Redes de computadores

Protocolo IPv4: Datagramas IP e Fragmentação

Pedro Calheno Pinto, Diogo do Rego Neto, and Samuel Macieira Ferreira

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a87983,a98197,a100654}@alunos.uminho.pt

Parte I

Questão 1 - Prepare uma topologia CORE para verificar o comportamento do traceroute. Na topologia deve existir: um host (pc) cliente designado Lost, cujo router de acesso é RA1; o router RA1 está simultaneamente ligado a dois routers no core da rede RC1 e RC2; estes estão conectados a um router de acesso RA2, que por sua vez, se liga a um host (servidor) designado Found. Ajuste o nome dos equipamentos atribuídos por defeito para o enunciado. Apenas nas ligações (links) da rede de core, estabeleça um tempo de propagação de 15 ms. Após ativar a topologia, note que pode não existir conectividade IP imediata entre Lost e Found até que o anúncio de rotas entre routers estabilize. .

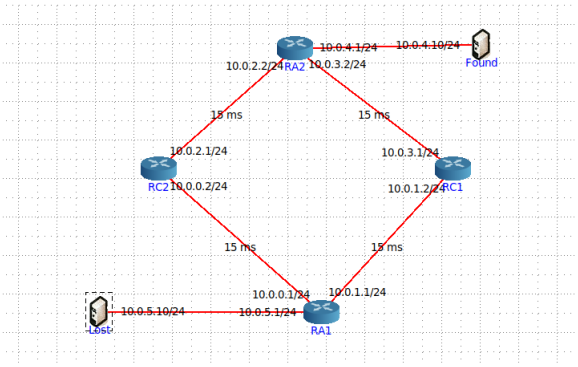

Fig. 1. Topologia core.

Alínea a - 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.

31	50.012728009	10.0.5.1	224.0.0.5	OSPF	78 Hello Packet
32	52.013823138	10.0.5.1	224.0.0.5	OSPF	78 Hello Packet
33	53.298149243	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=1/256, ttl=1 (no response…
34	53.298177858	10.0.5.1	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
35	53.298186364	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=2/512, ttl=1 (no response…
36	53.298190431	10.0.5.1	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
37	53.298193708	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=3/768, ttl=1 (no response…
38	53.298197184	10.0.5.1	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
39	53.298200771	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=4/1024, ttl=2 (no respons…
40	53.298211992	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=5/1280, ttl=2 (no respons…
41	53.298215950	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=6/1536, ttl=2 (no respons…
42	53.298219847	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=7/1792, ttl=3 (no respons…
43	53.298223204	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=8/2048, ttl=3 (no respons…
44	53.298226770	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=9/2304, ttl=3 (no respons…
45	53.298230227	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=10/2560, ttl=4 (reply in …
46	53.298233343	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=11/2816, ttl=4 (reply in …
47	53.298236539	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=12/3072, ttl=4 (reply in …
48	53.298239995	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=13/3328, ttl=5 (reply in …
49	53.298243262	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=14/3584, ttl=5 (reply in …
50	53.298246538	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=15/3840, ttl=5 (reply in …
51	53.298249884	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=16/4096, ttl=6 (reply in …
52	53.298614986	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=17/4352, ttl=6 (reply in …
53	53.298622280	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=18/4608, ttl=6 (reply in …
54	53.298626247	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=19/4864, ttl=7 (reply in …
55	53.328382515	10.0.0.2	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
56	53.328386042	10.0.0.2	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
57	53.328386603	10.0.0.2	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
58	53.328654350	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=20/5120, ttl=7 (reply in …
59	53.328665281	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=21/5376, ttl=7 (reply in …
60	53.328669168	10.0.5.10	10.0.4.10	ICMP	74 Echo (ping) request id=0x0030, seq=22/5632, ttl=8 (reply in …
61	53.359428677	10.0.2.2	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
62	53.359433285	10.0.2.2	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
63	53.359433856	10.0.2.2	10.0.5.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
64	53.359434418	10.0.4.10	10.0.5.10	ICMP	74 Echo (ping) reply id=0x0030, seq=10/2560, ttl=61 (request …
65	53.359435049	10.0.4.10	10.0.5.10	ICMP	74 Echo (ping) reply id=0x0030, seq=11/2816, ttl=61 (request …
66	53.359435660	10.0.4.10	10.0.5.10	ICMP	74 Echo (ping) reply id=0x0030, seq=12/3072, ttl=61 (request …

Fig. 2. Captura de tráfego do Wireshark após "traceroute -I IP_Found"

Ao analisar o tráfego ICMP enviado pelo sistema Lost e o tráfego ICMP recebido como resposta, é possível identificar os dispositivos intermediários ao longo da rota. Como nas ligações da rede do core estabelecemos um tempo de propagação (delay) de 15 ms, o tempo necessário para o pacote percorrer cada segmento irá ser maior, vemos também que o TTL tem um valor crescente até obter uma resposta do Found.

Alínea b - Qual deve ser o valor inicial mínimo do campo TTL para alcançar o servidor Found? Verifique na prática que a sua resposta está correta.

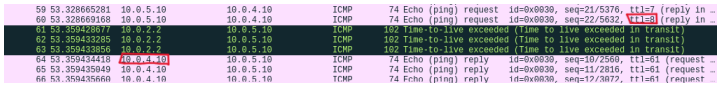

Fig. 3. Captura de tráfego do Wireshark após "traceroute -I IP_Found"

Para alcançar o Lost, o valor mínimo de TTL deve ser 8. O tráfego ICMP da figura anterior mostra-nos que o Found apenas recebe uma resposta do Lost quando o valor de TTL chega a 8.

Alínea c - 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.

Para calcular o tempo médio de ida-e-volta (RTT - Round-Trip Time), usamos os valores da ultima linha do traceroute.

$$\text{RTT} = \left(\frac{61.206+61.203+61.201}{3}\right) \simeq 61,203$$

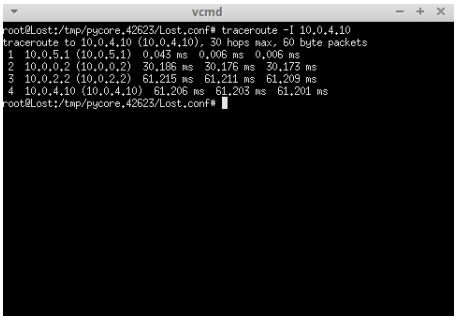

Fig. 4. Shell bash do host Lost após "traceroute -I IP_Found""

Alínea d - 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?

Não, pois o tempo de ida e o tempo de volta podem variar bastante entre si, e por isso o calculo do One-Way Delay não teria muita precisão.

Questão 2 - Pretende-se agora usar o traceroute na sua máquina nativa e gerar datagramas IP de diferentes tamanhos.

Usando o wireshark capture o tráfego gerado pelo traceroute sem especificar o tamanho do pacote, i.e., quando é usado o tamanho do pacote de prova por defeito. Utilize como máquina destino o host marco.uminho.pt. Pare a captura. Com base no tráfego capturado, identifique os pedidos ICMP Echo Request e o conjunto de mensagens devolvidas como resposta.nativa e gerar datagramas IP de diferentes tamanhos.

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

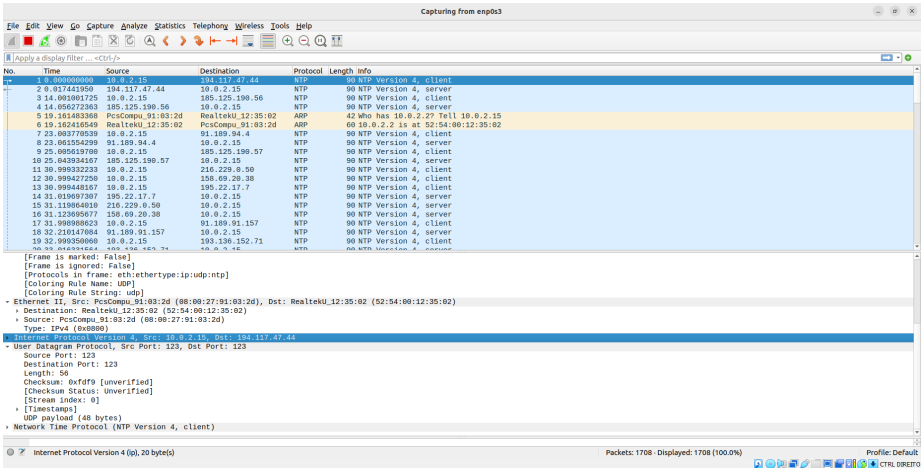

Fig. 5. Captura de Tráfego em Linux através do Wireshark

O endereço IP da interface ativa do computador é 10.0.2.15

Alínea b - Qual é o valor do campo protocol? O que permite identificar?

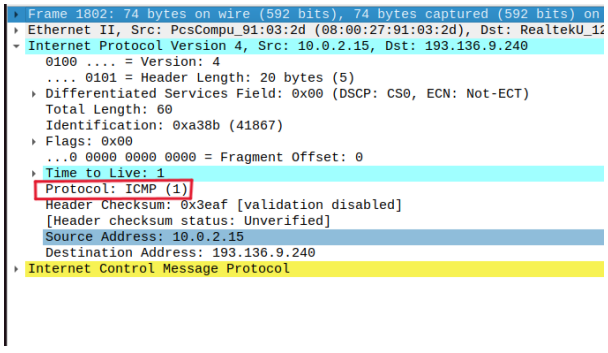

Fig. 6. Protocol : ICMP

O valor do campo protocolo e 1 e identifica que o protocolo usado é o ICMP. Este é usado para enviar mensagens de controlo e erros na rede.

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

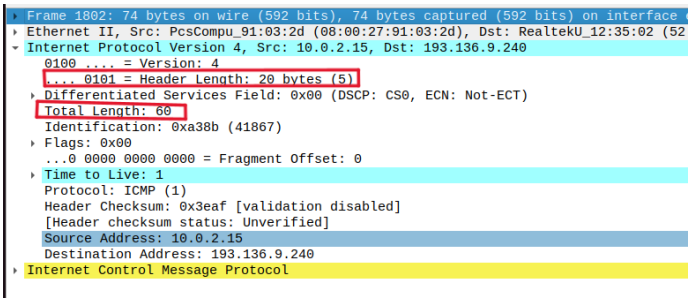

Fig. 7. Bytes do cabeçalho e payload

O cabeçalho IPv4 possui 20 bytes. O campo de dados (payload) do datagrama possui 40 bytes. Este valor pode ser calculado subtraindo o tamanho total, 60 bytes pelo tamanho do cabeçalho, 20 bytes.

Payload (Bytes) = 60 - 20 = 40

Alínea d - O datagrama IP foi fragmentado? Justifique.

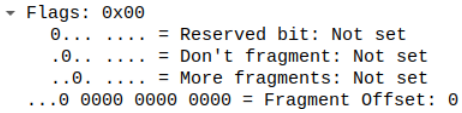

Fig. 8. Fragment offset

Podemos ver que tanto o campo "Fragment offset" como o campo "Flags" têm o valor 0 (0x00), logo ou este datagrama não foi fragmentado ou este é o primeiro fragmento. Porém, como a flag "More fragments" tem valor 0, podemos concluir que é a última parte do pacote e o datagrama não foi fragmentado.

Alínea e - 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 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.

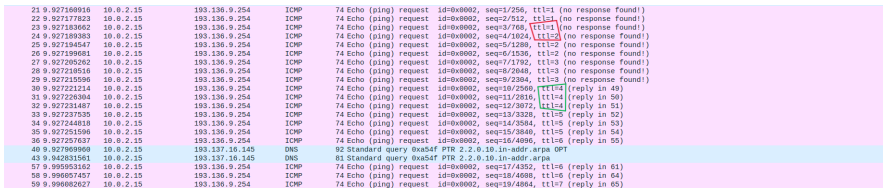

Fig. 9. Sequência Tráfego ICMP gerada no envio de mensagens da nossa máquina com endereço IP 10.0.2.15

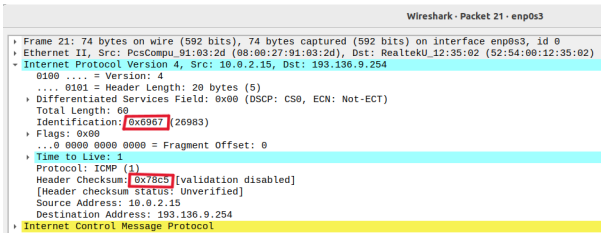

Fig. 10. Pacote 21

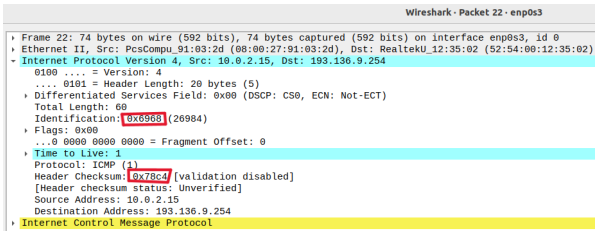

Fig. 11. Pacote 22

Podemos verificar abrindo os detalhes de cada pacote, que os campos que variam de pacote para pacote são ”Identification”, ”Time to live” e ”Header checksum”.

Alínea f - Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?

Nas figuras da alínea *e* podemos verificar que do pacote 21 para o pacote 22, o campo "Identification" incrementa 1 unidade. No campo TTL verificam-se incrementos de 1 unidade a cada 3 pacotes.

Alínea g - Ordene o tráfego capturado por endereço destino e encontre a série de respostas ICMP TTL Exceeded enviadas ao seu computador.

I - 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?

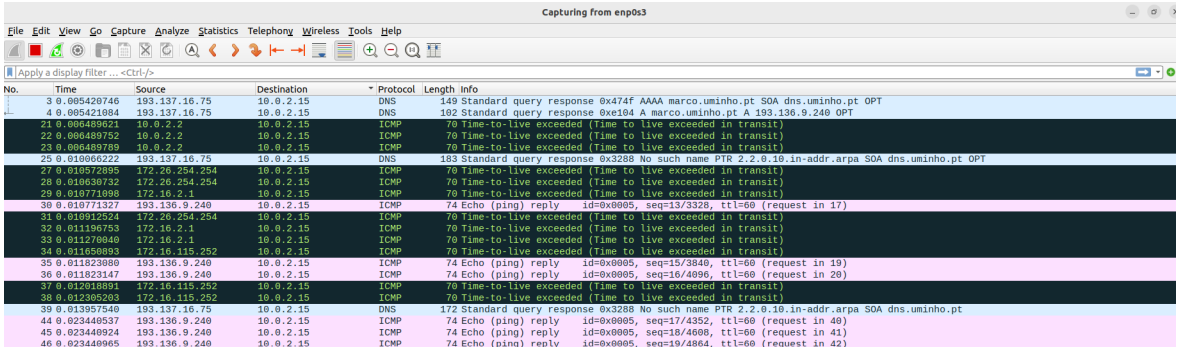

Fig. 12. Sequência Tráfego ICMP gerada com a receção de mensagens na nossa máquina com endereço IP 10.0.2.15

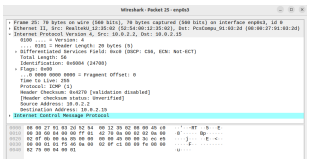

Fig. 13. Pacote com TTL = 255

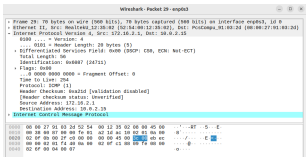

Fig. 14. Pacote com TTL = 254

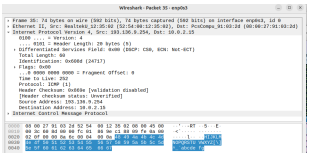

Fig. 15. Pacote com TTL = 252

O valor do campo TTL não se mantém constante, varia entre 252 e 255. Quando é preciso enviar uma mensagem de erro, geralmente é gerada por routers mais distantes, o que faz com que o pacote de regresso tenha que passar por mais routers intermédios. Isso acaba por diminuir o valor do campo TTL nas mensagens de erro, o que explica as diferenças nos valores de TTL apresentados para cada pacote.

II - Porque razão as mensagens de resposta ICMP TTL Exceeded são sempre enviadas na origem com um valor TTL relativamente alto?

Essas mensagens de resposta ICMP são enviadas na origem com um valor TTL relativamente alto porque o objetivo é evitar que as mensagens de erro "TTL Exceeded" sejam descartadas durante o caminho de volta ao host de origem. Se o valor TTL das mensagens de resposta fosse definido como 1, elas seriam descartadas pelo primeiro router que encontrassem, e o host de origem não seria capaz de receber a mensagem de erro. Por isso, as mensagens de resposta ICMP "TTL Exceeded" geralmente são enviadas com um valor TTL maior do que o valor máximo esperado para o caminho de volta, para garantir que a mensagem de erro alcance o host de origem.

Alínea h - Sabendo que o ICMP é um protocolo pertencente ao nível de rede, discuta se a informação contida no cabeçalho ICMP poderia ser incluída no cabeçalho IPv4? Quais seriam as vantagens/desvantagens resultantes dessa hipotética inclusão?

Na teoria, é possivel incluir informações contidas no cabeçalho ICMP diretamente no cabeçalho IPV4, mas isso não seria uma solução recomendada.

As vantagens que teríamos seriam a redução do tamanho do cabeçalho ICMP: O cabeçalho ICMP pode ser relativamente grande, especialmente quando as mensagem ICMP contêm dados adicionais, como o cabeçalho original do pacote IP. Ao incluir algumas informações no cabeçalho IPV4, é possivel reduzir o tamanho do cabeçalho ICMP, economizando assim espaço no pacote e também uma melhor facilidade na análise de tráfego.

Porém existem algumas desvantagens como a maior complexidade ao incluir informações do cabeçalho ICMP no cabeçalho IPv4, que aumentaria a complexidade do processamento de pacotes IP e poderia dificultar a implementação de protocolos de rede mais avançados e a falta de flexibilidade, já que o ICMP é um protocolo flexível e pode ser facilmente modificado ou estendido para atender a requisitos específicos. Ao incluir algumas informações do cabeçalho ICMP no cabeçalho IPv4, pode-se perder a capacidade de estender o protocolo ICMP de forma independente do IP.

Questão 3 -

Alínea a - Localize a primeira mensagem ICMP. Porque é que houve necessidade de fragmentar o pacote inicial?

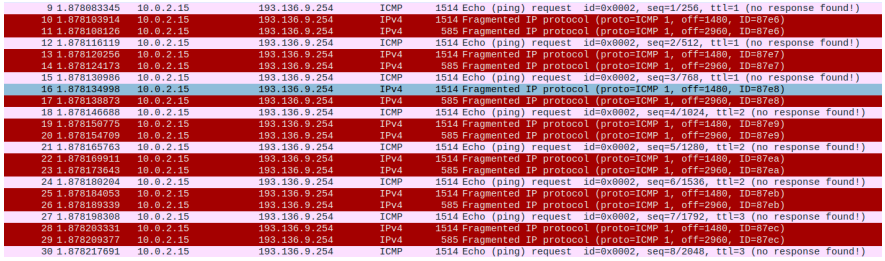

Fig. 16.

A primeira mensagem ICMP foi fragmentada porque o seu tamanho era demasiado grande para ser transmitida de uma só vez. Devido ao MTU (Max Transfer Unit) ser 1500 bytes e a mensagem ter 3531 (3500+31) bytes.

Alínea b - Imprima o primeiro fragmento do datagrama IP original. 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?

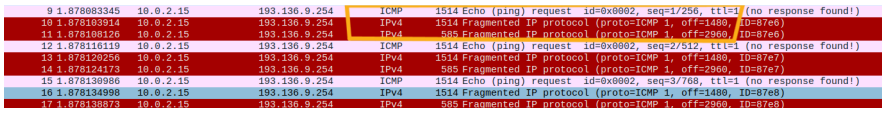

Fig. 17.

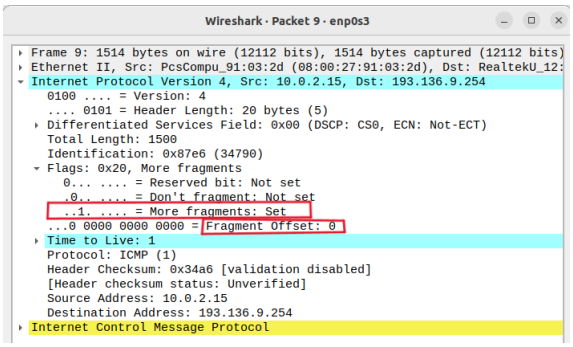

Fig. 18. 1º Fragmento (ICMP)

O primeiro fragmento do datagrama original tem protocolo ICMP em vez de IPV4 e no seu cabeçalho conseguimos ter indicação de que o pacote foi fragmentado a partir das flags. A flag “More fragments” tem o seu bit a 1, logo podemos concluir que existem mais fragmentos associados a este datagrama. Como o campo “Fragment offset” tem o valor zero, podemos concluir que se trata do primeiro fragmento. O tamanho deste datagrama IP é 1500 bytes.

Alínea c - 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?

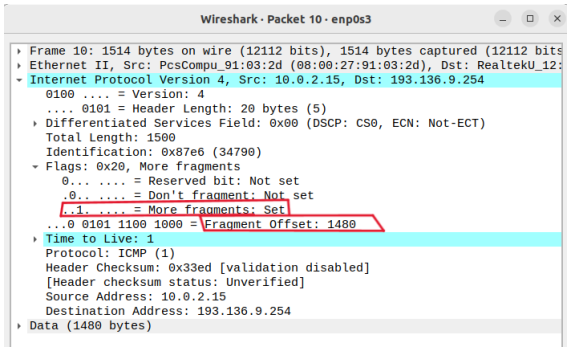

Fig. 19. 2º Fragmento (IPV4)

Com base na informação presente na captura de pacotes, podemos inferir que não estamos no primeiro fragmento do datagrama IP, uma vez que o valor do campo "Fragment Offset" é 1480 e não 0, indicando que já ocorreu a transferência de 1480 bytes do pacote original. Além disso, a presença da flag "More fragments" com valor igual a 1 indica que existem mais fragmentos a serem transmitidos para que o datagrama original possa ser reconstituído.

Alínea d - Estime teoricamente o número de fragmentos gerados a partir do datagrama IP original e o número de bytes transportados no último fragmento desse datagrama. Compare os dois valores estimados com os obtidos através do wireshark.

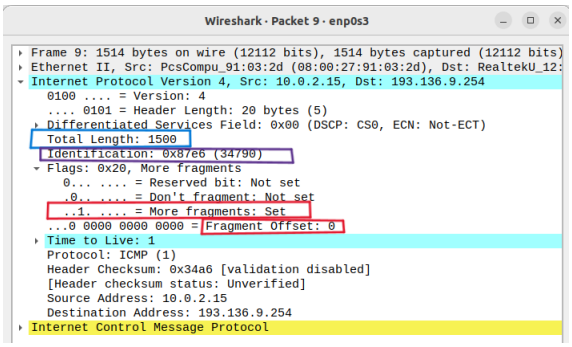

Fig. 20. 1º Fragmento (ICMP)

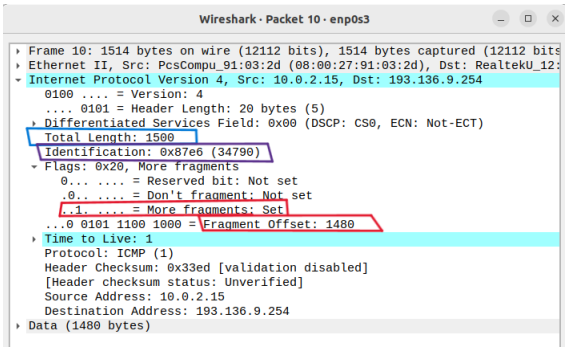

Fig. 21. 2º Fragmento (IPV4)

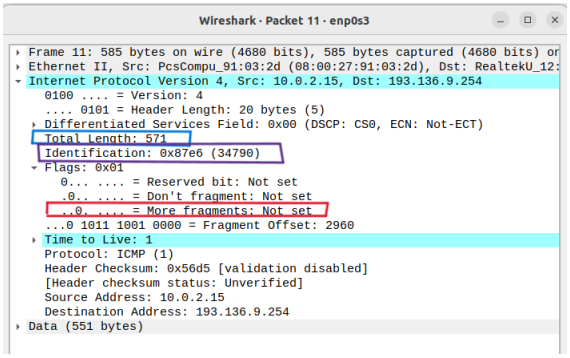

Fig. 22. 3º Fragmento (IPV4)

Três fragmentos foram criados a partir do datagrama original, pois apenas dois outros fragmentos têm um valor no campo "Identificação" igual ao primeiro fragmento. Nas imagem acima conseguimos verificar que os 3 fragmentos têm o mesmo valor no campo Identification, que é 0x87e6. O que significa que pertencem ao mesmo datagrama original e a flag "More fragments" tem o seu bit a 0, o que significa que este é o último fragmento. O número de bytes transportados no último fragmento seriam 531 bytes. Isto porque dos 3531 bytes do datagrama original, 1500 bytes são transportados pelo 1º fragmento, outros 1500 bytes são transportados pelo 2º fragmento, sobrando assim 531 bytes para o último fragmento.

Alínea e - Como se deteta o último fragmento correspondente ao datagrama original? Estabeleça um filtro no Wireshark que permita listar o último fragmento do primeiro datagrama IP segmentado.

Usamos a seguinte expressão lógica no filtro do wireshark : ip.src == 10.0.2.15 ip.frag offset gt 0 ip.flags.mf == 0

Alínea f - Identifique o equipamento onde o datagrama IP original é reconstruído a partir dos fragmentos. A reconstrução poderia ter ocorrido noutro equipamento diferente do identificado? Porquê?

O equipamento responsável por reconstruir o datagrama IP original a partir dos fragmentos é o destinatário final da comunicação, ou seja, o dispositivo de destino de rede. É neste equipamento que os fragmentos recebidos são reagrupados na ordem correta, utilizando as informações do cabeçalho IP que indicam a posição de cada fragmento no datagrama original.

Não é possível que a reconstrução ocorra noutro equipamento diferente do destinatário final, pois apenas este possui todas as informações necessárias para realizar a operação corretamente. Além disso, os fragmentos são enviados para o endereço IP do destinatário, e não para outro equipamento intermediário na rede.

Alínea g - 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.

Os campos que vão mudando entre diferentes fragmentos são:

length - será máxima em todos os fragmentos, excepto no ultimo, em que pode não chegar a esse valor, assim, pode permitir distinguir o ultimo fragmento dos restantes;
flag More fragments - que será 1 em todos menos no ultimo, pelo que nos permite distinguir de forma direta o ultimo fragmento;
offset - que corresponde ao número de bytes do datagrama inicial que já foram enviados, pelo que se ordenarmos os fragmentos por ordem crescente de offset podemos obter o datagrama ordenado pela ordem em que os dados estavam no datagrama original.

Alínea h - Por que razão apenas o primeiro fragmento de cada pacote é identificado como sendo um pacote ICMP?

Quando um problema ocorre na transmissão de um pacote fragmentado um pacote ICMP de erro é gerado para informar a origem da falha. Esse pacote ICMP contém informações sobre o endereço IP do host de origem e do host de destino, o número de identificador do pacote original e o número de fragmentos que apresentou o problema.

No entanto, como os fragmentos subsquentes contêm apenas parte dos dados do pacote original, estes não contêm todas as informações necessárias para a geração do pacote ICMP de erro. Por essa razão, apenas o primeiro fragmento de um pacote fragmentado é identificado como um pacote de ICMP de erro.

Alínea i - 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?

Quando o tamanho do datagrama é maior do que o MTU da rede, é necessário fragmentá-lo em pacotes menores para que possa ser transmitido pela rede. Para isso, o protocolo IP divide o datagrama em fragmentos menores e reagrupa-os novamente no destino final. O

cabeçalho IP contém informações que indicam a posição de cada fragmento no datagrama original.

Ao aumentar o valor máximo permitido para um datagrama sem fragmentação, a rede pode ter um melhor desempenho em termos de taxa de transferência, pois menos fragmentação é necessária, o que reduz o overhead de processamento e diminui o tempo de transmissão dos pacotes. No entanto, isso também pode levar a um aumento na probabilidade de perda de pacotes, já que pacotes maiores são mais propensos a serem descartados em caso de congestionamento na rede.

Por outro lado, ao diminuir o valor máximo permitido para um datagrama sem fragmentação, a rede pode ter uma melhor capacidade de lidar com pacotes de tamanhos variados, mas pode reduzir a taxa de transferência da rede, pois mais fragmentação será necessária, aumentando o overhead de processamento e aumentando o tempo de transmissão dos pacotes.

Alínea j - 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 <opção DF> <opção pkt_size> 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.

O valor máximo do SIZE sem que ocorra fragmentação depende do MTU da rede utilizada. O MTU é o tamanho máximo de dados que pode ser transmitido num único pacote numa rede. Se um pacote for maior que o MTU, ele será fragmentado em vários pacotes menores para ser trasmitido.

Depois de várias tentativas descobrimos que a valor máximo de SIZE será 1500 bytes, ou seja, o valor do MTU é 1528 bytes e a este valor subtraimos o tamanho do cabeçalho IP (20 bytes) e do cabeçalho ICMP (8 bytes).

Parte II

Questão 1 - D.Afonso Henriques afirma ter problemas de comunicação com a sua mãe, D.Teresa. Este alega que o problema deverá estar no dispositivo de D.Teresa, uma vez que no dia anterior conseguiu enviar a sua declaração do IRS para o portal das finanças, e não tem qualquer problema em ver as suas séries favoritas disponíveis na rede de conteúdos.

Alínea a - Averigue, através do comando ping, que AfonsoHenriques tem efetivamente conectividade com o servidor Financas e com os servidores da CDN.

Ao efectuar o comando ping no PC AfonsoHenriques para o endereço IP do servidor Finanças, conseguimos verificar que dos pacotes transmitidos pelo host, todos foram recebidos no servidor, houve 0% de packet loss. O mesmo acontece quando se efetua ping para um servidor do polo CDN, neste caso fizemos para o IP do servidor Spotify como exemplo. Isto implica que existe conectividade entre AfonsoHenriques e Finanças e entre AfonsoHenriques os servidores do polo CDN

Alínea b - 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.

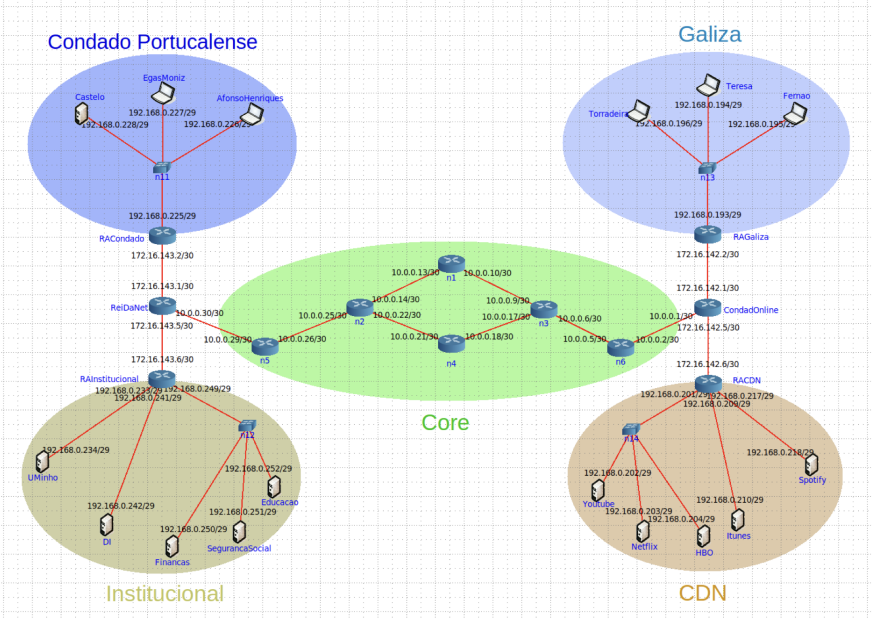

Fig. 23. Topologia da rede

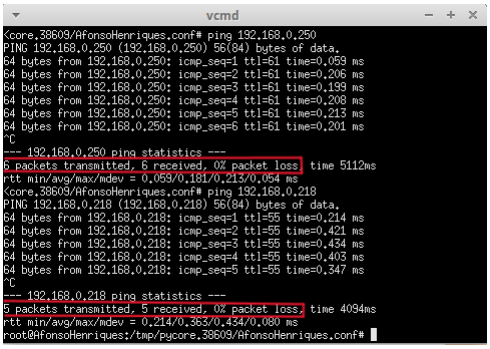

Fig. 24. Conectividade do AfonsoHenriques para o servidor Finanças e Spotify

No dispositivo AfonsoHenriques, a rota padrão (0.0.0.0) está definida para o gateway 192.168.0.225, o que indica que todo o tráfego destinado a redes fora da sub-rede local deve ser encaminhado para o gateway. Além disso, a entrada 192.168.0.224 indica que todo o tráfego destinado a essa sub-rede específica deve ser encaminhado diretamente para a rede local sem passar pelo gateway.

No dispositivo Teresa, a rota padrão (0.0.0.0) está definida para o gateway 192.168.0.193, o que indica que todo o tráfego destinado a redes fora da sub-rede local deve ser encaminhado para o gateway. Além disso, a entrada 192.168.0.192 indica que todo o tráfego destinado a essa sub-rede específica deve ser encaminhado diretamente para a rede local sem passar pelo gateway.

Estas rotas são importantes para garantir que os pacotes sejam encaminhados corretamente para seus destinos finais, permitindo que os dispositivos se comuniquem com outros dispositivos em redes diferentes ou na mesma sub-rede. Portanto, nas tabelas de encaminhamento, não há problemas aparentes com as rotas configuradas nestes dois dispositivos.

Alínea c - Utilize o Wireshark para investigar o comportamento dos routers do core da rede (n1 a n6) quando tenta estabelecer comunicação entre os hosts AfonsoHenriques e Teresa.

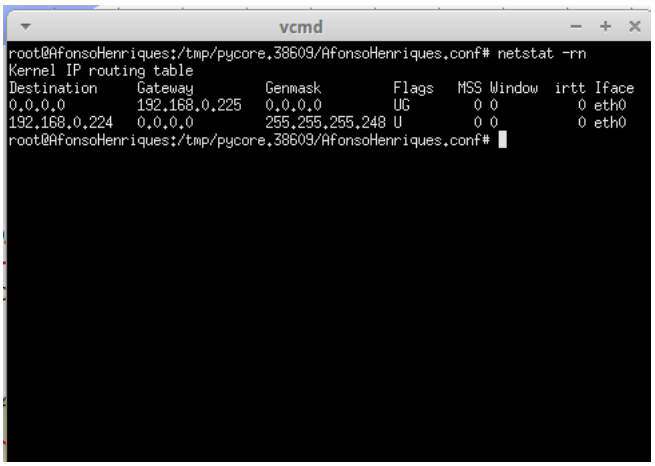

Fig. 25. Tabela de encaminhamento do AfonsoHenriques

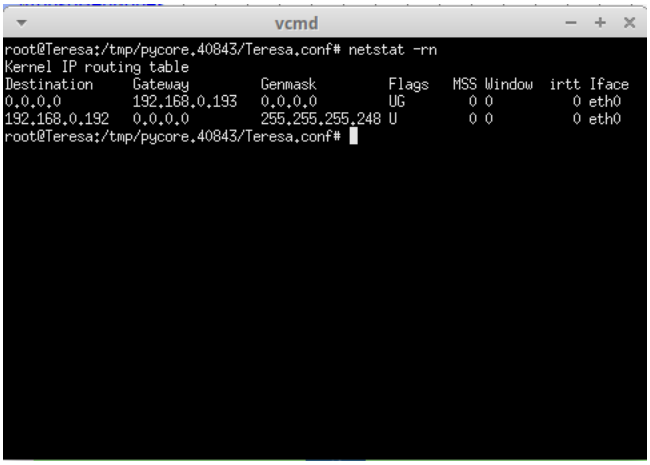

Fig. 26. Tabela de encaminhamento do Teresa

Indique que dispositivos não permitem o encaminhamento correto dos pacotes. Seguidamente, avalie e explique as causas 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.

Para avaliar a conexão entre os hosts AfonsoHenriques e Teresa realizamos o comando 'traceroute' no host AfonsoHenriques para o IP de Teresa (192.168.0.194). Podemos verificar na imagem abaixo que apenas existe roteamento do host AfonsoHenriques até ao router n5 com IP de acesso 10.0.0.29

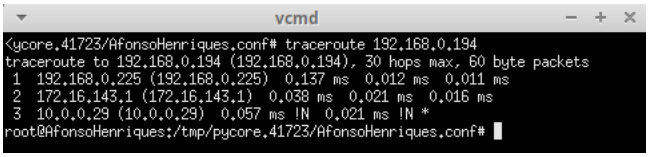

Fig. 27. Traceroute de AfonsoHenriques para o IP do Teresa

Ao efectuar o comando 'ip route' no router n5 pudemos verificar que este na sua tabela de encaminhamento não continha rota para o ip do host Teresa. Por isso neste router executamos o comando 'ip route add 192.168.0.192/29 via 10.0.0.25' que adicionou uma rota para o IP da rede local 192.168.0.192 através do IP de acesso do router n2 (10.0.0.25).

Fig. 28. ip route n5

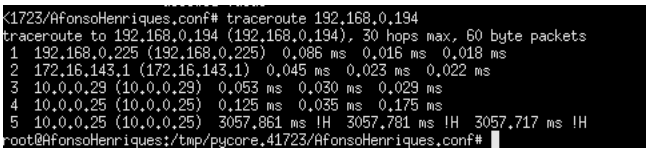

Fig. 29. Traceroute de AfonsoHenriques para o IP do Teresa após adicionada nova rota

Na tabela de roteamento do router n2, havia duas rotas para o mesmo destino. Embora a rota correta estivesse presente (192.168.0.192/29 via 10.0.0.13), havia outra rota que correspondia ao endereço IP que estávamos tentando alcançar. No entanto, essa rota encaminhava os pacotes para a interface com o endereço IP 10.0.0.25 do router n5, que por sua vez encaminhava os pacotes de volta para 10.0.0.13, criando um loop de envio de pacotes e tornando impossível estabelecer conexão com o host final. Essa situação pode ser verificada executando o comando traceroute no host AfonsoHenriques. Para resolver o problema, foi necessário executar o comando ip route del 192.168.0.194/31 via 10.0.0.25, removendo a rota incorreta da tabela de roteamento do router n2.

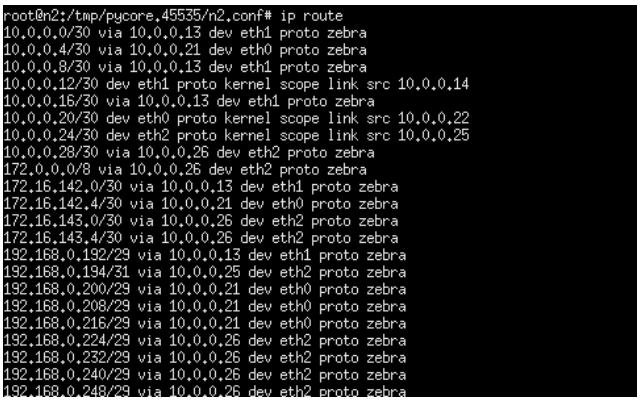

Fig. 30. ip route n3

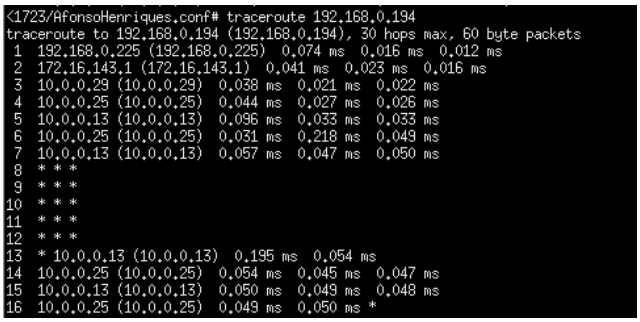

Fig. 31. Traceroute de AfonsoHenriques para o IP do Teresa com loop de envio de pacotes

No router n1, a rota para a rede 192.168.0.192/29 estava mal configurada e encaminhava os pacotes para o endereço IP 10.0.0.14, o que resultava em erro de conexão. Para corrigir a rota, foi necessário remover a rota incorreta usando o comando ip route del 192.168.0.192/29 via 10.0.0.14. Em seguida, adicionamos a nova rota correta usando o comando ip route add 192.168.0.192/29 via 10.0.0.9, garantindo que os pacotes sejam encaminhados corretamente para o destino desejado.

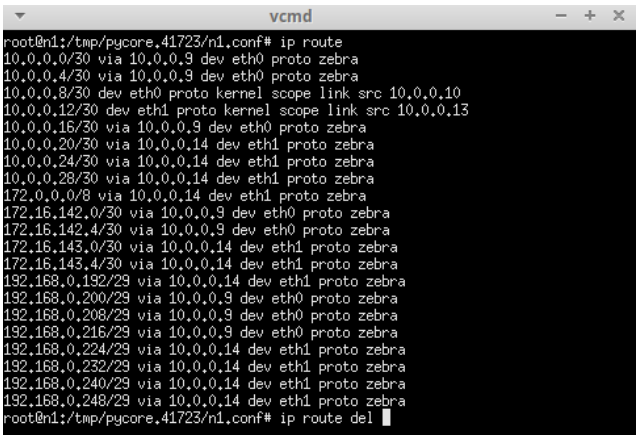

Fig. 32. ip route n1 antes das alterações

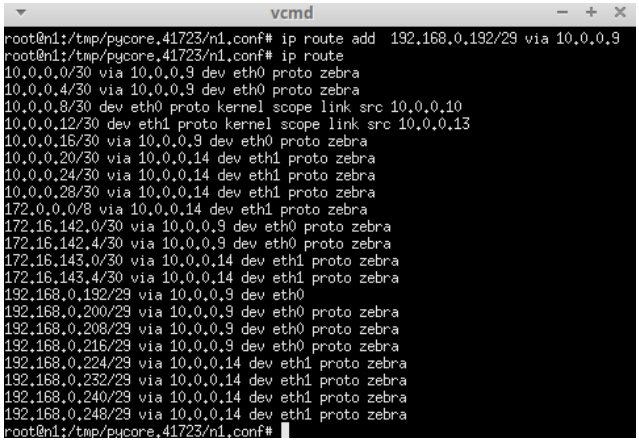

Fig. 33. ip route n1 depois das alterações

Podemos agora verificar através do comando traceroute 192.168.0.194 no Host AfonsoHenriques que existe encaminhamento nos routers do Core até ao ip de acesso ao router CondadOnline 10.0.0.1.

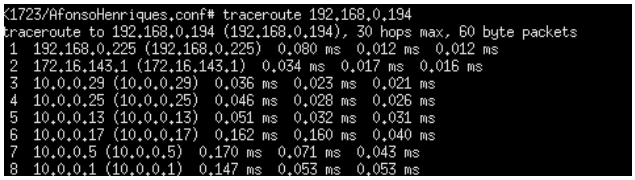

Fig. 34. Traceroute de AfonsoHenriques para o IP do Teresa depois das alterações finais

Alíena d - 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 - 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. II - 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.

Podemos verificar que não houve comunicação entre o host AfonsoHenriques e Teresa ao executar o comando 'ping 192.168.0.194', que resultou em 100% de perda de pacotes. Para resolver este problema, foi necessário adicionar uma rota usando o comando ip route add 192.168.0.224/29 via 172.16.142.1, criando uma rota entre os routers RAGaliza e CondadOnline, permitindo que o host AfonsoHenriques receba uma resposta do envio de pacotes para Teresa e, assim, estabelecendo uma comunicação bidirecional. Como resultado, ao executar o comando 'ping' novamente para Teresa, não houve perda de pacotes, indicando que os pacotes estão sendo encaminhados corretamente.

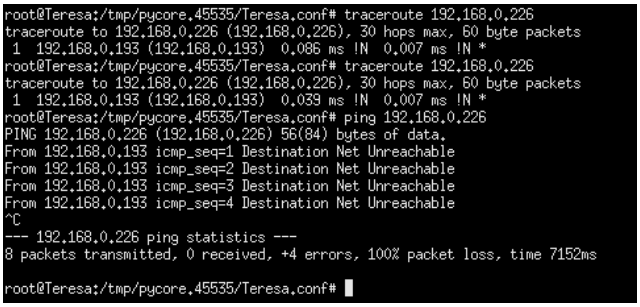

Fig. 35. Ping de AfonsoHenriques para o IP do Teresa que verifica 100% packet loss

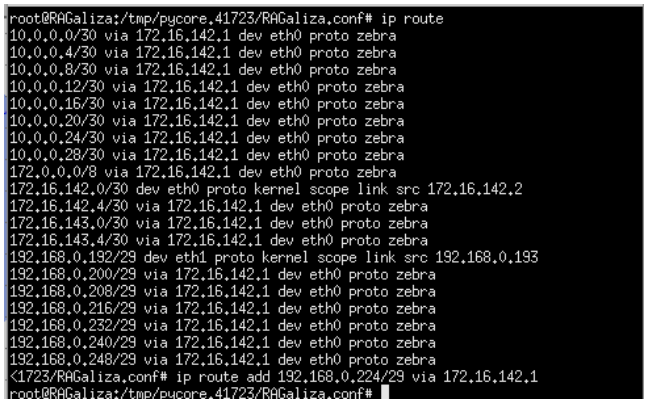

Fig. 36. ip route de RAGaliza

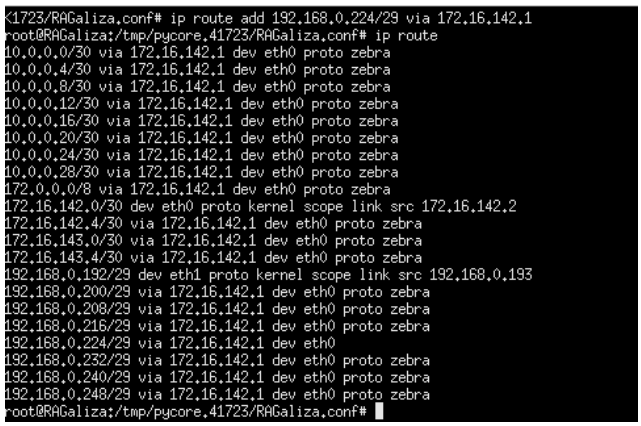

Fig. 37. ip route de RAGaliza depois de adicionar a nova rota

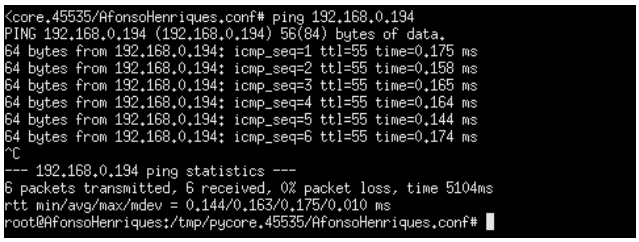

Fig. 38. Ping de AfonsoHenriques para o IP do Teresa que verifica 100% packet loss

Alíena e - Estando restabelecida a conectividade entre os dois hosts, obtenha a tabela de encaminhamento de n3 e foque-se na seguinte entrada:

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.

Existe correspondência entre esta rota e os pacotes enviados para o polo galiza como se pode verificar na tabela de encaminhamento anterior e ao executar o comando traceroute no router n4. Pois são enviados pacotes para n4 através do IP de acesso 10.0.0.18 e de seguida são encaminhados de volta para o router n3 e sucessivamente até ao host Teresa. Assim, existem diferentes rotas para chegar ao mesmo host mas o comprimento destas é diferente, a métrica mais comum usada pelos protocolos de roteamento é o número de saltos, ou seja, o número de routers intermediários que uma rota deve passar para alcançar o destino. Normalmente, rotas com menor número de saltos são preferidas, pois são consideradas mais rápidas e confiáveis e por isso o roteamento de pacotes neste caso tem menor número de saltos se não optar pela rota que com o IP de acesso ao router n4 10.0.0.8, mas sim um rota que encaminha tráfego diretamente para o router n6.

```
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.0.0        10.0.0.5        255.255.255.252 UG        0 0          0 eth2
10.0.0.4        0.0.0.0         255.255.255.252 U         0 0          0 eth2
10.0.0.8        0.0.0.0         255.255.255.252 U         0 0          0 eth0
10.0.0.12       10.0.0.10       255.255.255.252 UG        0 0          0 eth0
10.0.0.16       0.0.0.0         255.255.255.252 U         0 0          0 eth1
10.0.0.20       10.0.0.18       255.255.255.252 UG        0 0          0 eth1
10.0.0.24       10.0.0.18       255.255.255.252 UG        0 0          0 eth1
10.0.0.28       10.0.0.10       255.255.255.252 UG        0 0          0 eth0
172.0.0.0       10.0.0.10       255.0.0.0       UG        0 0          0 eth0
172.16.142.0    10.0.0.5        255.255.255.252 UG        0 0          0 eth2
172.16.142.4    10.0.0.5        255.255.255.252 UG        0 0          0 eth2
172.16.143.0    10.0.0.18       255.255.255.252 UG        0 0          0 eth1
172.16.143.4    10.0.0.10       255.255.255.252 UG        0 0          0 eth0
192.168.0.192   10.0.0.5        255.255.255.248 UG        0 0          0 eth2
192.168.0.192   10.0.0.18       255.255.255.240 UG        0 0          0 eth1
192.168.0.200   10.0.0.5        255.255.255.248 UG        0 0          0 eth2
192.168.0.208   10.0.0.5        255.255.255.248 UG        0 0          0 eth2
192.168.0.216   10.0.0.5        255.255.255.248 UG        0 0          0 eth2
192.168.0.224   10.0.0.18       255.255.255.248 UG        0 0          0 eth1
192.168.0.232   10.0.0.10       255.255.255.248 UG        0 0          0 eth0
192.168.0.240   10.0.0.10       255.255.255.248 UG        0 0          0 eth0
192.168.0.248   10.0.0.10       255.255.255.248 UG        0 0          0 eth0
root@n3:/tmp/pycore.45535/n3.conf#
```

Fig. 39. Tabela Encaminhamento n3

```
64 bytes from 192.168.0.194: icmp_seq=6 ttl=60 time=0.100 ms
64 bytes from 192.168.0.194: icmp_seq=7 ttl=60 time=0.102 ms
64 bytes from 192.168.0.194: icmp_seq=8 ttl=60 time=0.102 ms
64 bytes from 192.168.0.194: icmp_seq=9 ttl=60 time=0.116 ms
64 bytes from 192.168.0.194: icmp_seq=10 ttl=60 time=0.105 ms
64 bytes from 192.168.0.194: icmp_seq=11 ttl=60 time=0.111 ms
64 bytes from 192.168.0.194: icmp_seq=12 ttl=60 time=0.102 ms
64 bytes from 192.168.0.194: icmp_seq=13 ttl=60 time=0.094 ms
64 bytes from 192.168.0.194: icmp_seq=14 ttl=60 time=0.102 ms
64 bytes from 192.168.0.194: icmp_seq=15 ttl=60 time=0.099 ms
64 bytes from 192.168.0.194: icmp_seq=16 ttl=60 time=0.101 ms
64 bytes from 192.168.0.194: icmp_seq=17 ttl=60 time=0.104 ms
^C
--- 192.168.0.194 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16372ms
rtt min/avg/max/mdev = 0.081/0.104/0.122/0.008 ms
root@n4:/tmp/pycore.45535/n4.conf# traceroute 192.168.0.194
traceroute to 192.168.0.194 (192.168.0.194), 30 hops max, 60 byte packets
 1  10.0.0.17 (10.0.0.17)  0.045 ms  0.007 ms  0.006 ms
 2  10.0.0.5 (10.0.0.5)  0.024 ms  0.009 ms  0.008 ms
 3  10.0.0.1 (10.0.0.1)  0.023 ms  0.010 ms  0.011 ms
 4  172.16.142.2 (172.16.142.2)  0.027 ms  0.014 ms  0.013 ms
 5  192.168.0.194 (192.168.0.194)  0.028 ms  0.016 ms  0.017 ms
root@n4:/tmp/pycore.45535/n4.conf#
```

Fig. 40. traceroute n4

Alíena f - Os endereços utilizados pelos quatro polos são endereços públicos ou privados? E os utilizados no core da rede/ISPs? Justifique convenientemente.

Endereços IP públicos são aqueles que são roteáveis pela Internet e são exclusivos em todo o mundo. Esses endereços são atribuídos pelos Registradores Regionais da Internet (RIRs) e geralmente são fornecidos aos ISPs, empresas e outras organizações que se desejam conectar à Internet.

Já os endereços IP privados são aqueles que são usados em redes locais e não são roteáveis pela Internet. Esses endereços são usados para identificar dispositivos numa rede local, como numa empresa, por exemplo. Eles são reservados para uso privado e não são únicos em todo o mundo.

Com base nessa definição, podemos inferir que os endereços usados pelos quatro polos são endereços privados, pois são usados em redes locais e não precisam ser roteáveis pela Internet.

No entanto, os endereços usados no core da rede ou pelos ISPs podem ser endereços públicos, pois esses dispositivos precisam se comunicar com outras redes e dispositivos fora da rede local. Esses endereços são fornecidos pelos RIRs aos ISPs e outras organizações que fornecem acesso à Internet.

Alíena g - Os switches localizados em cada um dos polos têm um endereço IP atribuído? Porquê?

Os switches operam no nível 2 do modelo OSI e não usam endereços IP para encaminhamento de tráfego, mas sim endereços MAC.

Os switches usam a tabela CAM (Content Addressable Memory) para armazenar informações sobre os endereços MAC dos dispositivos conectados às suas portas. Quando um dispositivo envia um pacote para outro dispositivo na mesma rede local, o switch verifica a tabela CAM para encontrar o endereço MAC de destino e encaminha o pacote para a porta correspondente.

Portanto, não é necessário atribuir um endereço IP aos switches para permitir que eles encaminhem tráfego na rede local.

Questão 2 - Tendo feito as pazes com a mãe, D. Afonso Henriques vê-se com algum tempo livre e decide fazer remodelações no condado:

Alínea a - Não estando satisfeito com a decoração do Castelo, opta por eliminar a sua rota default. Adicione as rotas necessárias para que o Castelo continue a ter acesso a cada um dos três polos. Mostre que a conectividade é restabelecida, assim como a tabela de encaminhamento resultante. Explicite ainda a utilidade de uma rota default.

Eliminamos a rota default com o comando 'route del -net 0.0.0.0 netmask 0.0.0.0'. De seguida adicionamos uma rota do Castelo para o polo Institucional ao executar o comando 'ip route add 192.168.0.240/29 via 192.168.0.225'. Podemos confirmar que existe conectividade para o polo Institucional ao executar o comando ping para o IP de um dos hosts deste polo, neste caso escolhemos o host DI (192.168.0.242).

Fig. 41. Delete da rota default

Fig. 42. Adição de uma rota do polo Galiza para o polo Institucional

Fig. 43. Ping do host Castelo para o host DI

Adicionamos também uma rota do Castelo para o polo CDN ao executar o comando 'ip route add 192.168.0.208/29 via 192.168.0.225'. Podemos confirmar que existe conectividade para o polo CDN ao executar o comando ping para o IP de um dos hosts deste polo, neste caso escolhemos o host Itunes (192.168.0.210).

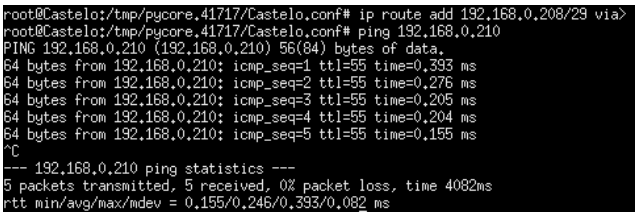

Fig. 44. Adição da rota para o polo CDN e Ping do host Itunes

Repetimos o processo para o polo Galiza mas devido a termos aberto a topologia de novo, as alterações no roteamento do exercicio 1 não foram gravadas e por isso o comando ping para um host do polo Galiza não vai ter o resultado esperado.

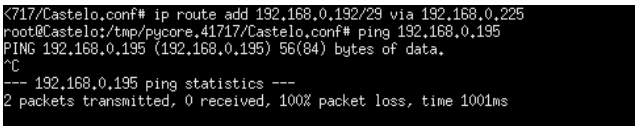

Fig. 45. Adição da rota para o polo Galiza e Ping do host Fernao

A rota default é uma rota de rede configurada num router que indica qual o caminho que deve ser tomado para qualquer tráfego que não corresponda a nenhuma outra rota na tabela de roteamento. Quando a rota padrão é removida, como no exemplo abaixo, o router não sabe para onde encaminhar os pacotes que não têm uma rota específica definida. Isto ilustra a importância da rota padrão para garantir que o tráfego de rede seja encaminhado corretamente para o destino desejado. Quando testamos a conectividade no router, por exemplo ReidaNet, não obtemos encaminhamento de pacotes pois não existe rota default.

Fig. 46. Ping do router ReidaNet

Alínea b - Por modo a garantir uma posição estrategicamente mais vantajosa e ter casa de férias para relaxar entre batalhas, ordena também a construção de um segundo Castelo, em Braga. Não tendo qualquer queixa do serviço prestado, recorre novamente aos serviços do ISP ReiDaNet para ter acesso à rede no segundo Castelo. O ISP atribuiu-lhe o endereço de rede IP 172.16.XX.128/26 em que XX corresponde ao seu número de grupo (PLXX). Defina um esquema de endereçamento que permita o estabelecimento de pelo menos 3 redes e que garanta que cada uma destas possa ter 10 ou mais hosts. Assuma que todos os endereços de sub-redes são utilizáveis.

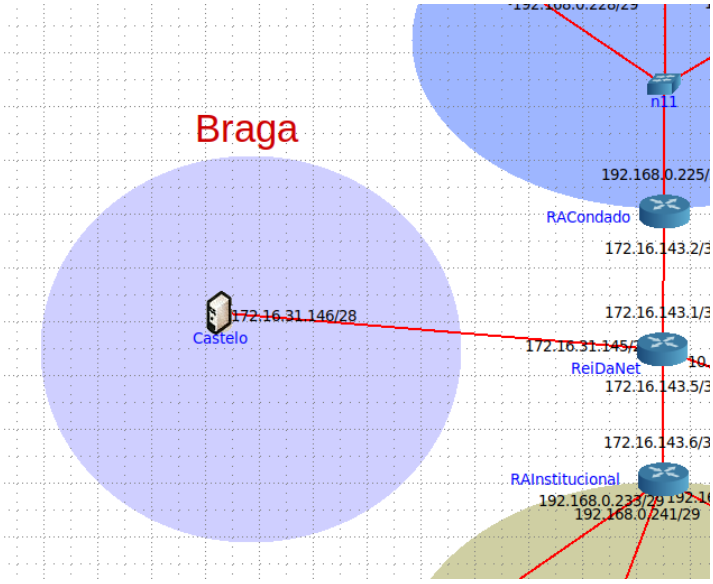

Fig. 47. Criação do host Castelo, em Braga

Ao definirmos um esquema de endereçamento com máscara /28 reservamos 2 bits para subredes e 4 bits para hosts, o que nos permite no máximo obter 4 subredes, sendo que que queremos estabelecer pelo menos 3. E os 4 bits para hosts permite 2^4, ou seja 16 hosts, mas os endereços XXXX0000 e XXXX1111 estão reservados, assim restam-nos 14 endereços para hosts, sendo que queremos estabelecer 10 ou mais hosts. Escolhemos os endereços 172.16.31.145/28 no router ReidaNet e 172.16.31.146/28 para o host Castelo do polo Braga, pertencentes á mesma subrede.

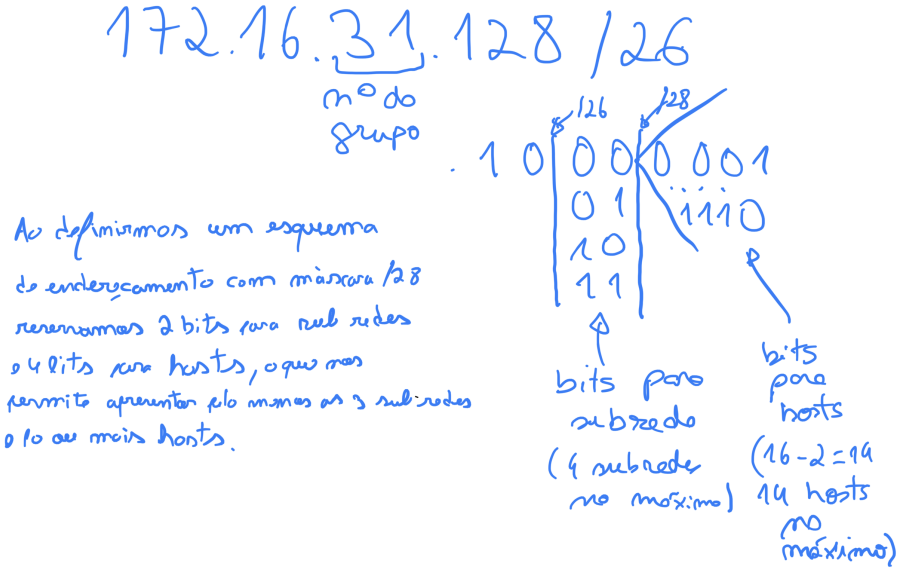

Fig. 48. Esquema de Subnetting

Para testarmos se existe conectividade do Castelo do polo Braga para os restantes polos, executamos o comando ping no Castelo para um host exemplo, neste caso o Spotify (192.168.0.218) e pudemos verificar na imagem abaixo que não houve perda de pacotes e por isso existe conectividade.

```
root@Castelo:/tmp/pycore.41717/Castelo.conf# ping 192.168.0.218
PING 192.168.0.218 (192.168.0.218) 56(84) bytes of data.
64 bytes from 192.168.0.218: icmp_seq=1 ttl=56 time=0.442 ms
64 bytes from 192.168.0.218: icmp_seq=2 ttl=56 time=0.264 ms
64 bytes from 192.168.0.218: icmp_seq=3 ttl=56 time=0.216 ms
64 bytes from 192.168.0.218: icmp_seq=4 ttl=56 time=0.215 ms
64 bytes from 192.168.0.218: icmp_seq=5 ttl=56 time=0.215 ms
64 bytes from 192.168.0.218: icmp_seq=6 ttl=56 time=0.203 ms
64 bytes from 192.168.0.218: icmp_seq=7 ttl=56 time=0.216 ms
^C
--- 192.168.0.218 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6143ms
rtt min/avg/max/mdev = 0.203/0.253/0.442/0.079 ms
root@Castelo:/tmp/pycore.41717/Castelo.conf#
```

Fig. 49. Ping do host Spotify

Alínea c - Ligue um novo host 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 da sub-rede escolhida). Verifique que tem conectividade com os diferentes polos. Existe algum host com o qual não seja possível comunicar? Porquê?

Criamos um novo host de nome Vassalo e IP de acesso 172.16.31.130 de uma diferente sub-rede do host Castelo. É possivel comunicar sem perdas de pacotes para os hosts dos polos CDN e Institucional. Para os hosts do polo o encaminhamento não está correto pelas razões faladas na alínea a e por isso não existe conectividade, mas em condições de um roteamento correto era esperado existir conectividade do host vassalo para esses hosts. Para os hosts do polo condado Portucalense existe comunicação para os hosts EgasMoniz e AfonsoHenriques mas não para o host Castelo.

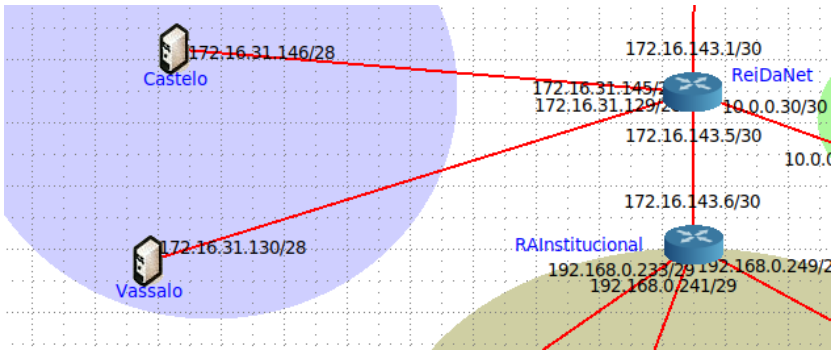

Fig. 50. Criação do novo host Vassalo

```
root@Vassalo:/tmp/pycore.34033/Vassalo.conf# ping 192.168.0.252
PING 192.168.0.252 (192.168.0.252) 56(84) bytes of data.
64 bytes from 192.168.0.252: icmp_seq=1 ttl=62 time=0.089 ms
64 bytes from 192.168.0.252: icmp_seq=2 ttl=62 time=0.065 ms
64 bytes from 192.168.0.252: icmp_seq=3 ttl=62 time=0.067 ms
64 bytes from 192.168.0.252: icmp_seq=4 ttl=62 time=0.066 ms
64 bytes from 192.168.0.252: icmp_seq=5 ttl=62 time=0.065 ms
^C
--- 192.168.0.252 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4091ms
rtt min/avg/max/mdev = 0.065/0.070/0.089/0.009 ms
root@Vassalo:/tmp/pycore.34033/Vassalo.conf#
```

Fig. 51. Ping para o host Educação - Polo Instuticional

```
root@Vassalo:/tmp/pycore.34033/Vassalo.conf# ping 192.168.0.218
PING 192.168.0.218 (192.168.0.218) 56(84) bytes of data.
64 bytes from 192.168.0.218: icmp_seq=1 ttl=56 time=0.229 ms
64 bytes from 192.168.0.218: icmp_seq=2 ttl=56 time=0.122 ms
64 bytes from 192.168.0.218: icmp_seq=3 ttl=56 time=0.113 ms
64 bytes from 192.168.0.218: icmp_seq=4 ttl=56 time=0.113 ms
64 bytes from 192.168.0.218: icmp_seq=5 ttl=56 time=0.209 ms
^C
--- 192.168.0.218 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4089ms
rtt min/avg/max/mdev = 0.113/0.157/0.229/0.050 ms
```

Fig. 52. Ping para o host spotify - Polo CDN

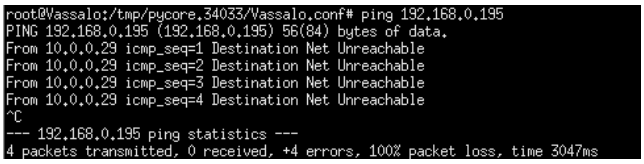

Fig. 53. Ping para o host Fernão - Polo Galiza

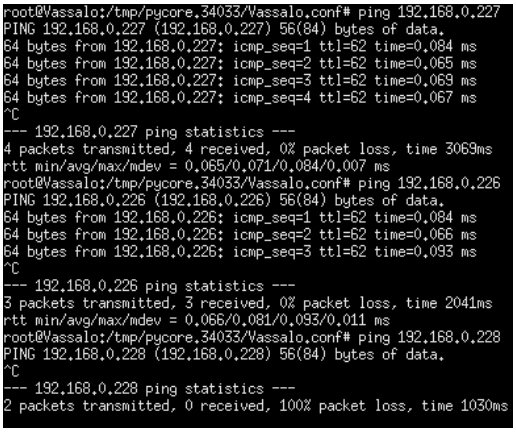

Fig. 54. Ping para o host Egas Moniz, AfonsoHenriques e Castelo - Polo CondadoPortucalense

Questão 3 - Ao planear um novo ataque, D. Afonso Henriques constata que o seu exército não só perde bastante tempo a decidir que direção tomar a cada salto como, por vezes, inclusivamente se perde.

Alínea a - 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.

Eliminamos as rotas referentes a Galiza (192.168.0.192) e CDN (192.168.0.200 e 192.168.0.208 e 192.168.0.216) através do comando 'ip route del 192.168.0.XXX via 10.0.0.1' e adicionamos a nova rota que encaminha pacotes para ambos os polos, com uma nova máscara de rede de /27 para podermos agrupar as redes dos dois polos numa rede maior com apenas uma rota de acesso (Supernetting), através do comando 'ip route add 192.168.0.192/27 via 10.0.0.1'. Através do ping conseguimos identificar que temos que conectividade do n6 para os hosts Spotify do polo CDN e do Fernao do polo Galiza embora so exista uma rota.

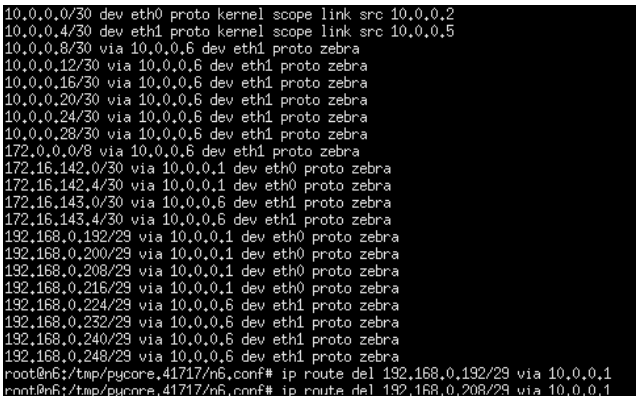

Fig. 55. Delete de rotas do polo Galiza e CDN

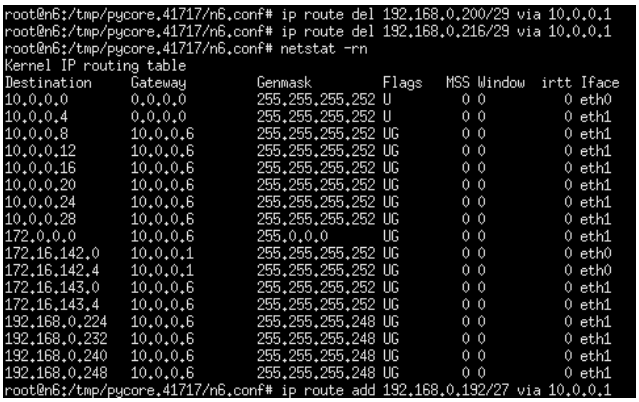

Fig. 56. Delete de rotas do polo CDN e Adição de rota de Supernetting

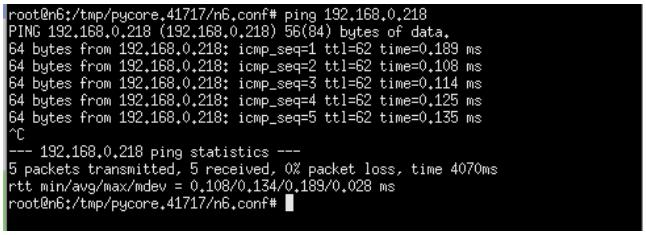

Fig. 57. Ping para o host Spotify - Polo CDN

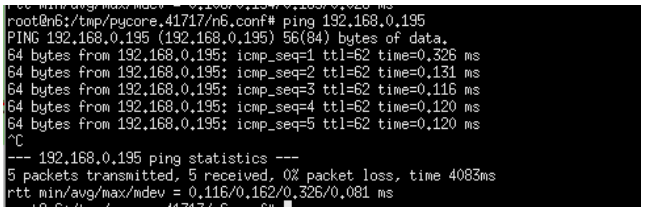

Fig. 58. Ping para o host Fernao - Polo Galiza

Alínea b - Repita o processo descrito na alínea anterior para CondadoPortucalense e Institucional, também no dispositivo n6.

O mesmo processo se aplica para CondadoPortucalense e Institucional, também no dispositivo n6, mas a gateway vai ser a 10.0.0.6. Foram apagadas as 4 rotas existentes, 1 para os hosts CondadoPortucalese e 3 outras para os hosts do Institucional. E foi adicionada a nova rota que encaminha pacotes para ambos os polos, com uma nova máscara de rede de /27 para podermos agrupar as redes dos dois polos numa rede maior com apenas uma rota de acesso (Supernetting), através do comando 'ip route add 192.168.0.224/27 via 10.0.0.6'. Verificamos a conectividade no router n6 para os hosts AfonsoHenriques e Educacao com o comando 'ping'.

Fig. 59. Delete das rotas do CondadoPortucalense e Institucional e adição da rota de Supernetting

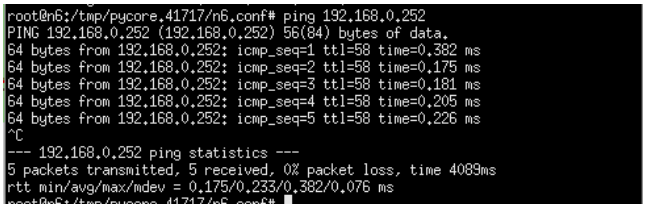

Fig. 60. Ping para o host Educaçao - Polo Institucional

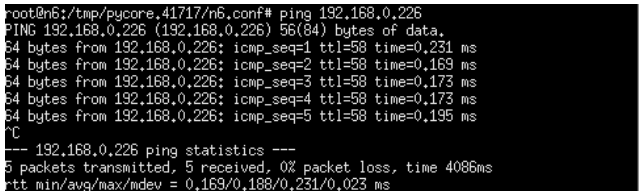

Fig. 61. Ping para o host AfonsoHenriques - Polo CondadoPortucalense

```
root@n6:/tmp/pycore.41717/n6.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.0.0        0.0.0.0         255.255.255.252 U         0 0          0 eth0
10.0.0.4        0.0.0.0         255.255.255.252 U         0 0          0 eth1
10.0.0.8        10.0.0.6        255.255.255.252 UG        0 0          0 eth1
10.0.0.12       10.0.0.6        255.255.255.252 UG        0 0          0 eth1
10.0.0.16       10.0.0.6        255.255.255.252 UG        0 0          0 eth1
10.0.0.20       10.0.0.6        255.255.255.252 UG        0 0          0 eth1
10.0.0.24       10.0.0.6        255.255.255.252 UG        0 0          0 eth1
10.0.0.28       10.0.0.6        255.255.255.252 UG        0 0          0 eth1
172.0.0.0       10.0.0.6        255.0.0.0       UG        0 0          0 eth1
172.16.142.0    10.0.0.1        255.255.255.252 UG        0 0          0 eth0
172.16.142.4    10.0.0.1        255.255.255.252 UG        0 0          0 eth0
172.16.143.0    10.0.0.6        255.255.255.252 UG        0 0          0 eth1
172.16.143.4    10.0.0.6        255.255.255.252 UG        0 0          0 eth1
192.168.0.192   10.0.0.1        255.255.255.224 UG        0 0          0 eth0
192.168.0.224   10.0.0.6        255.255.255.224 UG        0 0          0 eth1
```

Fig. 62. Tabela de encaminhamento do router n6

Alínea c - Comente os aspetos positivos e negativos do uso do Supernetting.

O supernetting tem várias vantagens e desvantagens na administração de redes. Entre as vantagens, destaca-se a redução do tamanho da tabela de roteamento, o que simplifica a administração e melhora a eficiência dos routers. Esta técnica também contribui para a redução do tráfego de atualização de roteamento, diminuindo a sobrecarga na rede e melhorando a convergência. Por fim, o supernetting promove o uso mais eficiente do espaço de endereçamento IP, evitando o desperdício e adia a necessidade de migração para o IPv6.

Por outro lado, o supernetting também apresenta desvantagens. Pode tornar a administração da rede mais complexa, exigindo um planeamento cuidadoso e uma compreensão avançada do endereçamento IP e das máscaras de sub-rede. Além disso, a agregação de rotas pode resultar em latência adicional em alguns casos, devido à possibilidade de caminhos subóptimos na rede. A dificuldade em isolar e diagnosticar problemas de conectividade numa rede específica também pode ser uma desvantagem, já que o supernetting combina várias redes numa única entrada de rota. Por fim, a técnica pode limitar a capacidade de implementar políticas de roteamento mais granulares, pois as decisões de roteamento são baseadas nas rotas agregadas.

Em resumo, o supernetting traz benefícios como a simplificação do gestão de roteamento e a melhoria do desempenho da rede, mas também pode introduzir complexidade adicional e limitar a flexibilidade do roteamento. É crucial avaliar cuidadosamente as necessidades de uma rede e os trade-offs associados antes de implementar o supernetting.

Conclusão -

Neste trabalho prático, conseguimos reforçar o que foi ensinado nas aulas teóricas, incluindo temas como datagramas IP, fragmentação, endereçamento, encaminhamento, supernetting e subnetting. Além disso, adquirimos habilidades no uso de ferramentas de gestão de redes, como o CORE e o Wireshark, bem como na utilização de comandos como traceroute e ping. Essas competências serão extremamente úteis não apenas no âmbito da unidade curricular e do programa de estudos, mas também na nossa rotina diária e carreira profissional.