

Trabalho Prático 3

José Pedro Silva, José Ricardo Cunha, and Válder Carvalho

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a84577,a84302,a84464}@alunos.uminho.pt

Resumo É explicada a resolução dos exercícios de todas as partes do trabalho prático número 3. São abordados os conceitos do protocolo ARP, endereços MAC, camada de transporte, deteção e correção de erros, Ethernet e Wi-Fi. São analisadas tramas de Ethernet usando o *Wireshark* assim como o funcionamento de switches e hubs.

1 Introdução

Com este trabalho pretendemos mostrar a nossa interpretação dos problemas apresentados, assim como tentar explicitar o nosso raciocínio lógico, de modo a conseguir uma melhor visão da tecnologia em Redes, mergulhando nas ideias intrínsecas por base neste ramo científico.

Tentamos de forma concisa agrupar as ideias e definições num pequeno relatório que apenas contém comentários gerais às várias questões apresentadas, tentando detalhar ao máximo os conceitos mais relevantes para cada problema.

Nos **Exercícios da parte 3** (secção 2.1), abordamos tramas de Ethernet num nível mais baixo, utilizando o *Wireshark*.

Nos **Exercícios da parte 4** (secção 2.2), abordamos o protocolo ARP e as suas especificidades, utilizando principalmente a linha de comandos.

Nos **Exercícios da parte 5** (secção 2.3), abordamos o controlo de colisões e dispositivos físicos para esse propósito.

Todos os exercícios são extremamente importantes porque fazem ponte com as aulas teóricas assim como o trabalho prático anterior, isto é, servem para cimentar conhecimentos já adquiridos e que, certamente, são importantes na formação de futuros engenheiros, não só para este ramo em concreto mas sim como conhecimentos transversais.

2 Questões e Respostas

2.1 Exercícios parte 3

Usando o Wireshark e conectando ao website <http://miei.di.uminho.pt>, obtenha o número de ordem da sequência de bytes capturada correspondente à primeira mensagem de HTTP GET, selecionando a respetiva trama Ethernet.

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

+	4	0.707469548	10.0.2.15	193.136.19.40	HTTP	452 GET / HTTP/1.1
+	6	0.709201107	193.136.19.40	10.0.2.15	HTTP	535 HTTP/1.1 301 Moved Permanently (text/html)

```
▶ Frame 4: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface 0
▼ Ethernet II, Src: PcsCompu_7b:63:0c (08:00:27:7b:63:0c), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
  ▼ Destination: RealtekU_12:35:02 (52:54:00:12:35:02)
    Address: RealtekU_12:35:02 (52:54:00:12:35:02)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: PcsCompu_7b:63:0c (08:00:27:7b:63:0c)
    Address: PcsCompu_7b:63:0c (08:00:27:7b:63:0c)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.19.40
▶ Transmission Control Protocol, Src Port: 37036, Dst Port: 80, Seq: 1, Ack: 1, Len: 398
▶ Hypertext Transfer Protocol
```

Figura relativa à trama Ethernet do pacote HTTP GET.

Como podemos ver pela figura, os endereços MAC são:

- Origem - 08:00:27:7b:63:0c
 - Destino - 52:54:00:12:35:02
- 2) Identifique a que sistemas se referem. Justifique.
O endereço MAC de origem refere-se ao endereço físico da interface da placa de rede Ethernet do computador que utilizamos localmente.
O de destino refere-se ao endereço físico da interface de rede do router a que estamos conectados durante a aula prática.
 - 3) Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?
O valor hexadecimal é 0x0800, que representa o protocolo de IPv4 encapsulado nesta 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.

```

▶ Frame 4: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface 0
▼ Ethernet II, Src: PcsCompu_7b:63:0c (08:00:27:7b:63:0c), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
  ▼ Destination: RealtekU_12:35:02 (52:54:00:12:35:02)
    Address: RealtekU_12:35:02 (52:54:00:12:35:02)
    .....1 ..... = LG bit: Locally administered address (this is NOT the factory default)
    .....0 ..... = IG bit: Individual address (unicast)
  ▼ Source: PcsCompu_7b:63:0c (08:00:27:7b:63:0c)
    Address: PcsCompu_7b:63:0c (08:00:27:7b:63:0c)
    .....0 ..... = LG bit: Globally unique address (factory default)
    .....0 ..... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.19.40
▶ Transmission Control Protocol, Src Port: 37036, Dst Port: 80, Seq: 1, Ack: 1, Len: 398
▼ Hypertext Transfer Protocol
  ▼ GET / HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
      Host: mie1.di.uminho.pt\r\n
      User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
      Accept-Language: en-US,en;q=0.5\r\n
      0000 52 54 00 12 35 02 08 00 27 7b 63 0c 08 00 45 00 RT: 5... 'c...E
      0010 01 b6 95 60 40 00 40 06 c3 22 0a 00 02 0f c1 88 ...@- ".....
      0020 13 28 90 ac 00 50 e7 a5 ef d2 04 10 0a 02 50 18 .(...P- .....P
      0030 72 10 e2 67 00 00 47 45 54 20 2f 20 48 54 54 50 r..g..[E] / HTTP
      0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 6d 69 65 69 /1.1..Ho st: mie1
      0050 2e 64 69 2e 75 6d 69 6e 68 6f 2e 70 74 0d 0a 55 .di.umin ho.pt..U
      0060 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c ser-Agen t: Mozil
      0070 6c 61 2f 35 2e 30 20 28 58 31 31 3b 29 55 62 75 la/5.0 ( X11; Ubu
      0080 6e 74 75 3b 20 4c 69 6e 75 78 20 78 38 36 5f 36 ntu; Lin ux x86_6
      0090 34 3b 20 72 76 3a 36 31 2e 30 29 20 47 65 63 6b 4; rv:61 .0) Geck
      00a0 6f 2f 32 30 31 30 30 31 30 31 20 46 69 72 65 66 o/201001 01 Firef
      HTTP Request Method (http.request.method), 3 bytes

```

Bytes antes do caractere ASCII 'G'

Os bytes envolvidos a vermelho (no total 54) representam os bytes que precedem o primeiro caractere "G" do método HTTP GET. O número total de bytes na trama são 452 bytes.

$overhead = 54/452 = 0.12 = 12\%$, aproximadamente.

- 5) Através de visualização direta de uma trama capturada, verifique que, possivelmente, o campo FCS (Frame Check Sequence) usado para deteção de erros não está a ser usado. Em sua opinião, porque será?

O FCS é usado na camada de ligação, na medida em que permite detetar se ocorreram erros na transmissão do pacote.

Ele não existe numa trama Ethernet (que é o que estamos a analisar) visto que nesta raramente acontecem erros, levando a que não haja necessidade de os corrigir na grande maioria dos casos.

- 6) Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique. O endereço Ethernet (MAC) da fonte é o endereço físico da interface da placa de rede Ethernet do computador que utilizamos, que é 08:00:27:7b:63:0c.

- 7) Qual é o endereço MAC do destino? A que sistema corresponde? O endereço MAC do destino é 52:54:00:12:35:02, corresponde à interface de rede do router a que estamos conectados durante a aula prática.

- 8) Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida. São contidos nesta trama os seguintes protocolos: HTTP, Ethernet, TCP, IPv4.

2.2 Exercícios parte 4

Verifique o conteúdo da cache ARP do seu computador usando o comando `arp -a`. Para observar o protocolo ARP em operação, apague novamente a cache ARP e assegure-se que a cache do browser está vazia. Inicie a captura de tráfego com o Wireshark, e acesse a <http://miei.di.uminho.pt>. Efetue também um ping para um host da sala de aula (e.g. ping 192.168.100.xxx) que esteja a ser usado por outro grupo. Pare a captura de tráfego e tente localizar o tráfego ARP.

- 9) Observe o conteúdo da tabela ARP. Explique o significado de cada uma das colunas.

```
C:\WINDOWS\system32>ping 192.168.100.197

Pinging 192.168.100.197 with 32 bytes of data:
Reply from 192.168.100.197: bytes=32 time=3ms TTL=64
Reply from 192.168.100.197: bytes=32 time=1ms TTL=64
Reply from 192.168.100.197: bytes=32 time=2ms TTL=64
Reply from 192.168.100.197: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.100.197:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms

C:\WINDOWS\system32>arp -a

Interface: 192.168.100.226 --- 0x11

    Internet Address      Physical Address      Type
    192.168.100.161       88-d7-f6-1b-5c-4b    dynamic
    192.168.100.197       88-d7-f6-2c-42-59    dynamic
    192.168.100.254       00-0c-29-d2-19-f0    dynamic
    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
```

Tabela ARP e ping para uma interface da rede local

Nas tabelas ARP temos vários campos, em que cada linha se refere a uma única interface de rede de um dispositivo nela:

- **Internet Address:** Endereço IP dessa interface
- **Physical Address:** Endereço MAC dessa interface
- **Type:** Dynamic/Static, em que *dynamic* é usado para referenciar que o dispositivo é mantido na tabela ARP enquanto ainda permanecer na subrede, caso saia, pode ser eliminado da tabela. Por outro lado, entradas *static* permanecem até serem eliminadas manualmente da tabela, como por exemplo endereços de broadcast, que são constantes e necessários a qualquer momento.

- **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?

```
> Frame 322: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: RealtekS_64:b8:08 (00:e0:4c:64:b8:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  Sender IP address: 192.168.100.226
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.197
```

Trama Ethernet para um ARP request

Temos neste caso um MAC de broadcast (ff:ff:ff:ff:ff:ff) como destino, proveniente da nossa interface de rede do computador que utilizamos, isto é, a placa identificada por 00:e0:4c:64:b8:08.

Tem de ser utilizado um pedido em broadcast porque na tabela ARP ainda não está registado o endereço MAC da interface de destino, pelo que é preciso fazer um *request* a todos os dispositivos na rede local (só o que pretendemos comunicar dá *reply*) para determinar qual o endereço MAC do destino pretendido e, portanto, conseguir estabelecer a comunicação.

- **11)** Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?
Pela figura anterior retiramos que o valor é 0x0806, ou seja, esta trama é do tipo ARP.
- **12)** 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 valor do campo ARP opcode é 1, ou seja, indica que esta trama é do tipo request.
- **13)** Identifique que tipo de endereços está contido na mensagem ARP? Que conclui?
Estão contidos endereços de destino e origem, em que o destino é do tipo broadcast e o de origem é apenas a identificação da placa de rede Ethernet do computador que utilizamos. Concluímos o mesmo que em **10**).
- **14)** Explícite que tipo de pedido ou pergunta é feito pelo host de origem?
Utilizando o raciocínio em **10**), este tipo de pedidos é feito a todos as interfaces da rede local para determinar qual o dispositivo que tem o IP de destino contido na trama, ao qual apenas um responde com o seu endereço físico para permitir a comunicação.
- **15)** Localize a mensagem ARP que é a resposta ao pedido ARP efectuado.

```

> Frame 323: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▼ Ethernet II, Src: AsustekC_2c:42:59 (88:d7:f6:2c:42:59), Dst: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  > Destination: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  > Source: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
  Type: ARP (0x0806)
  Padding: 00000000000000000000000000000000
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
  Sender IP address: 192.168.100.197
  Target MAC address: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  Target IP address: 192.168.100.226

```

Trama Ethernet para o ARP reply ao request anterior

- a) Qual o valor do campo ARP opcode? O que especifica?
Como vemos pela figura acima, o opcode é 2, que identifica um ARP do tipo *reply*.
- b) Em que posição da mensagem ARP está a resposta ao pedido ARP?

▼ Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
Sender IP address: 192.168.100.197
Target MAC address: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
Target IP address: 192.168.100.226

0000	00 e0 4c 64 b8 08 88 d7 f6 2c 42 59 08 06 00 01	..Ld....,BY...
0010	08 00 06 04 00 02 88 d7 f6 2c 42 59 c0 a8 64 c5,BY..d.
0020	00 e0 4c 64 b8 08 c0 a8 64 e2 00 00 00 00 00 00	..Ld....d.....
0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Posição na trama do MAC Address da fonte

Vemos pela figura que o endereço MAC do destino está 5 bytes à frente do endereço fonte, ou seja, encontra-se do byte 33 ao byte 38 (6 bytes total). Este endereço é, no fundo, a resposta pertinente ao *request* original.

- 16) 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?

```

> Frame 311: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
v Ethernet II, Src: RealtekS_64:b8:08 (00:e0:4c:64:b8:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  Type: ARP (0x0806)
v Address Resolution Protocol (request/gratuitous ARP)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  [Is gratuitous: True]
  Sender MAC address: RealtekS_64:b8:08 (00:e0:4c:64:b8:08)
  Sender IP address: 192.168.100.226
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.226

```

0000	ff ff ff ff ff ff 00 e0	4c 64 b8 08 08 06 00 01 Ld.....
0010	08 00 06 04 00 01 00 e0	4c 64 b8 08 c0 a8 64 e2 Ld....d.
0020	00 00 00 00 00 00 c0 a8	64 e2 d.

Gratuitous ARP request

Este tipo de ARP requests tem uma flag que é "[Is Gratuitous: True]", que é usada para indicar que é, de facto, um Gratuitous ARP Request.

Para além disso, é utilizado um endereço sempre em broadcast como destino, que serve para indicar às outras interfaces da rede local que a nossa interface ainda não se desconectou/saiu da rede local, pois as entradas ARP são voláteis e precisam de estar constantemente a ser atualizadas. Daí existir um "Gratuitous"ARP request, serve para garantir que a nossa entrada nas tabelas dos outros utilizadores se mantém.

2.3 Exercícios parte 5

Construa uma topologia no emulador CORE com um host (n1) e dois servidores (n2, n3) interligados através de um hub.

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

The screenshot displays a network simulation environment with three terminal windows and a network diagram. The top-left terminal window shows the output of `tcpdump` on interface `eth0` of node n2, capturing ARP requests and ICMP echo requests and replies between 10.0.0.11 and 10.0.0.20. The top-right terminal window shows the output of `tcpdump` on interface `eth0` of node n3, capturing similar traffic. The bottom-left terminal window shows the output of a `ping` command from node n1 to 10.0.0.11, indicating successful connectivity with 0% packet loss. The bottom-right window shows a network diagram with three nodes: n1 (a laptop icon), n2 (a server icon), and n3 (a server icon). Node n1 is connected to both n2 and n3 via red lines, and both n2 and n3 are connected to each other, forming a hub topology. The IP addresses 10.0.0.24 and 10.0.0.11 are labeled on the connections.

Nós 1 e 2 com `tcpdump` ativo, topologia pedida e ping para o nó 2.

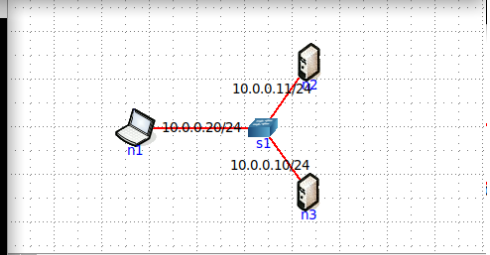
Usando o `tcpdump` e um `ping`, verificamos que o que é observado num nodo é idêntico nos restantes, em virtude da funcionalidade do `hub`, que envia a mesma informação para as várias interfaces ligadas a si.

- 18) 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.


```
root@n2:/tmp/pycore.42421/n2.conf
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:35:46.713572 ARP, Request who-has 10.0.0.11 tell 10.0.0.20, length 28
15:35:46.713626 ARP, Reply 10.0.0.11 is-at 00:00:00:aa:00:02 (oui Ethernet), length 28
15:35:46.713644 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 27, seq 1, length 64
15:35:46.713657 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 27, seq 1, length 64
15:35:47.743603 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 27, seq 2, length 64
15:35:47.743623 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 27, seq 2, length 64
15:35:48.768070 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 27, seq 3, length 64
15:35:48.768091 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 27, seq 3, length 64
15:35:51.840087 ARP, Request who-has 10.0.0.20 tell 10.0.0.11, length 28
15:35:51.840205 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:00 (oui Ethernet), length 28
10 packets captured
10 packets received by filter
0 packets dropped by kernel
root@n2:/tmp/pycore.42421/n2.conf#

root@n1:/tmp/pycore.42421/n1.conf# ping -c 3 10.0.0.11
PING 10.0.0.11 (10.0.0.11): 56(84) bytes of data:
64 bytes from 10.0.0.11: icmp_seq=1 ttl=64 time=0.124 ms
64 bytes from 10.0.0.11: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 10.0.0.11: icmp_seq=3 ttl=64 time=0.064 ms

--- 10.0.0.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2054ms
rtt min/avg/max/mdev = 0.060/0.082/0.124/0.031 ms
root@n1:/tmp/pycore.42421/n1.conf#
```



Nós 1 e 2 com tcpdump ativo, alteração à topologia anterior e ping para o nó 2.

Desta vez, seguindo o mesmo processo de 17), verificamos que apenas o *switch* redireciona informação apenas para a interface que é suposto que a receba, pelo que vemos que não há qualquer difusão da mesma informação em nodos diferentes.

3 Conclusões

No final de realizar este trabalho, num contexto geral, sentimos que conseguimos atingir os objetivos pretendidos, pelo que conseguimos por fim estabelecer uma ponte entre o que foi lecionado nas aulas teóricas e as aplicações práticas dos conceitos e temas propostos.

Em particular, nos **Exercícios da parte 3**, analisamos a fundo uma trama Ethernet utilizando o *Wireshark* e as suas ferramentas disponibilizadas, percebendo o funcionamento e a utilidade dos endereços MAC, em particular, que são únicos para cada máquina e que são essencialmente o seu endereço físico e, portanto, são a forma de as identificar nas subredes.

Nos **Exercícios da parte 4**, analisamos o funcionamento do protocolo ARP, acedendo às suas tabelas e verificando a sua utilidade assim como gestão. Permitiram-nos perceber que o ARP essencialmente estabelece estas ligações entre máquinas e a maneira como as estabelece, assim como as gere posteriormente.

Por fim, nos **Exercícios da parte 5**, conseguimos chegar à conclusão de que um *switch* gere melhor as colisões (eliminando-as) do que um *hub*, porque não difunde a mesma informação por todos os elementos imediatamente no seu domínio.

Sentimos que este projeto elucidou-nos bastante acerca do protocolo Ethernet, assim como as suas especificidades, como os endereços MAC, controlo de colisões e o protocolo ARP, que são a base para o funcionamento desta camada de *link*, que estabelece relação direta com a camada lógica do trabalho anterior.

Referências

1. Notas de Apoio das Aulas Teóricas
2. Protocolo ARP: <http://tools.ietf.org/html/rfc826.html>