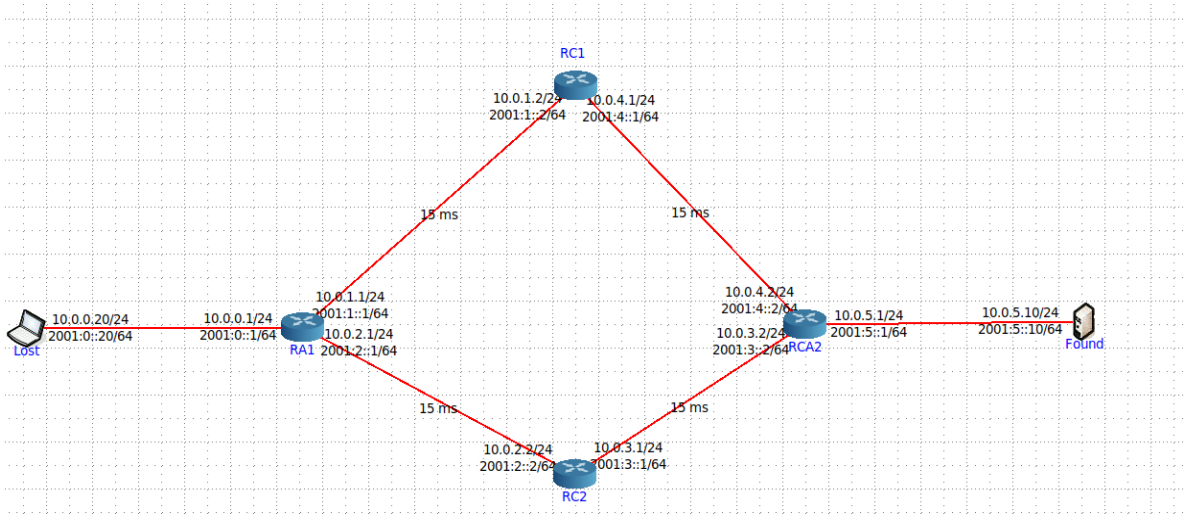
Relatório TP2

Grupo PL70 :
André Santos a106854
Daniel Parente a107363
Pedro Ferreira a107292

PARTE I - Protocolo IPv4 :: Datagramas IP e Fragmentação

Esta é a topologia proposta pelo enunciado do problema:



1.

a) **Questão**: Active o Wireshark no host Lost. Numa shell de Lost execute o comando traceroute -I para o endereço IP do Found. Registe e analise o tráfego ICMP enviado pelo sistema Lost e o tráfego ICMP recebido como resposta. Explique os resultados obtidos tendo em conta o princípio de funcionamento do traceroute.

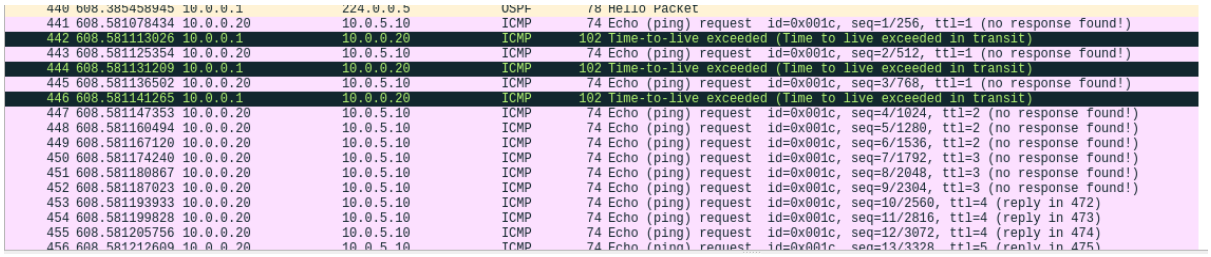
Resposta: O traceroute mostra o caminho que os pacotes seguem até o seu destino. O traceroute envia então pacotes, neste caso três por defeito, aumentando o **TTL**(Time-To-Live) de forma crescente. Isto é, os primeiros 3 pacotes foram enviados com um **TTL** = 1, chegando assim ao **RA1**, onde o valor é reduzido, atingindo o valor **zero** sendo enviada a mensagem de erro "***Time Exceeded"***. De seguida, os pacotes são enviados com um **TTL** = 2 e assim sucessivamente até alcançar o destino final, neste caso o ***Found.***

Resultado do traceroute :

```
root@Lost:/tmp/pycore.39905/Lost.conf# traceroute -I 10.0.5.10
traceroute to 10.0.5.10 (10.0.5.10), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.052 ms  0.009 ms  0.007 ms
 2  10.0.1.2 (10.0.1.2)  30.455 ms  30.444 ms  30.438 ms
 3  10.0.3.2 (10.0.3.2)  60.828 ms  60.825 ms  60.821 ms
 4  10.0.5.10 (10.0.5.10)  60.815 ms  60.811 ms  60.807 ms
```

1. **TTL** = 1, **Lost** - **RA1.**
2. **TTL** = 2, **Lost** - **RA1** - **RC1**. Como impusemos um tempo de propagação entre o **RA1** - **RC1** de **15ms**, tendo em conta que temos de contar a ida e a volta entre esses dois routers de acesso.
3. **TTL** = 3, **Lost** - **RA1** - **RC1** - **RA2**, também está imposto um tempo de propagação de **15ms** entre **RC1** - **RA2**.
4. **TTL** = 4, **Lost** - **RA1** - **RC1** - **RA2** - **Found**.

Os tempos de propagação entre tanto o **Lost** - **RA1** como do **RA2** - **Found** consideram-se de certa forma "irrelevantes".



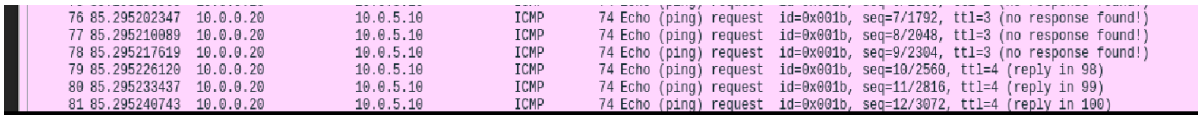
Aqui nesta figura, conseguimos observar que a cada valor de TTL corresponde três entradas diferentes, dado ao traceroute ter enviado três pacotes por valor.

b) **Questão**: Qual deve ser o valor inicial mínimo do campo TTL para alcançar o servidor Found? Esboce um esquema com o valor do campo TTL à chegada a cada um dos routers percorridos até ao servidor Found. Verifique na prática que a sua resposta está correta.

Resposta: O valor inicial mínimo do campo **TTL** é 4, como podemos ver analisando o wireshark (print ilustrativo em baixo), onde é apenas feito um reply quando o valor do **TTL** é

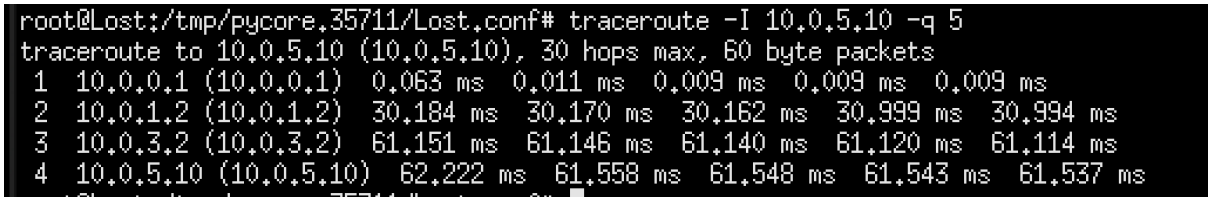
4, chegando assim ao destino, **Found.**

PC —--- RA1 (TTL = 1) —---- RC2 (TTL = 2) —------ RA2 (TTL = 3) —------ S (TTL = 4)



c) **Questão**: Calcule o valor médio do tempo de ida-e-volta (RTT - Round-Trip Time) obtido no acesso ao servidor. Por modo a obter uma média mais confiável, poderá alterar o número pacotes de prova com a opção -q.

Resposta: Com o comando ***traceroute -I 10.0.5.10 -q 5,*** mandando assim 5 pacotes, obtivemos os seguintes tempos :



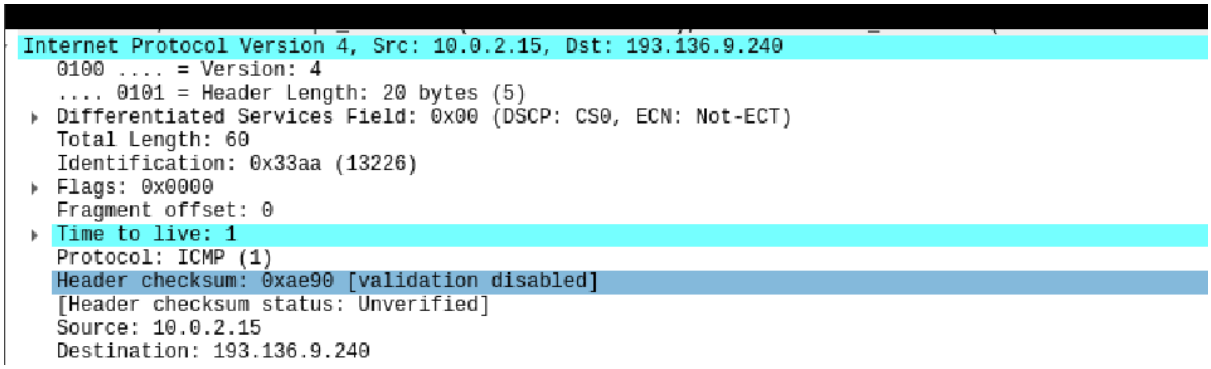
Fazendo a média dos valores, da linha 4, ou seja quando o valor do **TTL** = 4, concluimos que o valor médio do tempo de ida-e-volta (**RTT - Round-Trip Time**) é **61.682.**

d) **Questão**: O valor médio do atraso num sentido (One-Way Delay) poderia ser calculado com precisão dividindo o RTT por dois? O que torna difícil o cálculo desta métrica numa rede real?

Resposta: Dividir o **RTT** (Round-Trip Time) por dois não é uma forma precisa de calcular o **One-Way Delay** (OWD) em redes reais, porque o **RTT** mede o tempo total de ida e volta, e esse valor pode não ser simétrico nos dois sentidos, pelas seguintes razões:

- Variação do congestionamento da rede fazendo variar os tempos.
- Assimetria das rotas pois os pacotes podem seguir percursos diferentes.
- Routers podem processar os pacotes de forma diferente.

2-



a) **Questão**: Qual é o endereço IP da interface ativa do seu computador?

Resposta: Endereço IP da interface ativa do nosso computador: **10.0.2.15**.

b) **Questão**: Qual é o valor do campo protocol? O que permite identificar?

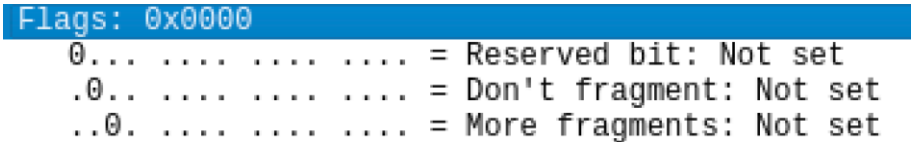
Resposta: O valor do campo protocolo é **ICMP**, este protocolo permite identificar o tipo de comunicação que está a ser estabelecida, pois é frequentemente usada para mensagens de erro e para "diagnósticos" de rede. O conteúdo do pacote pode ser analisado para determinar se é um ***(ping) Echo Request/Reply***, um ***Destination Unreachable***, ***Time Exceeded***, entre outros.

c) **Questão**: Quantos bytes tem o cabeçalho IPv4? Quantos bytes tem o campo de dados (payload) do datagrama? Como se calcula o tamanho do payload?

Resposta: O header tem 20 bytes, como podemos ver na figura. A partir do ***total length***, subtrai-se o tamanho do ***header***, para obter o tamanho do ***payload***, que são, neste caso, 60 - 20 = 40 bytes.

d) **Questão**: O datagrama IP foi fragmentado? Justifique.

Resposta: O datagrama **não** foi fragmentado. O ***fragment offset*** indica a posição do fragmento no datagrama original. Se fosse maior do que 0, isso significaria que é uma continuação de um datagrama previamente fragmentado, o que não se verifica. Além disso, como a flag do ***Don't fragment*** está posta a 0, isso indica que o datagrama não pode ser fragmentado.



e) **Questão**: Analise a sequência de tráfego ICMP gerado a partir do endereço IP atribuído à interface da sua máquina. Para a sequência de mensagens ICMP enviadas pelo seu computador, indique que campos do cabeçalho IP variam de pacote para pacote. Justifique estas mudanças.

Resposta: Até chegar ao destino, os únicos campos que mudam nos headers do IP de cada pacote são o identificador, o TTL, e o header checksum. O identificador incrementa a cada pacote que se manda, o TTL, de forma expectável, aumenta de cada vez que chega a um router novo que não é o destino. O header checksum não sei.

f) **Questão**: Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?

Resposta: Aumentam ambos em uma unidade.

g) **Questão**: Ordene o tráfego capturado por endereço destino e encontre a série de respostas ICMP TTL Exceeded enviadas ao seu computador.

i) **Questão**: Qual é o valor do campo TTL recebido no seu computador? Esse valor permanece constante para todas as mensagens de resposta ICMP TTL Exceeded recebidas no seu computador? Porquê?

Resposta: O valor do campo **TTL** não permanece constante para todas as mensagens de resposta ***ICMP TTL Exceeded*** recebidas. Detetamos os valores : 255, 2, 1, 3.
Os valores que obtemos são diferentes pois o valor do campo **TTL** nas mensagens **ICMP** de resposta são definidos pelo dispositivo que gera a mensagem, por isso vemos valores diferentes com mensagens de diferentes origens.

ii) **Questão**: Por que razão as mensagens de resposta ICMP TTL Exceeded são sempre enviadas na origem com um valor relativamente alto?

Resposta: São enviadas com valor relativamente alto para garantir que o valor é suficiente para chegar ao destino, sem que se perca a mensagem.

h) **Questão**: A informação contida no cabeçalho ICMP poderia ser incluída no cabeçalho IPv4? Se sim, quais seriam as suas vantagens/desvantagens?

Resposta: Na teoria, a informação contida no cabeçalho **ICMP** poderia ser incluída no cabeçalho **IPv4**, simplificando assim a pilha de protocolos, o que iria reduzir o tamanho total dos pacotes, economizando largura de banda aumentando também a rapidez de processamento. No entanto, traria também algumas desvantagens pois iria aumentar a dificuldade de filtrar as informações de diagnóstico e controlo fornecidas

pelo **ICMP,** como também poderia piorar a segurança pois todos os diapositivos processam **IPv4** e desta forma seria ampliada a superfície de ataque onde podia agora ser obtida informação de controlo.

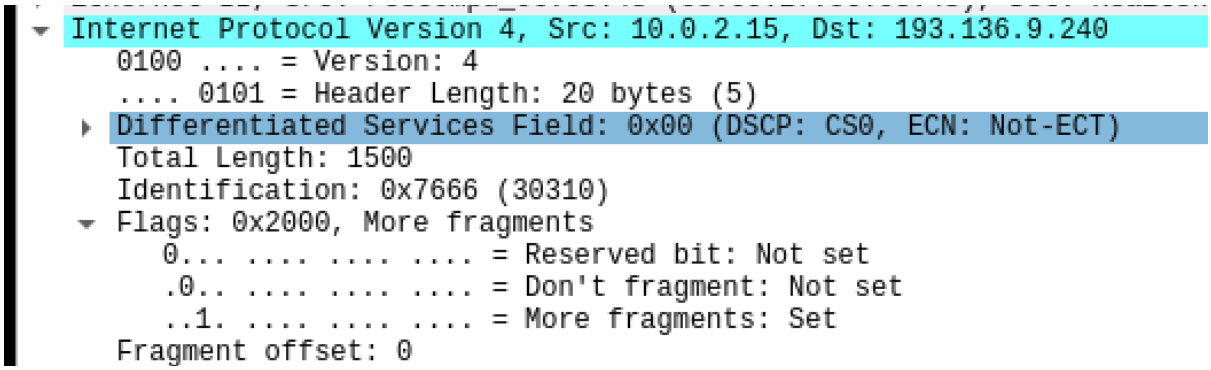
3-

a) **Questão**: Localize a primeira mensagem ICMP. Porque é que houve necessidade de fragmentar o pacote inicial?

Resposta: Houve necessidade de fragmentar o pacote inicial pois ao definir que o número de bytes enviados no campo de dados do pacote **ICMP** fossem 3870 bytes, o tamanho total do pacote excedeu o **MTU(Maximum Transmission Unit)** da rede, tendo então que ser fragmentado para ser transmitido.

b) **Questão**: Imprima o primeiro fragmento do datagrama IP segmentado. Que informação no cabeçalho indica que o datagrama foi fragmentado? Que informação no cabeçalho IP indica que se trata do primeiro fragmento? Qual é o tamanho deste datagrama IP?

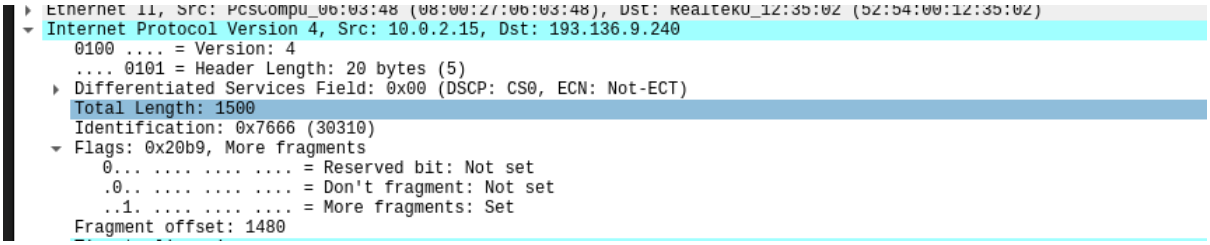
Resposta: Como podemos ver na imagem abaixo, a partir do primeiro fragmento, conseguimos saber que o segmento foi fragmentado através da flag ***"More Fragments"*** igualada a 1. Conseguimos também saber que se trata do primeiro fragmento pois o ***"Fragment offset"*** é 0. O tamanho total do datagrama é de 1500 bytes.



c) **Questão**: Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1º fragmento? Existem mais fragmentos? O que nos permite afirmar isso?

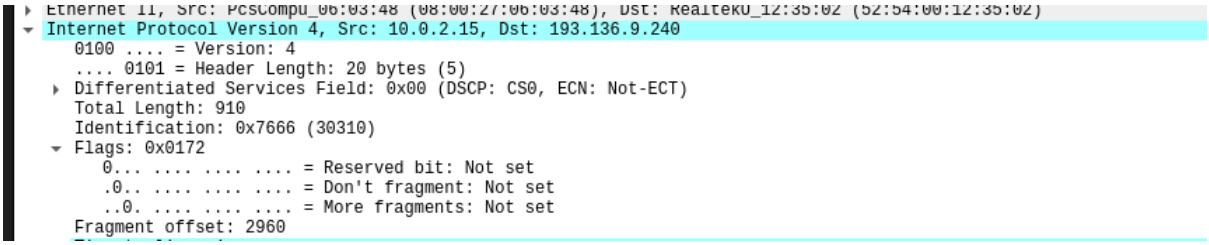
Resposta: Verificando o Fragment offset na imagem em baixo, vemos o valor 1480, mostrando assim que não se trata do primeiro fragmento. Vemos também que a flag ***More fragments*** está igualada a 1 mostrando que há mais fragmentos.Podemos reparar também que o valor ***Identification*** é o mesmo da alínea anterior confirmando

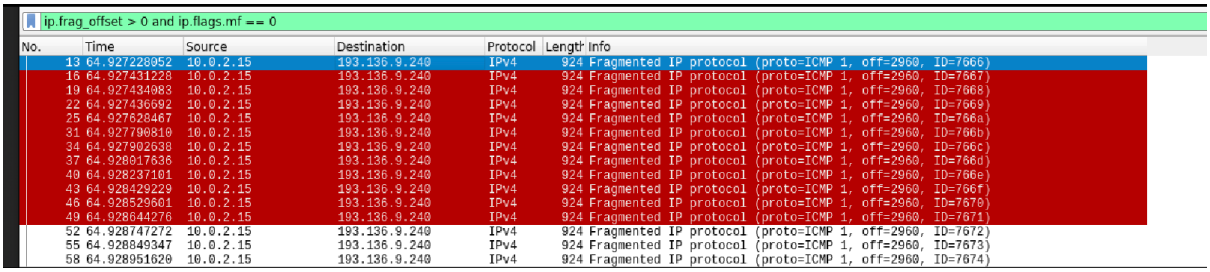
assim que trata de outro fragmento do mesmo pacote.



d) **Questão**: Quantos fragmentos foram criados a partir do datagrama original? Como se detecta o último fragmento correspondente ao datagrama original? Estabeleça um filtro no Wireshark que permita listar apenas o último fragmento do primeiro datagrama IP segmentado.

Resposta: A partir do datagrama original foram criados 3 fragmentos. Podemos detetar o último fragmento vendo que o número de identificação se mantém,o ***Fragment offset*** é maior que zero, mas a flag ***More fragments*** fica igualada a 0 indicando que não há mais fragmentos. Sendo assim, um filtro que permita listar apenas o último fragmento do primeiro datagrama IP segmentado é "*ip.frag_offset > 0 and ip.flags.mf == 0*"





e) **Questão**: Indique, resumindo, os campos que mudam no cabeçalho IP entre os diferentes fragmentos, e explique a forma como essa informação permite reconstruir o datagrama original.

Resposta: Resumindo, os campos que mudam são :

- ***Fragment offset***, que indica a posição do fragmento em relação ao datagrama.

- No campo das Flags, pode variar a flag ***More fragments*** que nos permite perceber se há mais fragmentos, e juntamente com a informação do ***fragment offset***, permite detetar o último fragmento.
- ***Total Length*** que indica o tamanho total do fragmento
- ***Header Checksum*** é calculado para cada fragmento, pois o cabeçalho IP muda

f) **Questão**: Estime teoricamente o número de fragmentos gerados e o número de bytes transportados em cada um dos fragmentos. Apresente todos os cálculos efetuados, incluindo os campos do cabeçalho IP relevantes para cada um dos fragmentos.

Resposta: 3870 -> só para o ICMP, adiciona-se o cabeçalho do IP (20 bytes)
max 1500
1º fragmento = 1480 + 20
sobram 3870 - 1480 = 2390 bytes
2º fragmento = 1480 + 20
sobram 2390 - 1480 = 910 bytes
3 º fragmento = 910 + 20 bytes

g) **Questão**: Por que razão apenas o primeiro fragmento de cada pacote é identificado pelo Wireshark como sendo um pacote ICMP? Justifique a sua resposta com base no conceito de Fragmentação apresentado nas aulas teóricas.

Resposta: Ao especificar que o tamanho do pacote **ICMP** seja 1370 bytes, esses bytes são preenchidos com a informação habitual do **ICMP** e o resto é ocupado por "lixo".
Quando o pacote inicial é fragmentado, a informação do **ICMP** fica presente no primeiro fragmento, ficando assim os restantes com "lixo" . Devido à ausência das informações do cabeçalho **ICMP** nos outros fragmentos o ***wireshark*** reconhece apenas o primeiro fragmento como sendo um pacote **ICMP**.

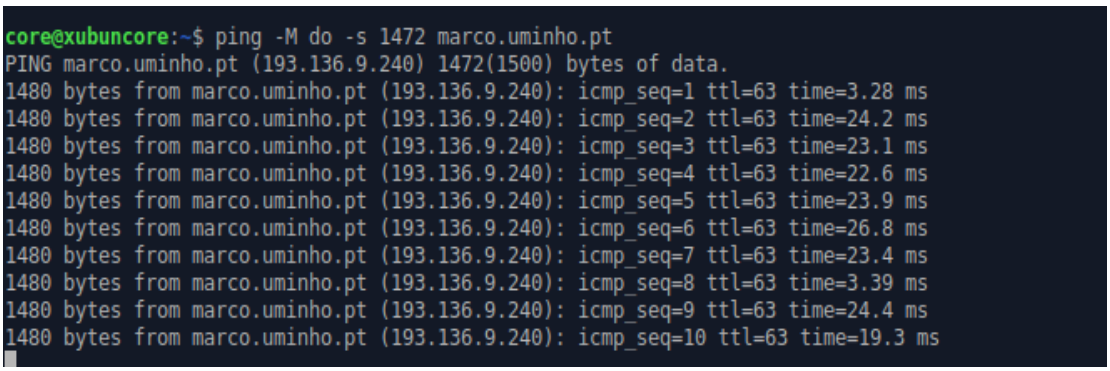
h) **Questão:** Com que valor é o tamanho do datagrama comparado a fim de se determinar se este deve ser fragmentado? Quais seriam os efeitos na rede ao aumentar/diminuir este valor?

Resposta: 1500 bytes. Aumentar esse valor iria reduzir o número de pacotes necessários para a mesma quantidade de informação, o que conduz a uma diminuição da sobrecarga de cabeçalhos, no entanto, pacotes maiores pode aumentar a probabilidade de perdas de pacotes em redes congestionadas e podem também não

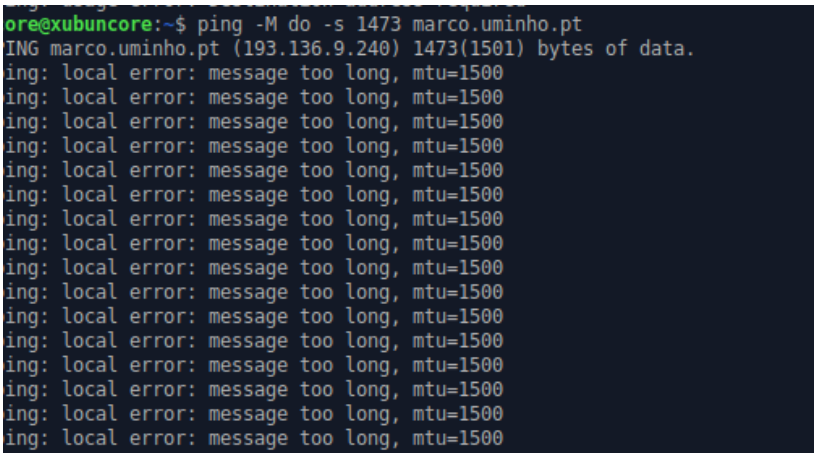
ser suportados por todos os dispositivos. Por outro lado, diminuir o valor teria o efeito contrário, aumentando a sobrecarga de cabeçalhos, mas iria beneficiar redes com baixa largura de banda pois os pacotes teriam menor chance de serem corrompidos e podem ser transmitidos mais rapidamente.

i) **Questão**: Sabendo que no comando ping a opção "-f" (Windows), "-M do" (Linux) ou "–D" (Mac) ativa a flag "Don't Fragment" (DF) no cabeçalho do IPv4, usando ping SIZE marco.uminho.pt, (opção pkt_size = –l (Windows) ou –s (Linux, Mac), determine o valor máximo de SIZE sem que ocorra fragmentação do pacote? Justifique o valor obtido.

Resposta: O valor máximo de ***SIZE*** sem que ocorra fragmentação é 1472. Obteve-se este valor pois o comando permite definir o tamanho do payload do ***ICMP***, e sabemos também que ***Total Length = IP Header + ICMP Header + ICMP Payload.*** Sabendo que o ***Total Length*** máximo são 1500 bytes temos que ***1500 = 20 + 8 + ICMP Payload .*** Podemos assim concluir que o valor máximo de bytes para o ***ICMP Payload*** são 1472.



Como podemos ver também na imagem abaixo, com o ***SIZE*** = 1473 já obtemos a seguinte mensagem de erro :



PARTE II - – Protocolo IPv4:: Endereçamento e Encaminhamento IP

1 -

a) **Questão**: De modo a garantir uma posição estrategicamente mais vantajosa e ter casa de férias para relaxar entre batalhas, ordena a construção de um segundo Castelo, em Braga. Não tendo qualquer queixa do serviço prestado, recorre aos serviços do ISP ReiDaNet, que já utiliza no condado, para ter acesso à rede no segundo Castelo. O ISP atribuiu-lhe o endereço de rede IP 172.68.XX.192/26 em que XX corresponde ao seu número de grupo (PLXX). Defina um esquema de endereçamento que permita o estabelecimento de pelo menos 6 redes e que garanta que cada uma destas possa ter 5 ou mais hosts. Assuma que todos os endereços de sub-redes são utilizáveis.

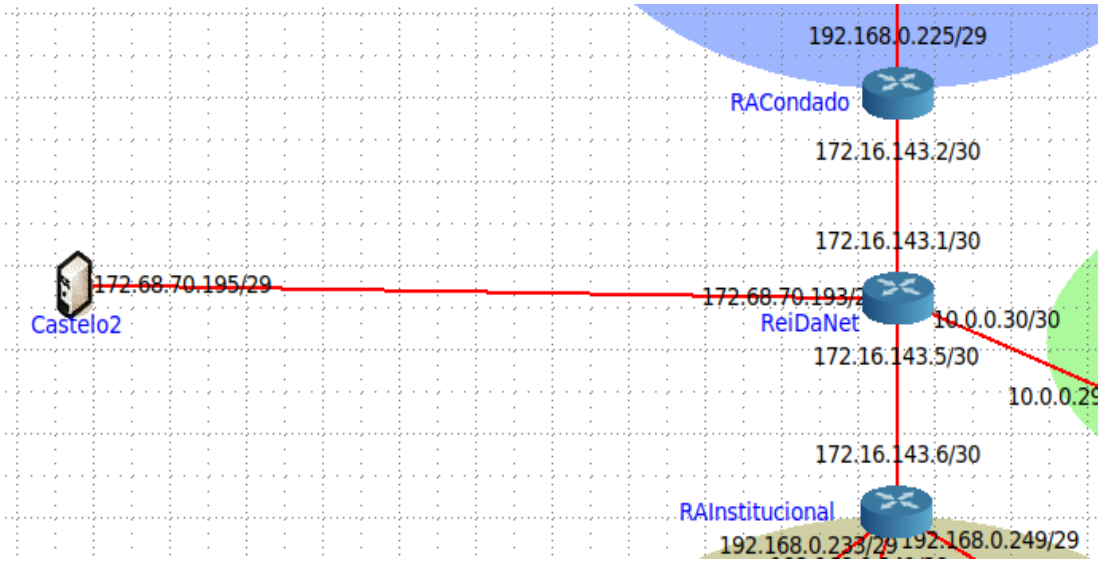
Resposta: Máscara /26 => 26 bits para o prefixo de rede
32-26 = 6 bits para : 1. identificar SRs (sub-redes);
2. identificar hosts.
2^6 = 64 endereços IP/SR.

Determinar a máscara da sub-rede:
2^(x-26) > 6 ⇔x-26 = 3 ⇔x = 29 (**Máscara**: /29)
Isto atende ao primeiro requerimento: pelo menos 6 subredes

Com máscara /29, 2^(32-29) = **8 endereços IP/SR**
2 endereços serão para o **host** e para broadcast, o que deixa 6 endereços para usar
Isso atende ao segundo requerimento: 5 ou mais hosts

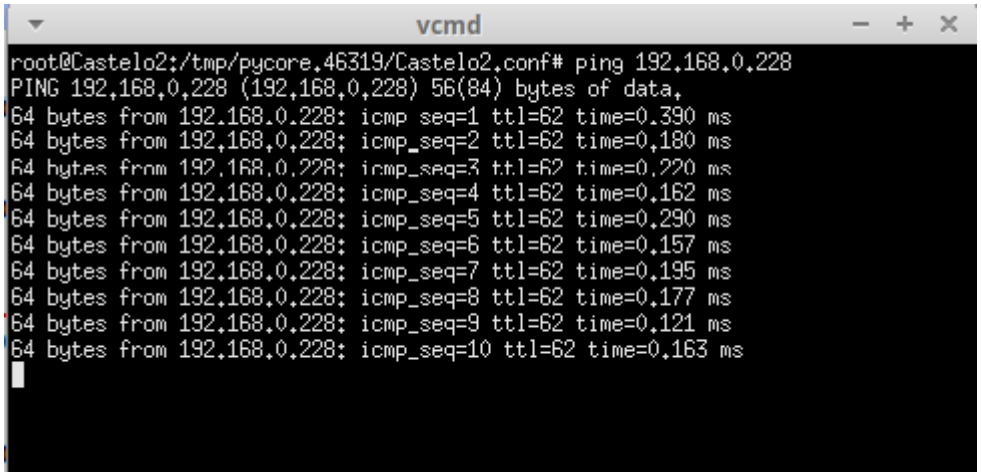
SR1: 172.68.70.192/29
Endereços válidos: 172.68.70.193 - 172.68.70.198
SR2: 172.68.70.200/29
Endereços válidos: 172.68.70.201 - 172.68.70.206
SR3: 172.68.70.208/29
Endereços válidos: 172.68.70.209 - 172.68.70.214
SR4: 172.68.70.216/29
Endereços válidos: 172.68.70.217 - 172.68.70.222
SR5: 172.68.70.224/29
Endereços válidos: 172.68.70.225 - 172.68.70.230
SR6: 172.68.70.232/29
Endereços válidos: 172.68.70.233 - 172.68.70.238

b) **Questão**: Ligue um novo host Castelo2 diretamente ao router ReiDaNet. Associe-lhe um endereço, à sua escolha, pertencente a uma sub-rede disponível das criadas na alínea anterior (garanta que a interface do router ReiDaNet utiliza o primeiro endereço válido da sub-rede escolhida). Verifique que tem conectividade com os dispositivos do Condado Portucalense.



Resposta: Como pode-se ver, foi escolhido o endereço 172.68.70.195/29 para o host, pertencente à 1ª sub-rede, e o router do **ReiDaNet** ficou com o endereço 172.68.70.193/29, o primeiro endereço válido dessa sub-rede.

Nesta imagem, podemos confirmar que existe conectividade com o dispositivo “Castelo”, da sub-rede “Condado Portucalense”:



c) **Questão**: Não estando satisfeito com a decoração deste novo Castelo, opta por eliminar a sua rota default. Adicione as rotas necessárias para que o Castelo2 continue a ter acesso ao Condado Portucalense e à rede Institucional. Mostre que a conectividade é restabelecida, assim como a tabela de encaminhamento resultante. Explicite ainda a utilidade de uma rota default.

```
root@Castelo2:/tmp/pycore.39449/Castelo2.conf# route del default
root@Castelo2:/tmp/pycore.39449/Castelo2.conf# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.68.70.192   0.0.0.0         255.255.255.248 U     0      0        0 eth0
root@Castelo2:/tmp/pycore.39449/Castelo2.conf#
```

Resposta:
Primeiramente, eliminamos a rota default ***"route del default"***.
De seguida, adicionamos a rota necessária para restabelecer a conectividade com o Condado.

```
<9449/Castelo2.conf# ip route add 192.168.0.224/29 via 172.68.70.193
root@Castelo2:/tmp/pycore.39449/Castelo2.conf# ping 192.168.0.227
PING 192.168.0.227 (192.168.0.227) 56(84) bytes of data.
64 bytes from 192.168.0.227: icmp_seq=1 ttl=62 time=0.171 ms
64 bytes from 192.168.0.227: icmp_seq=2 ttl=62 time=0.106 ms
64 bytes from 192.168.0.227: icmp_seq=3 ttl=62 time=0.092 ms
^C
--- 192.168.0.227 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.092/0.123/0.171/0.034 ms
root@Castelo2:/tmp/pycore.39449/Castelo2.conf#
```

Podemos então ver que 192.168.0.224/27 engloba todas as sub-redes. Podemos então aplicar supernetting. Fazemos então o comando ***"route add -net 192.168.0.224 netmask 255.255.255.224 gw 172.68.70.193"***

Obtemos então a seguinte tabela de encaminhamento :

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.68.70.192   0.0.0.0         255.255.255.248 U     0      0        0 eth0
192.168.0.224   172.68.70.193   255.255.255.224 UG    0      0        0 eth0
```

A **rota default** é uma regra de roteamento essencial em redes IP, que define para onde um dispositivo (como um computador ou ***router***) deve enviar pacotes quando **não se conhece**

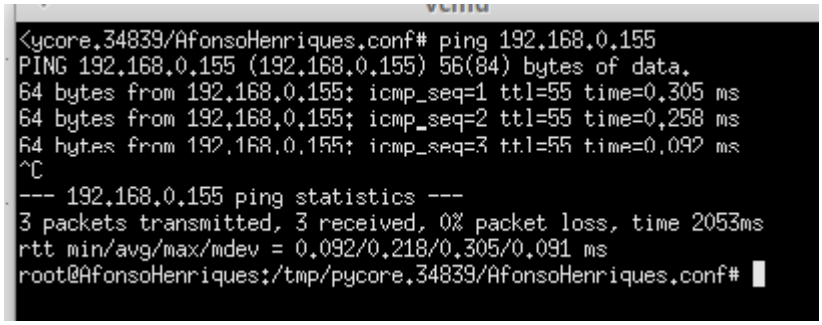
um caminho específico para o destino. A sua principal utilidade é garantir a conectividade com redes externas ou redes não diretamente conectadas.

2.

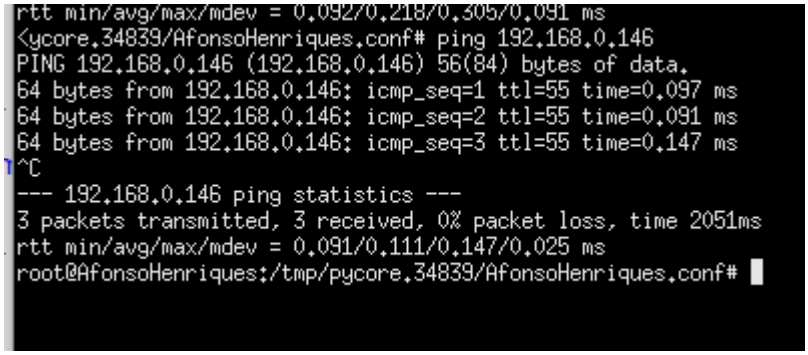
a) **Questão**: Confirme, através do comando ping, que AfonsoHenriques tem efetivamente conectividade com os servidores Reddit e Twitch.

Resposta:

Reddit:

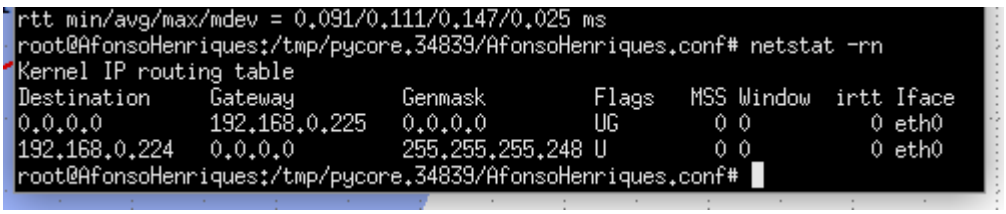


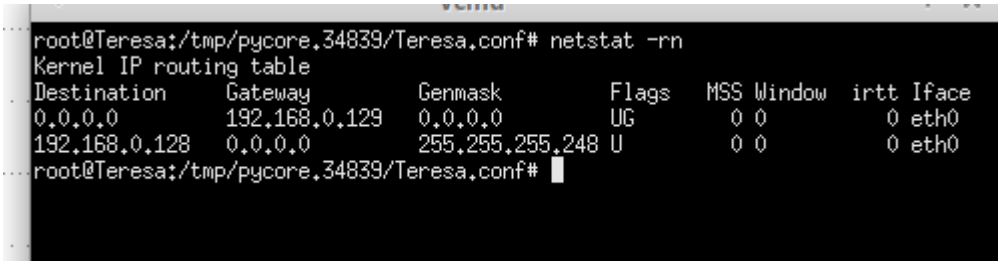
Twitch:



b) **Questão**: Recorrendo ao comando netstat -rn, analise as tabelas de encaminhamento dos dispositivos AfonsoHenriques e Teresa. Existe algum problema com as suas entradas? Identifique e descreva a utilidade de cada uma das entradas destes dois hosts.

Resposta:





Após analisar as tabelas de encaminhamento, não encontramos nenhum problema.
Tomando como exemplo a primeira tabela :
1- A primeira entrada,(Destination : 0.0.0.0 Gateway : 192.168.0.225) corresponde à ***rota Default.*** É através desta entrada que nos permite encaminhar pacotes para redes externas, e sem ela não iriamos conseguir sair da sub-rede local.

2- A segunda entrada (Destination: 192.168.0.224 Gateway : 0.0.0.0) corresponde à rota que nos permite ter acesso local, como podemos verificar pelo destino 192.168.0.224/29 que corresponde à rede local.

c) **Questão**: Analise o comportamento dos routers do core da rede (n1 a n6) quando tenta estabelecer comunicação entre os hosts AfonsoHenriques e Teresa. Indique que dispositivo(s) não permite(m) o encaminhamento correto dos pacotes. Seguidamente, avalie e explique a(s) causa(s) do funcionamento incorreto do dispositivo. Utilize o comando ip route add/del para adicionar as rotas necessárias ou remover rotas incorretas. Verifique a sintaxe completa do comando a usar com man ip-route ou man route. Poderá também utilizar o comando traceroute para se certificar do caminho nó a nó. Considere a alínea resolvida assim que houver tráfego a chegar ao ISP CondadOnline.

Resposta:
Fazendo ***"traceroute -I 192.168.0.130"*** vemos que o pacote chega ao n2, indicado nele algum problema.

1- Analisando a tabela de encaminhamento, vimos que existia uma ligação para D.Teresa, e outra para a rede onde está D.Teresa .No entanto, a com destino a D.Teresa estava errada pois o Gateway era 10.0.0.25, sendo então mapeado para ele mesmo. Por outro, a ligaçao para a rede estava correta.

```
root@n2:/tmp/pycore.39449/n2.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.0.0        10.0.0.13       255.255.255.252 UG        0 0          0 eth1
10.0.0.4        10.0.0.21       255.255.255.252 UG        0 0          0 eth0
10.0.0.8        10.0.0.13       255.255.255.252 UG        0 0          0 eth1
10.0.0.12       0.0.0.0         255.255.255.252 U         0 0          0 eth1
10.0.0.16       10.0.0.13       255.255.255.252 UG        0 0          0 eth1
10.0.0.20       0.0.0.0         255.255.255.252 U         0 0          0 eth0
10.0.0.24       0.0.0.0         255.255.255.252 U         0 0          0 eth2
10.0.0.28       10.0.0.26       255.255.255.252 UG        0 0          0 eth2
172.0.0.0       10.0.0.26       255.0.0.0       UG        0 0          0 eth2
172.16.142.0    10.0.0.13       255.255.255.252 UG        0 0          0 eth1
172.16.142.4    10.0.0.21       255.255.255.252 UG        0 0          0 eth0
172.16.143.0    10.0.0.26       255.255.255.252 UG        0 0          0 eth2
172.16.143.4    10.0.0.26       255.255.255.252 UG        0 0          0 eth2
192.168.0.128   10.0.0.13       255.255.255.248 UG        0 0          0 eth1
192.168.0.130   10.0.0.25       255.255.255.254 UG        0 0          0 eth2
192.168.0.136   10.0.0.21       255.255.255.248 UG        0 0          0 eth0
192.168.0.144   10.0.0.21       255.255.255.248 UG        0 0          0 eth0
192.168.0.152   10.0.0.21       255.255.255.248 UG        0 0          0 eth0
192.168.0.224   10.0.0.26       255.255.255.248 UG        0 0          0 eth2
192.168.0.232   10.0.0.26       255.255.255.248 UG        0 0          0 eth2
192.168.0.240   10.0.0.26       255.255.255.248 UG        0 0          0 eth2
192.168.0.248   10.0.0.26       255.255.255.248 UG        0 0          0 eth2
```

Realizamos então, o seguinte comando "***route del -net 192.168.0.130 netmask 255.255.255.254"***, eliminando assim a ligação para D.Teresa.

2- Reparamos depois ao analisar a tabela do n1, que o Gateway também estava errado . Fizemos então os seguintes comandos : ***"route del -net 192.168.0.128 netmask 255.255.255.248"*** e de seguida ***"route add -net 192.168.0.128 netmask 255.255.255.248 gw 10.0.0.9"***

```
<1.conf# route del -net 192.168.0.128 netmask 255.255.255.248
<1.conf# route add -net 192.168.0.128 netmask 255.255.255.248 gw 10.0.0.9
root@n1:/tmp/pycore.43617/n1.conf#
```

3- Voltamos então a fazer o comando traceroute e vimos que os pacotes chegavam agora ao n3 mas não avançaram. Reparamos então que faltava na tabela de encaminhamento, rota para a rede de D.Teresa. Fizemos então o comando ***"route add -net 192.168.0.128 netmask 255.255.255.248 gw 10.0.0.5"***

4- Agora, se realizarmos o comando no AfonsoHenriques ***"ping 10.0.0.1"*** vemos que os pacotes já chegam ao CondadOnline.

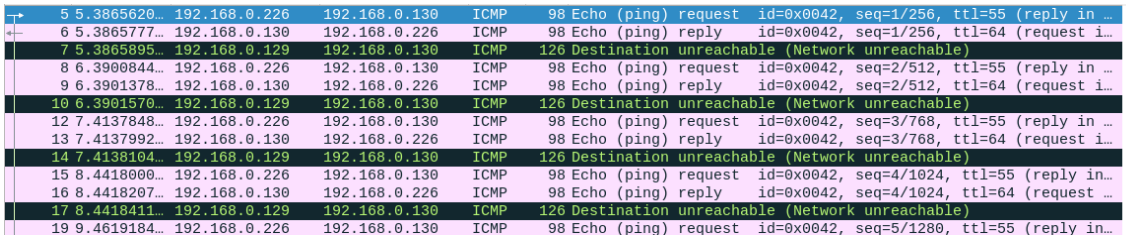
```
root@AfonsoHenriques:/tmp/pycore.39449/AfonsoHenriques.conf# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=57 time=0.251 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=57 time=0.134 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=57 time=0.195 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2044ms
rtt min/avg/max/mdev = 0.134/0.193/0.251/0.047 ms
root@AfonsoHenriques:/tmp/pycore.39449/AfonsoHenriques.conf#
```

Concluindo, os dispositivos que impediam o encaminhamento dos dados eram o n2,n1 e o n4. No n2, a rota para D.Teresa estava errada, porém a rota para a rede estava correta.No n1, a rota para a rede estava mal definida voltando para trás. No n4, não havia rota para a rede de D.Teresa nem para D.Teresa.

d) **Questão**: Uma vez que o core da rede esteja a encaminhar corretamente os pacotes enviados por AfonsoHenriques, confira com o Wireshark se estes são recebidos por Teresa.

i) **Questão**: Em caso afirmativo, porque é que continua a não existir conectividade entre D.Teresa e D.Afonso Henriques? Efetue as alterações necessárias para garantir que a conectividade é restabelecida e o confronto entre os dois é evitado.

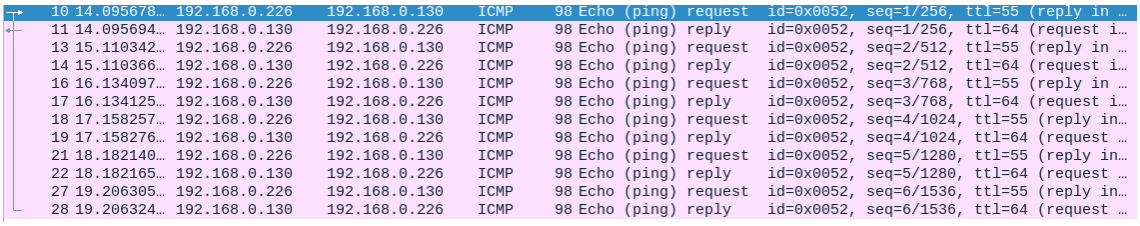
Resposta:



5	5.3865620…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0042, seq=1/256, ttl=55 (reply in …
6	5.3865777…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0042, seq=1/256, ttl=64 (request i…
7	5.3865895…	192.168.0.129	192.168.0.130	ICMP	126	Destination unreachable (Network unreachable)
8	6.3900844…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0042, seq=2/512, ttl=55 (reply in …
9	6.3901378…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0042, seq=2/512, ttl=64 (request i…
10	6.3901570…	192.168.0.129	192.168.0.130	ICMP	126	Destination unreachable (Network unreachable)
12	7.4137848…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0042, seq=3/768, ttl=55 (reply in …
13	7.4137992…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0042, seq=3/768, ttl=64 (request i…
14	7.4138104…	192.168.0.129	192.168.0.130	ICMP	126	Destination unreachable (Network unreachable)
15	8.4418000…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0042, seq=4/1024, ttl=55 (reply in…
16	8.4418207…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0042, seq=4/1024, ttl=64 (request …
17	8.4418411…	192.168.0.129	192.168.0.130	ICMP	126	Destination unreachable (Network unreachable)
19	9.4619184…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0042, seq=5/1280, ttl=55 (reply in…

Como podemos ver, D.Teresa recebe os pacotes mas não os consegue enviar de volta para D.Afonso Henriques.

Analisando a tabela de encaminhamento do RAGaliza vemos que falta definir a rota para a rede de D.Afonso Henriques. Adicionamos então fazendo o comando ***"route add -net 192.168.0.224 netmask 255.255.255.248 gw 172.16.142.1"***

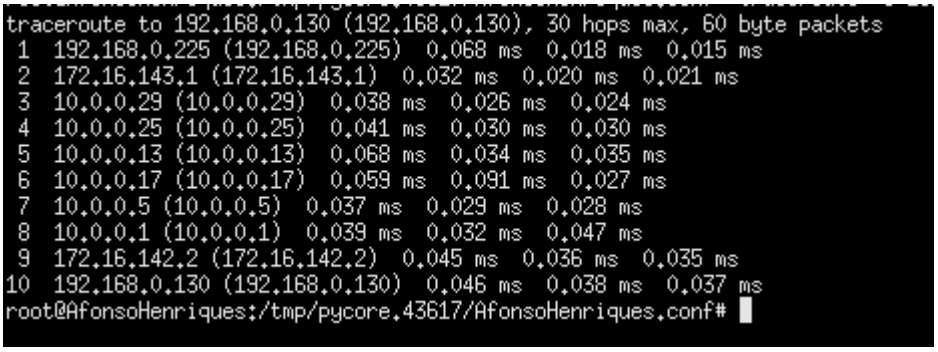


10	14.095678…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0052, seq=1/256, ttl=55 (reply in …
11	14.095694…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0052, seq=1/256, ttl=64 (request i…
13	15.110342…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0052, seq=2/512, ttl=55 (reply in …
14	15.110366…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0052, seq=2/512, ttl=64 (request i…
16	16.134097…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0052, seq=3/768, ttl=55 (reply in …
17	16.134125…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0052, seq=3/768, ttl=64 (request i…
18	17.158257…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0052, seq=4/1024, ttl=55 (reply in…
19	17.158276…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0052, seq=4/1024, ttl=64 (request …
21	18.182140…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0052, seq=5/1280, ttl=55 (reply in…
22	18.182165…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0052, seq=5/1280, ttl=64 (request …
27	19.206305…	192.168.0.226	192.168.0.130	ICMP	98	Echo (ping) request id=0x0052, seq=6/1536, ttl=55 (reply in…
28	19.206324…	192.168.0.130	192.168.0.226	ICMP	98	Echo (ping) reply id=0x0052, seq=6/1536, ttl=64 (request …

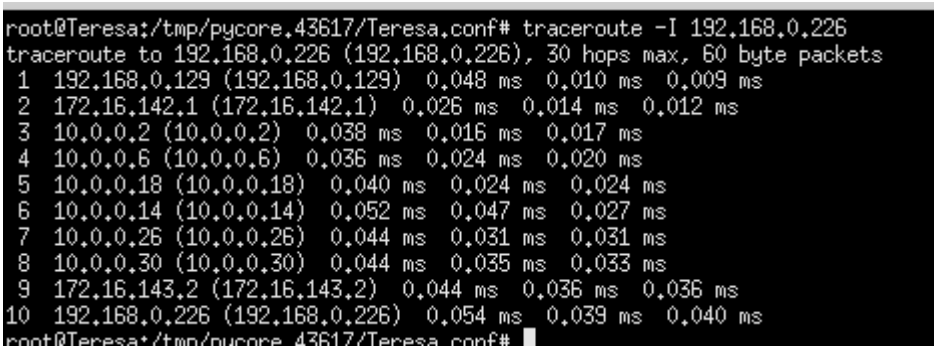
Agora sim, a conectividade foi restabelecida como podemos ver :

ii) **Questão**: As rotas dos pacotes ICMP echo reply são as mesmas, mas em sentido inverso, que as rotas dos pacotes ICMP echo request enviados entre AfonsoHenriques e Teresa? (Sugestão: analise as rotas nos dois sentidos com o traceroute). Mostre graficamente a rota seguida nos dois sentidos por esses pacotes ICMP.

Resposta: O caminho de D.Afonso Henriques até D.Teresa é



O caminho de D.Teresa até D.Afonso Henriques é :



Concluímos então que as rotas dos pacotes ICMP echo reply não são as mesmas que as rotas dos pacotes ICMP echo request enviados pois diferem em 1 router. A ligação Afonso-Teresa usa o router n1 como ligação entre n2 e n3 .
Por outro lado, a ligação Teresa- Afonso usa o n4 como ligação entre n3 e n2.

e) **Questão**: Estando restabelecida a conectividade entre os dois hosts, obtenha a tabela de encaminhamento de n5 e foque-se na seguinte entrada:

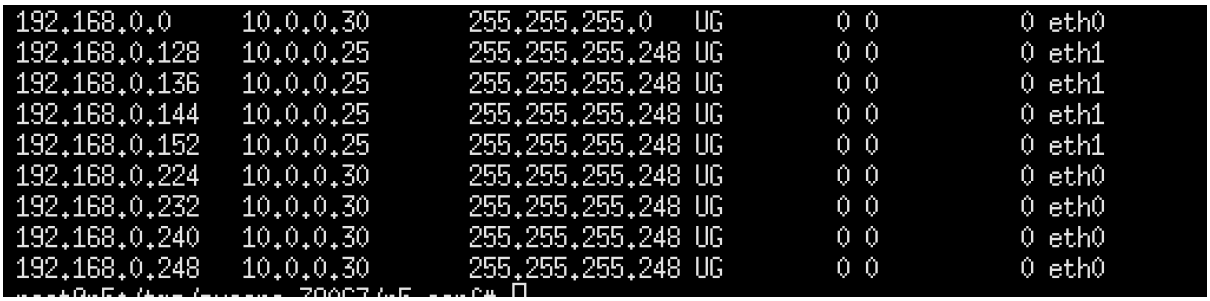
ip route 192.168.0.0 255.255.255.0 10.0.0.30

Existe uma correspondência (match) nesta entrada para pacotes enviados para o polo Galiza? E para CDN? Caso seja essa a entrada utilizada para o encaminhamento, permitirá o funcionamento esperado do dispositivo? Ofereça uma explicação pela qual essa entrada é ou não utilizada.

Resposta: 192.168.0.0/24 permite correspondência para o polo Galiza (192.168.0.1298) e para o polo CDN (192.168.0.153 | 192.168.0.145 | 192.168.137) pois tem uma faixa de Ips válidos de 192.0.168.0.1 - 192.0.168.254) . No entanto, os pacotes não chegariam lá

pois a gateway é 10.0.0.30, logo não permitirá caso seja utilizada o funcionamento do dispositivo.

Analisando a tabela, vemos que a entrada não é utilizada pois apresenta ips maiores e direcionados às redes do Condado, Institucional, CDN e Galiza que serão então priorizadas.



f) **Questão:** Os endereços utilizados pelos quatro polos são endereços públicos ou privados? E os utilizados no core da rede/ISPs? Justifique convenientemente.

Resposta: RFC 1918 define intervalos específicos para endereços privados que não devem aparecer na internet pública. Especificamente,

- 10.0.0.0 - 10.255.255.255 (10/8 prefixo)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefixo)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefixo)

Por convenção, nenhum dispositivo público usa esses endereços.

Logo, todos os endereços utilizados pelos quatro pólos e pelo core da rede/ISPs, como estão todos nos intervalos especificados para endereços privados definidos no RFC 1928, são considerados endereços privados.

g) Questão: Os switches localizados em cada um dos polos têm um endereço IP atribuído? Porquê?

Resposta: Os switches localizados em cada polo não têm endereço IP atribuído por defeito, pois são dispositivos de camada 2, que operam com base em endereços MAC para encaminhar pacotes na rede local (LAN) sem exigir configuração IP.

3.

a) **Questão**: De modo a facilitar a travessia, elimine as rotas referentes a Galiza e CDN no dispositivo n6 e defina um esquema de sumarização de rotas (Supernetting) que permita o uso de apenas uma rota para ambos os polos. Confirme que a conectividade é mantida.

Resposta: Comecemos então por eliminar a rotas para o CDN e Galiza :

```
<6.conf# route del -net 192.168.0.152 netmask 255.255.255.248
<6.conf# route del -net 192-168.0.144 netmask 255.255.255.248
192-168.0.144: Host name lookup failure
<6.conf# route del -net 192.168.0.144 netmask 255.255.255.248
<6.conf# route del -net 192.168.0.136 netmask 255.255.255.248
<6.conf# route del -net 192.168.0.128 netmask 255.255.255.248
root@n6:/tmp/pycore.39067/n6.conf#
```

De seguida adicionamos uma rota para o endereço 192.168.0.128/27 usando o comando ***"route add -net 192.168.0.128 netmask 255.255.255.224 gw 10.0.0.1"***

```
ping: connect: Network is unreachable
<6.conf# route add -net 192.168.0.128 netmask 255.255.255.224 gw 10.0.0.1
root@n6:/tmp/pycore.39067/n6.conf# ping 192.168.0.132
PING 192.168.0.132 (192.168.0.132) 56(84) bytes of data.
64 bytes from 192.168.0.132: icmp_seq=1 ttl=62 time=0.088 ms
64 bytes from 192.168.0.132: icmp_seq=2 ttl=62 time=0.137 ms
64 bytes from 192.168.0.132: icmp_seq=3 ttl=62 time=0.082 ms
^C
--- 192.168.0.132 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.082/0.102/0.137/0.024 ms
```

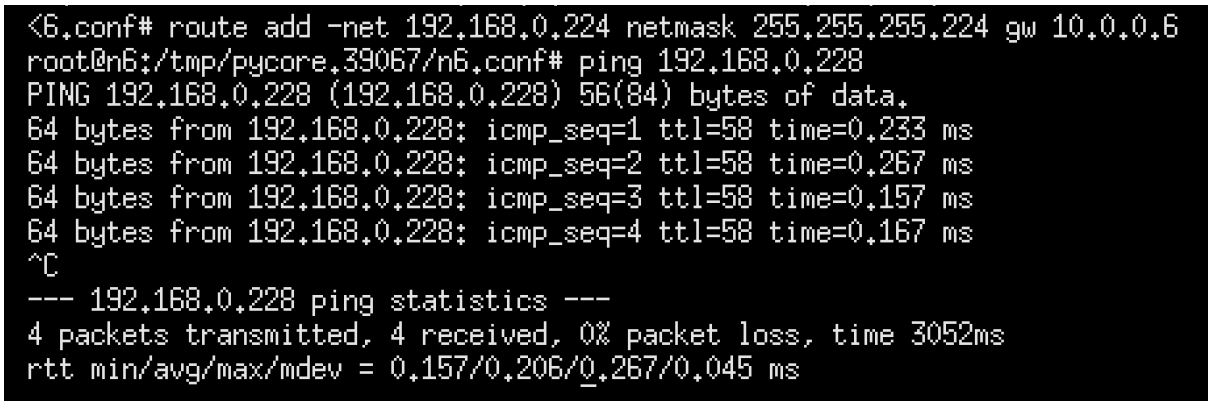
```
root@n6:/tmp/pycore.39067/n6.conf# ping 192.168.0.155
PING 192.168.0.155 (192.168.0.155) 56(84) bytes of data.
64 bytes from 192.168.0.155: icmp_seq=1 ttl=62 time=0.122 ms
64 bytes from 192.168.0.155: icmp_seq=2 ttl=62 time=0.135 ms
64 bytes from 192.168.0.155: icmp_seq=3 ttl=62 time=0.138 ms
64 bytes from 192.168.0.155: icmp_seq=4 ttl=62 time=0.081 ms
^C
--- 192.168.0.155 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.081/0.119/0.138/0.022 ms
root@n6:/tmp/pycore.39067/n6.conf#
```

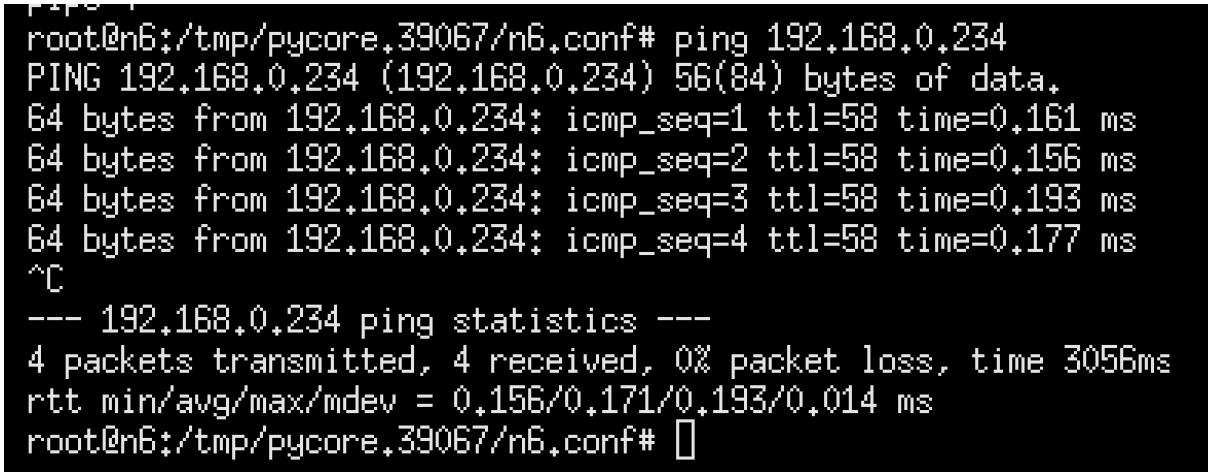
b) **Questão**: Repita o processo descrito na alínea anterior para CondadoPortucalense e Institucional, também no dispositivo n6.

Resposta: Comecemos então por eliminar as 4 rotas :

```
192.168.0.248   10.0.0.6        255.255.255.248 UG        0 0
<6.conf# route del -net 192.168.0.232 netmask 255.255.255.248
<6.conf# route del -net 192.168.0.240 netmask 255.255.255.248
<6.conf# route del -net 192.168.0.248 netmask 255.255.255.248
<6.conf# route del -net 192.168.0.224 netmask 255.255.255.248
root@n6:/tmp/pycore.39067/n6.conf#
```

De seguida adicionamos a rota 192.168.0.224/27 usando o comando ***"route add -net 102.168.0.224 netmask 255.255.255.224 gw 10.0.0.6"***





c) **Questão**: Comente os aspetos positivos e negativos do uso do Supernetting.

Resposta: Vantagens:

1. Permite lookups mais rápidos, já que as tabelas de endereçamento são mais pequenas => isto acontece pois a base do wwww é agrupar redes menores (sub-redes) em redes maiores, o que também reduz o desperdício de endereços;
2. Com esta agregação, reduz-se também a quantidade de rotas que se precisam atualizar e fazer manutenção, baixando a quantidade de recursos necessários para manter a rede a funcionar;
3. Também com isso, é mais fácil para redes escalarem sem aumentarem as suas tabelas de endereçamento em demasia, pois com o supernetting uma única entrada na tabela de endereçamento cobre mais endereços IP do que cobriria individualmente sem o supernetting.

Desvantagens:

1. Complexo para implementar, se mal implementado pode causar conflitos de endereçamento e problemas de encaminhamento, e precisa de um planeamento cuidadoso das sub-redes a agregar (é necessário serem contíguas).

2. Pode dificultar a segmentação da rede, pois ao juntar várias sub-redes num único bloco, perde-se a granularidade no controlo de tráfego, segurança (ex.: filtragem por sub-rede), e políticas de QoS (*Quality of Service*).
3. Depende de routers modernos (ou, pelo menos, capazes de lidar corretamente com supernetting) para funcionar em pleno, não sendo propriamente, por isso, compatível com routers antigos.