**Grupo 4**

**Laboratório:** Sexta-Feira (8:30-10:00 - RC91795L04)

**Membros:**

* 84718 - Francisco Aguiar
* 84719 - Gonçalo Marques
* 84740 - Manuel Sousa

LAB #9

### Completar o Código

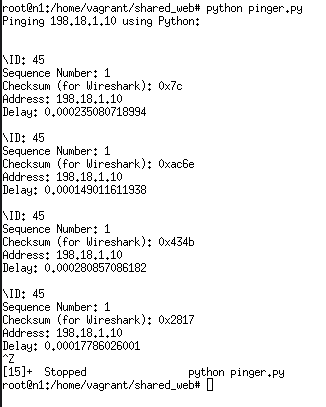
Primeiro completámos a parte que faltava do código. Bastava completar um pouco da função receiveOnePing(), responsável por receber o *pong* isto é, a mensagem do recetor de volta ao emissor (o que inicia o *ping*). Para isso apenas tivemos de separar os headers, do restante pacote que era recebido. Com o headers separámos entre os vários componentes que o constituem, são eles o tipo, o código, o checksum, o id e o sequence number. Contudo apenas necessitamos do ID que o receptor nos envia, de forma a verificar na receção de um pong se recebemos o pong associado ao ping feito pela máquina de origem. O ID é um Process ID criado na máquina que inicia o ping.

Código adicionado:

#Fill in start  
  
 # Explicacao do pacote  
 # 0 - 19 Bytes do pacote IP  
 # 20 - 27 Bytes do pacote ICMP  
 icmpHeader = recPacket[20:28] # 8 Bytes do Header do pacote ICMP  
 # o fornato 'b' e signed\_char, um 'H' e um unsigned short e um 'h' short (isto em C, porque para o python e tudo inteiros)  
 (Type, Code, Checksum, rID, Sequence) = struct.unpack('bbHHh', icmpHeader) # Parse do binario  
 if ID == rID:   
 # O Standard Size de um Double em python sao 8 Bytes, e e esse o valor que consideramos  
 # o formato 'd' diz que espera receber um Double  
 # Estes 8 Bytes sao os 8 bytes do timestamp  
 timeData = struct.unpack('d', recPacket[28:28 + 8])[0] # Faz unpack de acordo com o formato  
 checkS = str(hex(int(Checksum)))[2:]  
 checkS = "0x" + checkS[2:] + checkS[0:2]  
 print("\n\ID: " + str(ID) + "\nSequence Number: " + str(Sequence) + "\nChecksum (for Wireshark): " + checkS + "\nAddress: " + str(addr[0]))  
 return timeReceived - timeData  
 else:  
 return 'different IDS'

### Executar e Analisar a experiência com o Wireshark

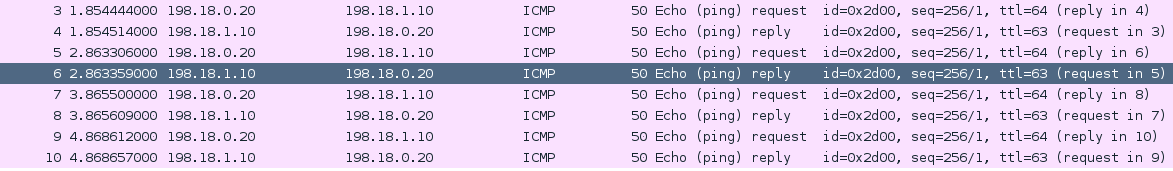
Utilizámos o Wireshark para comprovar se o nosso código Python está completo e correto. Para esta experiência abrimos o core, tal como indicado no guião, e iniciamos o nosso script no terminal associado ao host computador (198.18.0.20), que vai fazer um ping ao host servidor (198.18.1.10).



Colocámos algum output na função que foi completada de forma a ter informação sobre o que está a acontecer. Por observação à figura 1, podemos ver conteúdo relativo aos *pongs* enviados pelo servidor que estamos a pingar. Vamos observar a nossa experiência no wireshark para comprovar que as interseções e o output dado pelo código estão em sintonia.

**Figura 1 - Execução do script de ping**

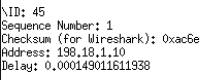
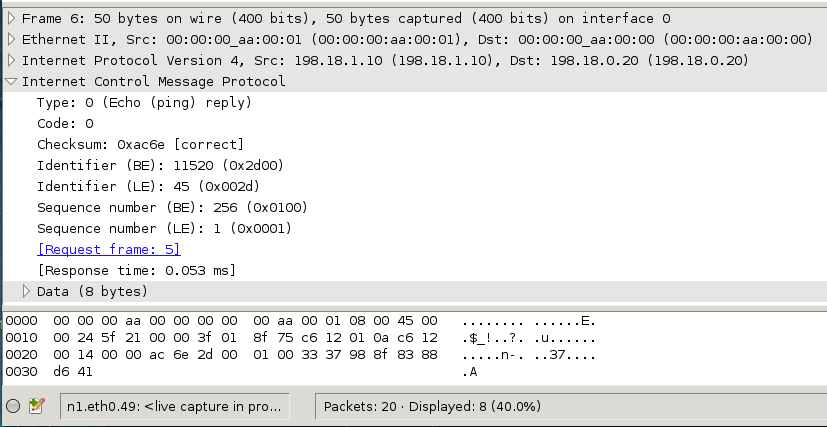
Facilmente verificamos por observação aos logs do Wireshark (Figura 2) que estamos a efetuar corretamente um ping, visto que ele reconhece o pacote como ICMP. Por cada ping também podemos observar o respetivo *pong*. Vamos olhar com mais detalhe para os pacotes snifados.



**Figura 2 - Intersecção do Wireshark**

Para estudarmos ao detalhe escolhemos o segundo pong enviado pelo receptor, assinalado na figura 2.

Comprovamos em comparação com a figura 1 que o Checksum,o sequence number o endereço e o ID são iguais, por isso podemos concluir que a nossa experiência foi efetuada corretamente. Além disso, todos os tamanhos associados às várias partes constituintes do pacote snifado, são exatamente iguais aos pacotes enviados pela ferramenta ping do sistema operativo.



**Figura 3 - Detalhes do segundo *pong***