

# **TP2: Protocolo IP**

Diogo Afonso Costa, Daniel Maia, and Vitor Castro

University of Minho, Department of Informatics, 4710-057 Braga, Portugal  
e-mail: {a78034,a77531,a77870}@alunos.uminho.pt

**Abstract.** Resumo...

## **1 Introdução**

## **2 Parte I - Datagramas e Fragmentação**

### **2.1 Exercício 1.b.**

**Questão**

**Resposta**

**Realização**

### **2.2 Exercício 1.c.**

**Questão**

**Resposta**

**Realização**

### **2.3 Exercício 1.d.**

**Questão**

**Resposta**

**Realização**

### **2.4 Exercício 2.a.**

**Questão**

Qual é o endereço IP da interface ativa do seu computador?

## Resposta

O endereço IP é 192.168.100.216.

## Realização

43	3.903320123	192.168.100.216	192.168.100.216	DNS	519 Standard query response 0x7472 A marco.uminho.pt A 193.136.9.240 no unres
44	3.903910826	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=1/256, ttl=1 (no response found!)
45	3.903938117	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=2/512, ttl=1 (no response found!)
46	3.903947143	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=3/768, ttl=1 (no response found!)
47	3.903956246	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=4/1024, ttl=2 (no response found!)
48	3.903964672	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=5/1280, ttl=2 (no response found!)
49	3.903970621	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=6/1536, ttl=2 (no response found!)
50	3.903975159	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=7/1792, ttl=3 (reply in 63)
51	3.903982289	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=8/2048, ttl=3 (reply in 64)
52	3.903988913	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=9/2304, ttl=3 (reply in 66)
53	3.903994876	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=10/2560, ttl=4 (reply in 67)
54	3.904000785	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=11/2816, ttl=4 (reply in 68)
55	3.904009311	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=12/3072, ttl=4 (reply in 69)
56	3.904015173	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=13/3328, ttl=5 (reply in 70)
57	3.904020348	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=14/3584, ttl=5 (reply in 71)
58	3.904023767	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=15/3840, ttl=5 (reply in 72)
59	3.904032851	192.168.100.216	193.136.9.240	ICMP	74 Echo (ping) request id=0x3ea1, seq=16/4096, ttl=6 (reply in 73)
60	3.904040755	192.168.100.216	192.168.100.216	ICMP	102 Time-to-live exceeded (time to live exceeded in transit)

Fig. 1. Identificação do endereço IP.

## 2.5 Exercício 2.b.

### Questão

Qual é o valor do campo protocolo? O que identifica?

### Resposta

O campo protocolo tem o valor "ICMP (1)". ICMP significa *Internet Control Message Protocol*. Este é utilizado para reportar erros no processamento de datagramas. Efetivamente, dentro dos possíveis erros temos, *destination unreachable* (quando o datagrama não consegue alcançar o destino), *time exceeded message* (quando um *gateway* processa um datagrama e descobre que o *TTL* é zero e tem que descartar o datagrama e consequentemente notificar o *host*), *echo request/reply* (quando são enviadas mensagens para funções de teste e controle da rede (*request*), caso a máquina esteja ligada responde com um *reply*) [1] [2]. Como as mensagens ICMP encontram-se ao nível de rede, estas são também elas encapsuladas em datagramas IP que, consequentemente, usam o protocolo IP.

Assim sendo, analisando a primeira mensagem ICMP, nomeadamente no separador do *Internet Protocol Version 4*, percebemos que se trata de uma mensagem que vem num protocolo *ICMP*. Além disso, é possível concluir que se trata de uma mensagem de *echo request*, se observarmos o campo *Type* no separador *Internet Control Message Protocol*. Desta forma, pode-se concluir que o computador usado para a resolução deste trabalho está a tentar perceber se consegue estabelecer uma ligação com o *host* marco.uminho.pt e para isso usa mensagens *ICMP* do tipo *echo request*.

## Realização

```
Internet Protocol Version 4, Src: 192.168.100.216, Dst: 193.136.9.240
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0xb224 (45604)
Flags: 0x00
Fragment offset: 0
Time to live: 1
Protocol: ICMP (1)
Header checksum: 0x16a4 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.100.216
Destination: 193.136.9.240
[Source GeoIP: Unknown]
[Destination GeoIP: Portugal]
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x43d8 [correct]
[Checksum Status: Good]
Identifier (BE): 16033 (0x3ea1)
Identifier (LE): 41278 (0xa13e)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[No response seen]
Data (32 bytes)
```

**Fig. 2.** Identificação do campo *Protocol*.

## 2.6 Exercício 2.c.

### Questão

Quantos bytes tem o cabeçalho IP(v4)? Quantos bytes tem o campo de dados (payload) do datagrama? Como se calcula o tamanho do payload?

### Resposta

O cabeçalho IPv4 tem 20 bytes.

O campo de dados (payload) do datagrama tem 40 bytes.

O cálculo do payload é feito retirando o tamanho do cabeçalho ao tamanho total do datagrama (60 bytes). Desta forma, basta fazer  $60 - 20 = 40bytes$ .

### Realização

```
Internet Protocol Version 4, Src: 192.168.100.216, Dst: 193.136.9.240
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0xb224 (45604)
Flags: 0x00
Fragment offset: 0
Time to live: 1
Protocol: ICMP (1)
Header checksum: 0x16a4 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.100.216
Destination: 193.136.9.240
[Source GeoIP: Unknown]
[Destination GeoIP: Portugal]
Internet Control Message Protocol
```

**Fig. 3.** Identificação do tamanho do *header* e do *payload*.

## 2.7 Exercício 2.d.

### Questão

O datagrama IP foi fragmentado? Justifique.

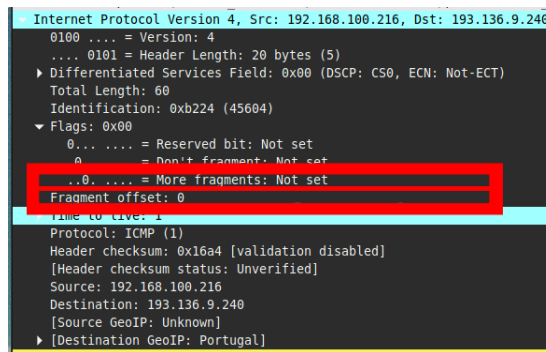
### Resposta

A fragmentação acontece quando o tamanho total do datagrama excede o *MTU* disponível. Tendo em conta que por defeito o *traceroute* usa 60 bytes por datagrama e tem-se um *MTU* disponível de 1500 bytes, podemos conjecturar que não haverá fragmentação.

A verificação se um datagrama foi ou não fragmentado é feita com base em dois valores, o *fragment offset* (indica o *offset* em que o datagrama atual encaixa no datagrama original) e a *flag more fragments* (indica se existe mais fragmentos). Neste datagrama em específico o *fragment offset* = 0 e a *flag more fragments* = 0. Desta forma, tendo em conta o *fragment offset*, sabe-se que se o datagrama foi fragmentado então ele é necessariamente o primeiro. Além disso, se analisarmos a *flag more fragments* concluímos que para além do datagrama atual não existe mais nenhum associado a este.

Assim sendo, conjugando a informação dos dois parâmetros percebe-se que se o datagrama é o primeiro e não existe mais nenhum associado, então este é único e não foi fragmentado.

### Realização



```
Internet Protocol Version 4, Src: 192.168.100.216, Dst: 193.136.9.240
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xb224 (45604)
  Flags: 0x00
    0... .... = Reserved bit: Not set
    0... .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    Fragment offset: 0
  Time to live: 1
  Protocol: ICMP (1)
  Header checksum: 0x16a4 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.100.216
  Destination: 193.136.9.240
  [Source GeoIP: Unknown]
  [Destination GeoIP: Portugal]
```

Fig. 4. Fragmentação.

## 2.8 Exercício 2.e.

### Questão

Ordene os pacotes capturados de acordo com o endereço IP fonte (e.g., selecionando o cabeçalho da coluna Source), e analise a sequência de tráfego ICMP gerado a partir do endereço IP atribuído à sua máquina. Para a sequência de mensagens ICMP enviadas pelo seu computador, indique que campos do cabeçalho IP variam de pacote para pacote.

### **Resposta**

Os campos que vêm os seus valores alterados correspondem à *identification*, *header checksum* e *time to live (TTL)*.

A *identificação* muda pois este campo identifica unicamente cada datagrama e visto que estes são sempre diferentes então o campo também o será.

O *header checksum* permite verificar que determinado header foi ou não corrompido. Desta forma o *checksum* identifica um determinado *header* num determinado estado. Assim sendo, o *checksum* muda pois este campo utiliza no seu algoritmo todas as palavras de 16 bits do *header* [1]. Efetivamente, sabendo que o *header*, propriamente dito, muda de datagrama para datagrama então o seu *checksum* também vai mudar.

### **Realização**

#### **2.9 Exercício 2.f.**

##### **Questão**

##### **Resposta**

##### **Realização**

#### **2.10 Exercício 2.g.**

##### **Questão**

##### **Resposta**

##### **Realização**

#### **2.11 Exercício 3.a.**

##### **Questão**

##### **Resposta**

##### **Realização**

#### **2.12 Exercício 3.b.**

##### **Questão**

**Resposta**

**Realização**

**2.13 Exercício 3.c.**

**Questão**

**Resposta**

**Realização**

**2.14 Exercício 3.d.**

**Questão**

**Resposta**

**Realização**

**2.15 Exercício 3.e.**

**Questão**

**Resposta**

**Realização**

### **3 Parte II - Endereçamento e Encaminhamento IP**

#### **3.1 Exercício 2.1.a**

**Questão**

**Resposta**

**Realização**

#### **3.2 Exercício 2.1.b**

**Questão**

**Resposta**

**Realização**

#### **3.3 Exercício 2.1.c**

**Questão**

**Resposta**

**Realização**

#### **3.4 Exercício 2.1.d**

**Questão**

**Resposta**

**Realização**

#### **3.5 Exercício 2.1.e**

**Questão**

**Resposta**

**Realização**

**3.6 Exercício 2.2.a**

**Questão**

**Resposta**

**Realização**

**3.7 Exercício 2.2.b**

**Questão**

**Resposta**

**Realização**

**3.8 Exercício 2.2.c**

**Questão**

**Resposta**

**Realização**

**3.9 Exercício 2.2.d**

**Questão**

**Resposta**

**Realização**



### **3.10 Exercício 2.2.e**

**Questão**

**Resposta**

**Realização**

### **3.11 Exercício 3.1**

**Questão**

**Resposta**

**Realização**

### **3.12 Exercício 3.2**

**Questão**

**Resposta**

**Realização**

### **3.13 Exercício 3.3**

**Questão**

**Resposta**

**Realização**

According to Table 5...

## **4 Conclusions**

Neste trabalho...

## **References**

1. : Internet Control Message Protocol. RFC 792 (1981)
2. Wikipedia: Internet control message protocol. [https://pt.wikipedia.org/wiki/Internet\\_Control\\_Message\\_Protocol](https://pt.wikipedia.org/wiki/Internet_Control_Message_Protocol) (2017) [Online; acedido a 4-Novembro-2017].

(a) Delay and jitter	(b) Delay and loss
(c) Delay and throughput	(d) Jitter and loss
(e) Jitter and throughput	(f) Loss and throughput

**Fig. 5.** Tabela exemplo.