15:42:13.786000 IP6 fe80::48b0:b5ff:fe80:1fc4 > ff02::2: ICMP6, router solicitation, lengt
h 16
15:42:14.158938 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 196, seq 6, length 64
15:42:14.159055 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 196, seq 6, length 64
15:42:14.212185 ARP, Request who-has 10.0.0.20 tell 10.0.0.10, length 28
15:42:14.212199 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:00, length 28
15:42:15.172206 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 196, seq 7, length 64
15:42:15.172253 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 196, seq 7, length 64
15:42:16.185524 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 196, seq 8, length 64
15:42:16.185550 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 196, seq 8, length 64
15:42:17.199333 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 196, seq 9, length 64
15:42:17.199357 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 196, seq 9, length 64
15:42:37.678968 IP6 fe80::200:ff:feaa:0 > ff02::2: ICMP6, router solicitation, length 16
15:42:39.385989 IP6 fe80::200:ff:feaa:1 > ff02::2: ICMP6, router solicitation, length 16
15:42:41.092222 IP6 fe80::4cce:39ff:fec7:f40c > ff02::2: ICMP6, router solicitation, lengt
h 16
```

Com este trabalho ganhamos experiência em análise de protocolos, MAC addresses, constituição de tramas e técnicas como ARP que ajudam a mapear a rede e reduzir colisões.

Em conclusão, este trabalho foi essencial para o nosso desenvolvimento e aprendizagem sobre pilhas protocolares e mapeamento de rede .