Trabalho Prático 2

Alexandra Santos, Inês Ferreira e Joana Branco
Universidade do Minho, Departamento de Informática
email: {a94523, a97372, a96584}@alunos.uminho.pt

1ª parte

Exercício 1

Prepare uma topologia CORE para verificar o comportamento do traceroute. Na topologia deve existir: um host (pc) cliente designado Bela cujo router de acesso é R2; o router R2 está simultaneamente ligado a dois routers R3 e R4; estes estão conectados a um router R5, que por sua vez, se liga a um host (servidor) designado Monstro. Ajuste o nome dos equipamentos atribuídos por defeito para o enunciado. Nas ligações (links) da rede de core estabeleça um tempo de propagação de 10ms. Após ativar a topologia, note que pode não existir conectividade IP imediata entre a Bela e o Monstro até que o anúncio de rotas entre routers estabilize.

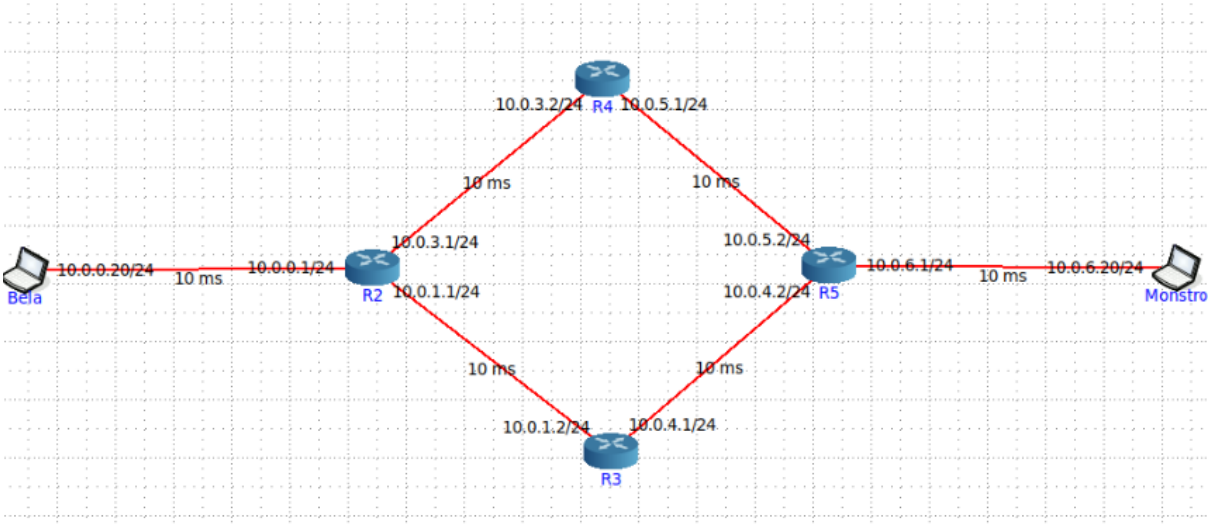


Figura 1: Core de acordo com a topologia

A. Ative o wireshark ou o tcpdump no host Bela. Numa shell de Bela execute o comando traceroute -I para o endereço IP do Monstro.

```
root@Bela:/tmp/pycore.38225/Bela.conf# traceroute -I 10.0.6.20
traceroute to 10.0.6.20 (10.0.6.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  21.358 ms  21.307 ms  21.301 ms
 2  10.0.1.2 (10.0.1.2)  42.144 ms  42.141 ms  42.138 ms
 3  10.0.4.2 (10.0.4.2)  63.579 ms  63.575 ms  63.572 ms
 4  10.0.6.20 (10.0.6.20)  85.110 ms  85.108 ms  85.104 ms
root@Bela:/tmp/pycore.38225/Bela.conf#
```

Figura 2: Terminal com o comando traceroute -I 10.0.6.20

B. Registe e analise o tráfego ICMP enviado pelo sistema Bela e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.

R: Através do tráfego ICMP, o host Bela envia três pacotes ao servidor Monstro. O TTL é apresentado de forma crescente e inicia-se com o valor de 1.

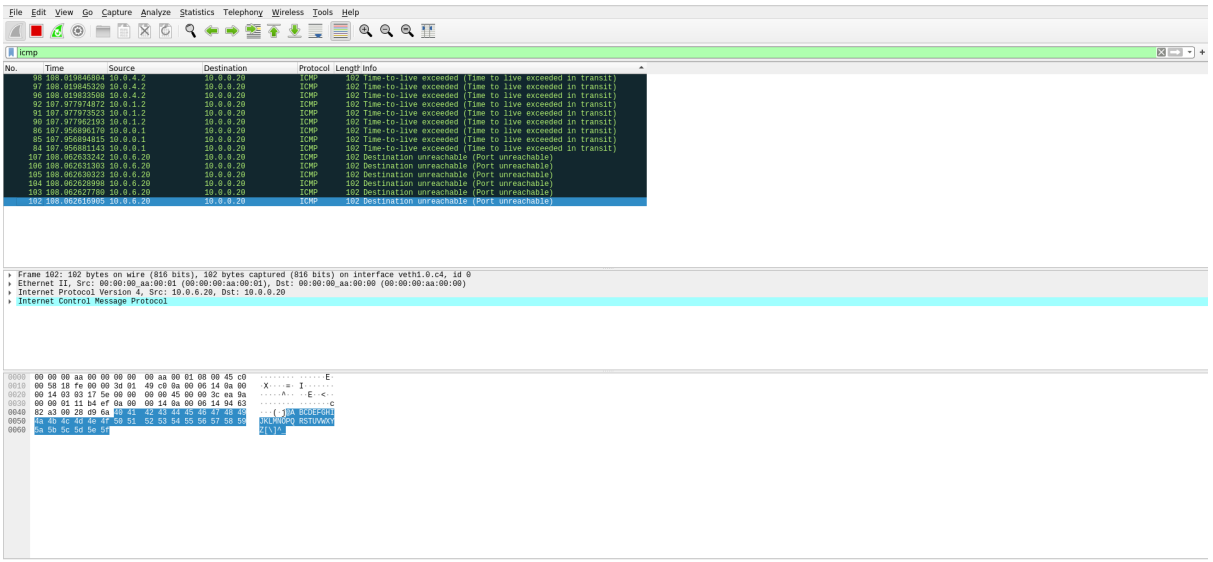


Figura 3: Tráfego ICMP no WireShark

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

R: De maneira a alcançar o servidor Monstro, é necessário um valor mínimo de TTL correspondente a 4. Só é a partir deste mesmo valor que o servidor Monstro recebe uma resposta.

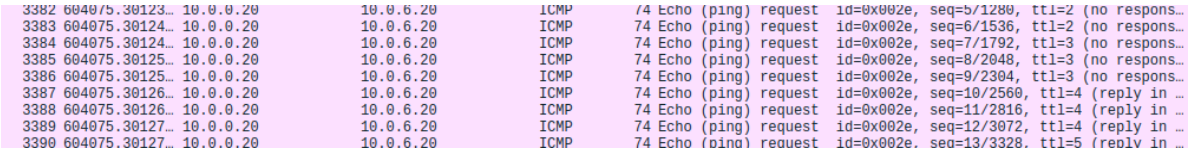


Figura 4: Menor valor de TTL

D. Calcule o valor médio do tempo de ida-e-volta (RTT - Round-Trip Time) obtido no acesso ao servidor. Para melhorar a média, poderá alterar o número pacotes de prova com a opção -q.

R: Ao aceder ao servidor, o valor RTT obtido foi de 93.216 ms. Comparando com os valores máximo e mínimo resultantes de 10 pacotes, podemos concluir que o resultado tende para o de menor valor.

```
root@Bela:/tmp/pycore.41931/Bela.conf# traceroute -I 10.0.6.20 -q 10
traceroute to 10.0.6.20 (10.0.6.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  40.750 ms  40.720 ms  40.718 ms  40.718 ms  40.717 ms
40.716 ms * * * *
 2  10.0.1.2 (10.0.1.2)  81.909 ms  81.910 ms  81.909 ms  81.907 ms  81.908 ms
81.907 ms * * * *
 3  10.0.4.2 (10.0.4.2)  83.062 ms  83.055 ms  61.635 ms  61.580 ms  61.570 ms
61.562 ms * * * *
 4  10.0.6.20 (10.0.6.20)  104.082 ms  104.070 ms  99.737 ms  99.686 ms  87.891
ms  87.820 ms  87.807 ms  87.798 ms  86.653 ms  86.619 ms
root@Bela:/tmp/pycore.41931/Bela.conf#
```

Figura 5: Terminal com o comando -q

E. 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?

R: Não, o valor médio de atraso pode ser diferente em cada um dos sentidos. Ou seja, o tempo de ida do pacote pode ser maior que o tempo de volta, ou vice-versa logo, não pode ser calculado com precisão dividindo o RTT por dois.

Exercício 2

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

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

R: O endereço IP da interface ativa é 172.26.254.254.

B. Qual é o valor do campo protocolo? O que permite identificar?

R: O valor do campo protocolo corresponde ao ICMP e permite identificar a mensagem capturada.

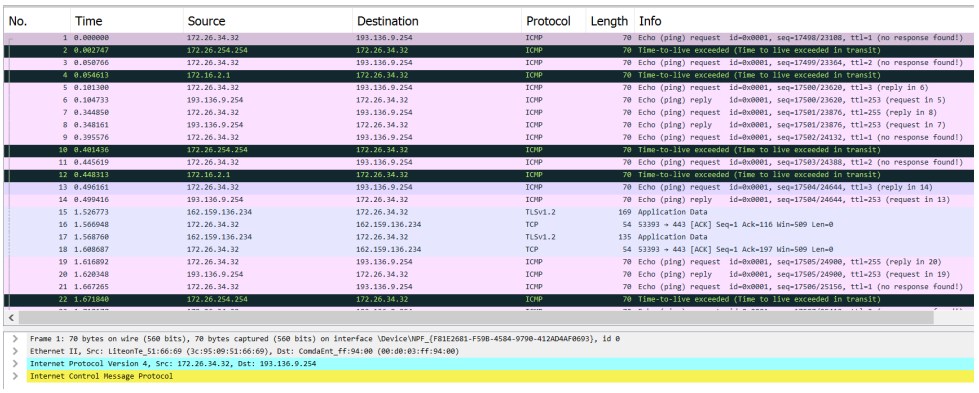


Figura 6: Primeira mensagem ICMP capturada

C. Quantos bytes tem o cabeçalho IPv4? Quantos bytes tem o campo de dados (payload) do datagrama? Como se calcula o tamanho do payload?

R: O cabeçalho IPv4 tem 20 *bytes* e são 56 *bytes* para o campo de dados, de comprimento total. Logo, o tamanho do payload é de 36 *bytes* (56-20=36).

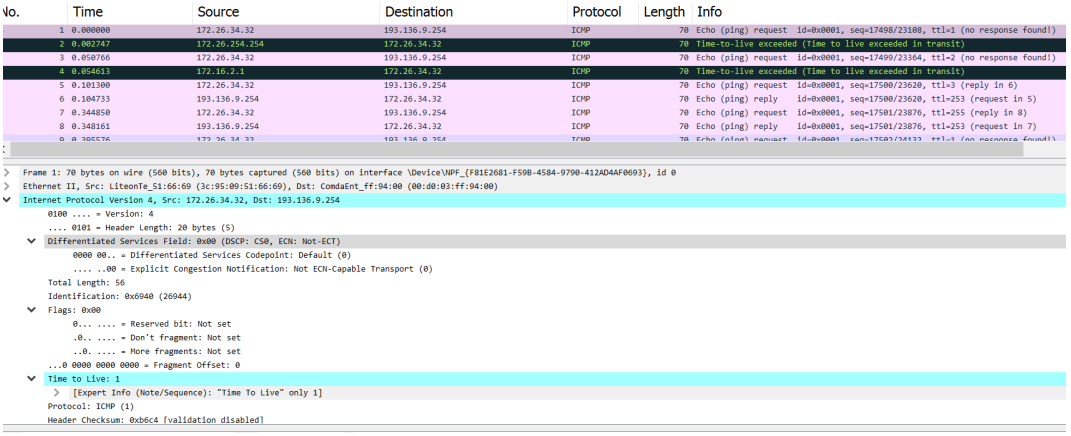


Figura 7: Detalhes da primeira mensagem

D. O datagrama IP foi fragmentado? Justifique.

R: Não, o datagrama IP não foi fragmentado, tal como se pode verificar pelo campo das *Flags* com valor 0. Para além disto o campo *More Fragments* e *Fragment Offset* encontram-se ambos com o valor 0 e, este último é o que permite assumir se este é o primeiro fragmento ou se não foi fragmentado de todo.

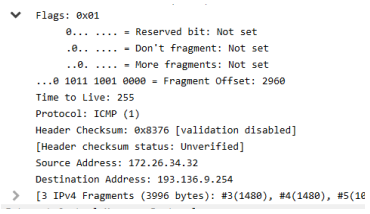
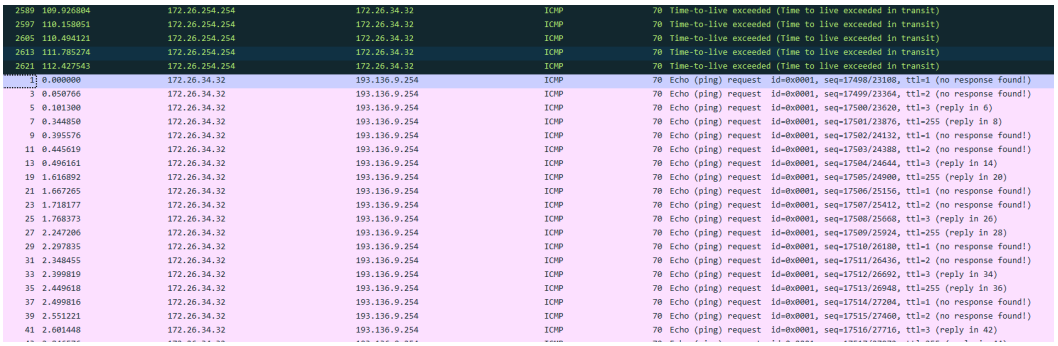


Figura 8: Detalhes de fragmentação da primeira mensagem

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.

R: O valor de TTL, de *Identification* e o campo de *Checksum* vão se alterando conforme se selecionam mensagens diferentes.



2589	109.926804	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2597	110.158051	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2605	110.494121	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2613	111.785274	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2621	112.427543	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
1	0.000000	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17498/23108, ttl=1 (no response found!)
3	0.050766	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17499/23364, ttl=2 (no response found!)
5	0.101300	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17500/23620, ttl=3 (reply in 6)
7	0.344850	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17501/23876, ttl=255 (reply in 8)
9	0.395576	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17502/24132, ttl=1 (no response found!)
11	0.445619	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17503/24388, ttl=2 (no response found!)
13	0.496161	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17504/24644, ttl=3 (reply in 14)
19	1.616892	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17505/24900, ttl=255 (reply in 20)
21	1.667265	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17506/25156, ttl=1 (no response found!)
23	1.718177	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17507/25412, ttl=2 (no response found!)
25	1.768373	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17508/25668, ttl=3 (reply in 26)
27	2.247206	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17509/25924, ttl=255 (reply in 28)
29	2.297835	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17510/26180, ttl=1 (no response found!)
31	2.348455	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17511/26436, ttl=2 (no response found!)
33	2.399819	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17512/26692, ttl=3 (reply in 34)
35	2.449618	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17513/26948, ttl=255 (reply in 36)
37	2.499816	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17514/27204, ttl=1 (no response found!)
39	2.551221	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17515/27460, ttl=2 (no response found!)
41	2.601448	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17516/27716, ttl=3 (reply in 42)

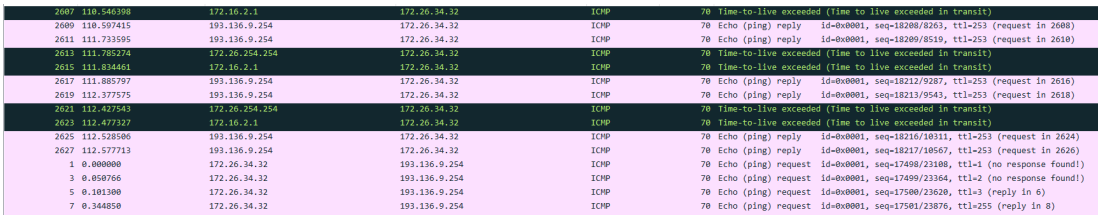
Figura 9: Pacotes capturados ordenados pela *source*

F. Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?

R: Os valores do campo *Identification* são sempre somados mais 1 de cada vez que é capturada uma mensagem e o TTL também vai aumentando de 1 em 1 a cada 3 mensagens.

G. Ordene o tráfego capturado por endereço destino e encontre a série de respostas ICMP TTL exceeded enviadas ao seu computador. Qual é o valor do campo TTL? Esse valor permanece constante para todas as mensagens de resposta ICMP TTL exceeded enviados ao seu host? Porquê?

R: Segundo a imagem abaixo, os valores do campo TTL vão se alternando entre 253 e 255.



2607	110.546398	172.16.2.1	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2609	110.597415	193.136.9.254	172.26.34.32	ICMP	70 Echo (ping) reply id=0x0001, seq=18208/8263, ttl=253 (request in 2608)
2611	111.733595	193.136.9.254	172.26.34.32	ICMP	70 Echo (ping) reply id=0x0001, seq=18209/8519, ttl=253 (request in 2610)
2613	111.785274	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2615	111.834461	172.16.2.1	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2617	111.885797	193.136.9.254	172.26.34.32	ICMP	70 Echo (ping) reply id=0x0001, seq=18212/9287, ttl=253 (request in 2616)
2619	112.377575	193.136.9.254	172.26.34.32	ICMP	70 Echo (ping) reply id=0x0001, seq=18213/9543, ttl=253 (request in 2618)
2621	112.427543	172.26.254.254	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2623	112.477327	172.16.2.1	172.26.34.32	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
2625	112.528506	193.136.9.254	172.26.34.32	ICMP	70 Echo (ping) reply id=0x0001, seq=18216/10311, ttl=253 (request in 2624)
2627	112.577713	193.136.9.254	172.26.34.32	ICMP	70 Echo (ping) reply id=0x0001, seq=18217/10567, ttl=253 (request in 2626)
1	0.000000	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17498/23108, ttl=1 (no response found!)
3	0.050766	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17499/23364, ttl=2 (no response found!)
5	0.101300	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17500/23620, ttl=3 (reply in 6)
7	0.344850	172.26.34.32	193.136.9.254	ICMP	70 Echo (ping) request id=0x0001, seq=17501/23876, ttl=255 (reply in 8)

Figura 10: Pacotes capturados ordenados pela *destination*

Exercício 3

Pretende-se agora analisar a fragmentação de pacotes IP. Reponha a ordem do tráfego capturado usando a coluna do tempo de captura. Observe o tráfego depois do tamanho do pacote ter sido definido para (4000 + X) bytes.

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

R: Houve necessidade de fragmentar o pacote inicial porque este excede o limite de 1500 *bytes* de transmissão de uma mensagem e, neste caso, redefiniu-se o pacote para 4016 *bytes*.

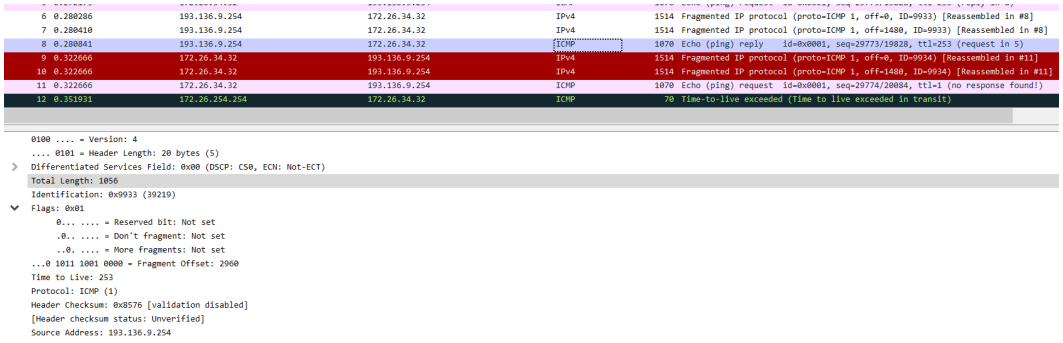


Figura 11: Detalhes da fragmentação do pacote

B. 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?

R: Pode-se verificar que o datagrama IP foi fragmentado pelo campo das *Flags* com valor 1. Trata-se do primeiro fragmento pois *Fragment Offset* tem valor 0 e como o campo *More Fragments* encontra-se a 1 é porque existem mais fragmentos para além deste. O tamanho deste datagrama IP é de 1500 *bytes*.

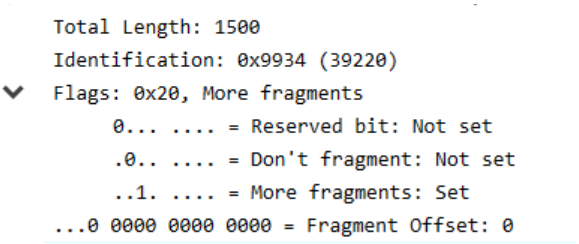


Figura 12: Detalhes do primeiro fragmento

C. Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1º fragmento? Há mais fragmentos? O que nos permite afirmar isso?

R: Como este fragmento tem o mesmo *Identification* do anterior, confirma-se que é o segundo fragmento do datagrama IP original. O campo *Fragment Offset* indica que não se trata do primeiro fragmento, é diferente de 0. O campo *More Fragments* é 1 logo, existem mais fragmentos.

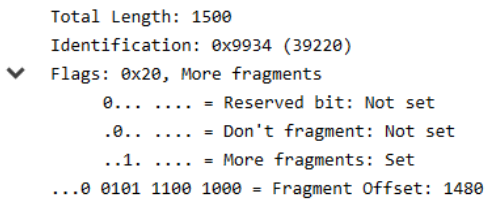


Figura 13: Detalhes do segundo fragmento

D. Quantos fragmentos foram criados a partir do datagrama original?

R: A partir do datagrama original foram criados 3 fragmentos.

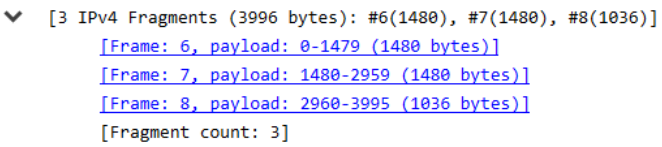


Figura 14: Fragmentos criados

E. 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.

R: Os campos que se alteram entre os diferentes fragmentos são o *Fragment Offset*, *Total Length* e, também, a *Header Checksum*. Como todos os fragmentos provenientes do datagrama original tem o mesmo *Identification* podemos reconstruir o mesmo. Para ser possível saber a ordem dos fragmentos basta ter em atenção ao campo *Fragment Offset* e ao *Total Length* de modo a alocar o espaço corretamente.

F. Verifique o processo de fragmentação através de um processo de cálculo.

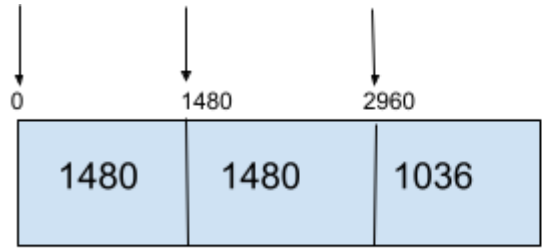
R:

Length = 4016 *bytes*
4016-20=3996 *bytes* (20 bytes de cabeçalho)
MTU= 1500 *bytes*
Identification= 0x9934 =39220
Fragmentos = 1500-20=1480 bytes dos dados

Fragmento 1
campo de dados = 3996 *bytes*
More Fragments = 1
Fragment offset = 0
3996-1480= 2516 *bytes*

Fragmento 2
campo de dados= 2516 *bytes*
2516-1480= 1036 *bytes*
More Fragments = 1
Fragment offset=1480 *bytes*

Fragmento 3
campo de dados= 1036 *bytes*
More Fragments=0
Fragment offset = 2960
1 pacote=4016 bytes
3 pacotes= 4016 + 3*20 = 4076



G. Escreva uma expressão lógica que permita detetar o último fragmento correspondente ao datagrama original.

R:
MoreFragments==0 &&(*offset* + Tamanho Fragmento>*length*) && *identification*==39220

De forma a detetar o último fragmento temos que ter em atenção ao campo *Fragment Offset*, este valor deve ser 0. O campo *Identification* também deve ser o mesmo em todos os fragmentos, inclusive no último.

2ª parte

Exercício 1

Atenda aos endereços IP atribuídos automaticamente pelo CORE aos diversos equipamentos da topologia.

A. Indique que endereços IP e máscaras de rede foram atribuídos pelo CORE a cada equipamento. Para simplificar, pode incluir uma imagem que ilustra de forma clara a topologia definida e o endereçamento usado.

R: Os endereços IP são referidos na imagem seguinte e as máscaras de redes tem 24 bits.

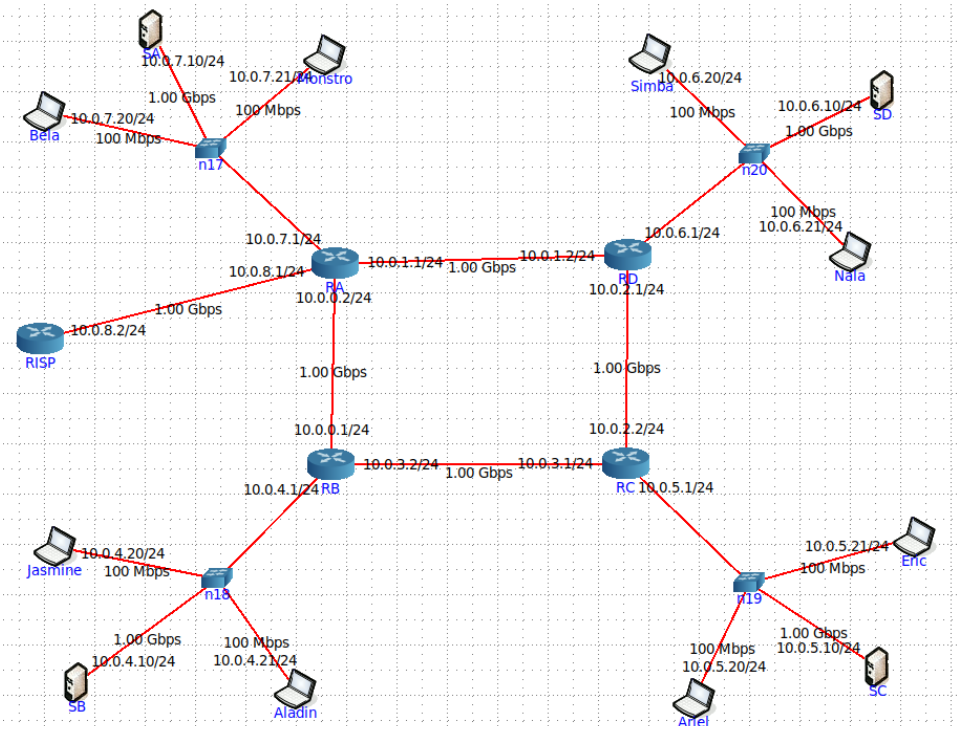


Figura 15: Core de acordo com a topologia

B. Tratam-se de endereços públicos ou privados? Porquê?

R: Estão no range dos endereços privados, logo são endereços privados.

C. Por que razão não é atribuído um endereço IP aos switches?

R: Os *switches* são dispositivos que se conectam com todos os elementos de uma rede, operando como ligação de dados, por tanto não precisam de adquirir um endereço IP pois a rede que eles propagam já contém um endereço.

D. Usando o comando ping certifique-se que existe conectividade IP interna a cada departamento (e.g. entre um laptop e o servidor respectivo).

R: Depois de efetuarmos o comando ping para um dos computadores dentro do mesmo departamento, confirmou-se que existe conectividade entre os dois, tal como podemos observar na figura 16.

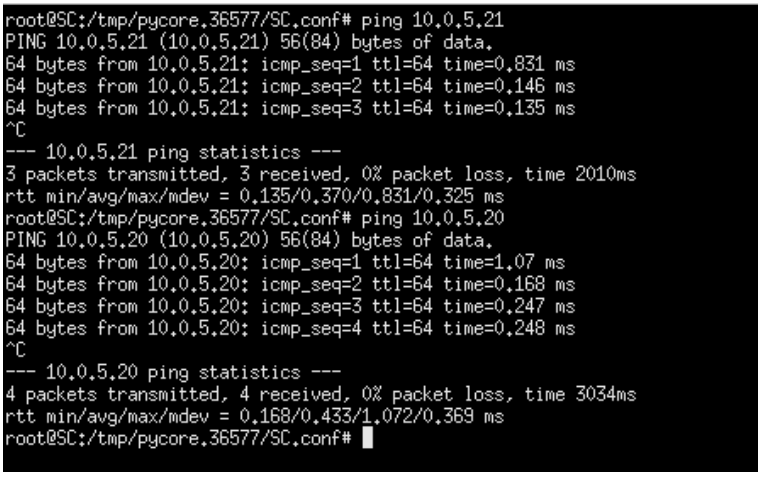


Figura 16: Comando ping na shell do servidor do departamento C, nele mesmo

E. Execute o número mínimo de comandos ping que lhe permite verificar a existência de conectividade IP entre departamentos.

R: Depois de efetuarmos o comando ping para um dos computadores de cada um dos departamentos, confirmou-se que existe conectividade entre os dois, tal como podemos observar na figura abaixo.

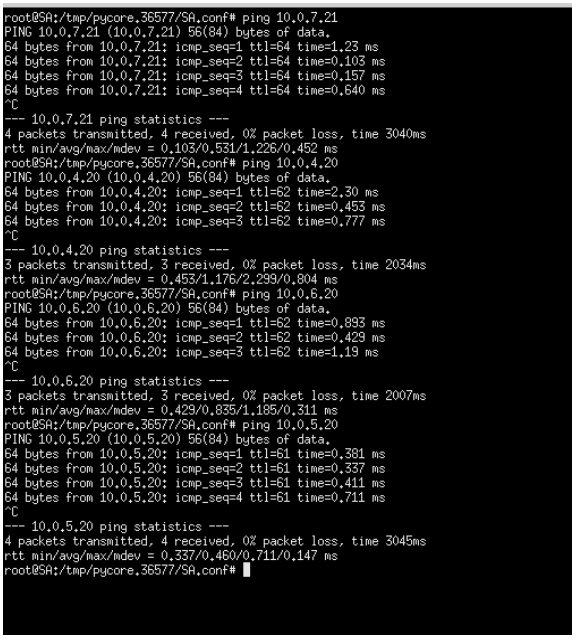


Figura 17: Comando ping na shell do servidor do departamento A, entre outros

F. Verifique se existe conectividade IP do portátil Bela para o router de acesso RISP.

R: Através do comando ping é possível verificar se existe conectividade entre o portátil Bela e o router de acesso RISP. Tal como se pode visualizar na imagem, foram enviados 5 pacotes e recebidos do mesmo valor, ou seja, há conectividade entre eles.

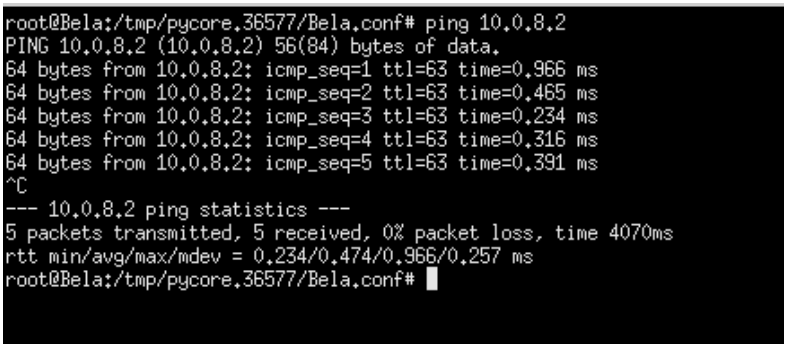


Figura 18: Comando ping entre o portátil Bela e router RISP

Exercício 2

Para o router RA e o portátil Bela:

A. Execute o comando netstat –rn por forma a poder consultar a tabela de encaminhamento unicast (IPv4). Inclua no seu relatório as tabelas de encaminhamento obtidas; interprete as várias entradas de cada tabela. Se necessário, consulte o manual respectivo (man netstat).

R: A primeira coluna, *Destination*, da tabela de encaminhamento refere-se aos endereços IP destino, especificando o destino da conexão .Na segunda coluna podemos ver o *Gateway* utilizado pela conexão. A coluna *Genmask* mostra a máscara de rede utilizada. A coluna *Flags* mostra as *flags* da conexão. A coluna MSS representa o *Maximum Segment Size* desta rota. *Window* especifica o tamanho da janela para conexões TCP nesta *route*. Por fim, a última coluna diz-nos qual é a interface que está a ser utilizada no encaminhamento desta conexão.

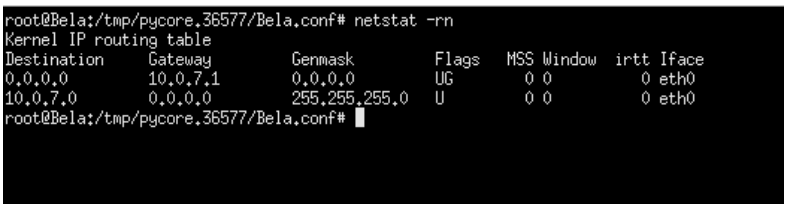


Figura 19: Tabela de encaminhamento do portátil Bela

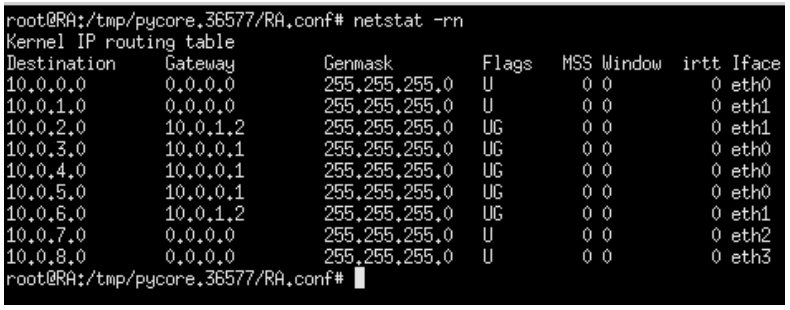


Figura 20: Tabela de encaminhamento do router A

B. Diga, justificando, se está a ser usado encaminhamento estático ou dinâmico (sugestão: analise que processos estão a correr em cada sistema, por exemplo, ps -ax ou equivalente).

R: De acordo com a figura abaixo, podemos observar que o protocolo OSPF está ativo, logo está a ser utilizado um encaminhamento dinâmico.

Figura 21: Comando ps-ax do portátil Bela

Figura 22: Comando ps-ax do router A

C. Admita que, por questões administrativas, a rota por defeito (0.0.0.0 ou default) deve ser retirada definitivamente da tabela de encaminhamento do servidor SA. Use o comando route delete para o efeito. Que implicações tem esta medida para os utilizadores da LEI-RC que acedem ao servidor. Justifique.

R: Se a rota por defeito for retirada definitivamente, o servidor de um certo departamento apenas consegue uma conexão com outros da mesma sub-rede. Tal como podemos observar na figura abaixo, não existe comunicação entre o servidor do departamento A e um host de uma qualquer outra sub-rede.

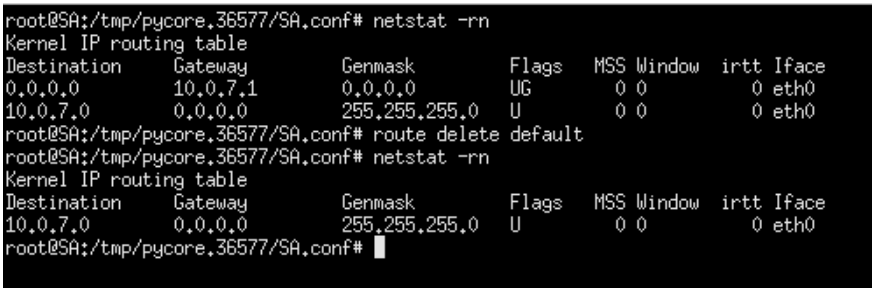


Figura 23: Tabela de encaminhamento do servidor A sem a rota por defeito

D. Não volte a repor a rota por defeito. Adicione todas as rotas estáticas necessárias para restaurar a conectividade para o servidor SA, por forma a contornar a restrição imposta na alínea c). Utilize para o efeito o comando route add e registe os comandos que usou.
R:

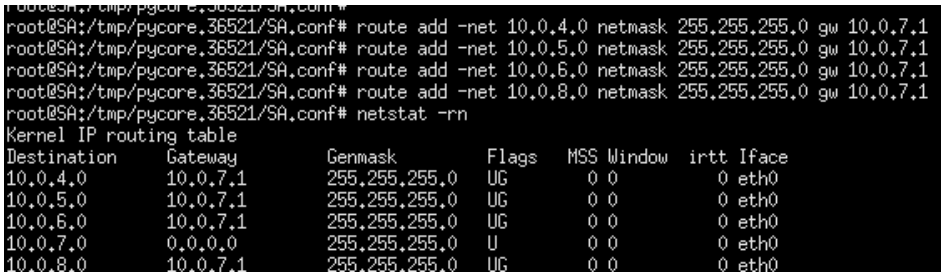


Figura 24: Adição de rotas necessárias

E. Teste a nova política de encaminhamento garantindo que o servidor está novamente acessível, utilizando para o efeito o comando ping. Registe a nova tabela de encaminhamento do servidor.

R:

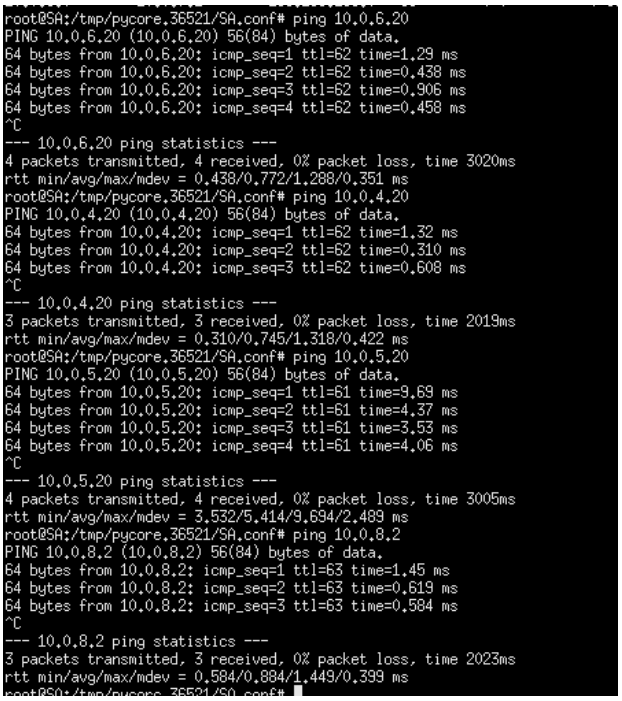


Figura 25: Comando ping na shell do servidor do departamento A, entre outros

Exercício 3

Por forma a minimizar a falta de endereços IPv4 é comum a utilização de sub-redes. Além disso, a definição de sub-redes permite uma melhor organização do espaço de endereçamento das redes em questão.

Para definir endereços de sub-rede é necessário usar a parte prevista para endereçamento de host, não sendo possível alterar o endereço de rede original. Recorda-se que o subnetting, ao recorrer ao espaço de endereçamento para host, implica que possam ser endereçados menos hosts.

Considere a topologia definida anteriormente. Assuma que o endereçamento entre os routers (rede de backbone) se mantém inalterado, contudo, o endereçamento em cada departamento deve ser redefinido.

1. Considere que dispõe apenas do endereço de rede IP 192.168.XXX.128/25, em que XXX é o decimal correspondente ao seu número de grupo (PLXX). Defina um novo esquema de endereçamento para as redes dos departamentos (mantendo as redes de acesso externo e backbone inalteradas), sabendo que o número de departamentos pode vir a aumentar no curto prazo. Atribua endereços às interfaces dos vários sistemas envolvidos. Assuma que todos os endereços de sub-redes são usáveis. Justifique as opções tomadas no planejamento.

   R: Usamos 4 sub-redes logo vamos precisar de 2 bits para a representar (em binário). Ficamos então com uma nova máscara de 29. Depois disso mudamos o endereço IP em todos os departamentos usando a nova máscara. Na figura abaixo vemos a nova topologia com 4 sub-redes.

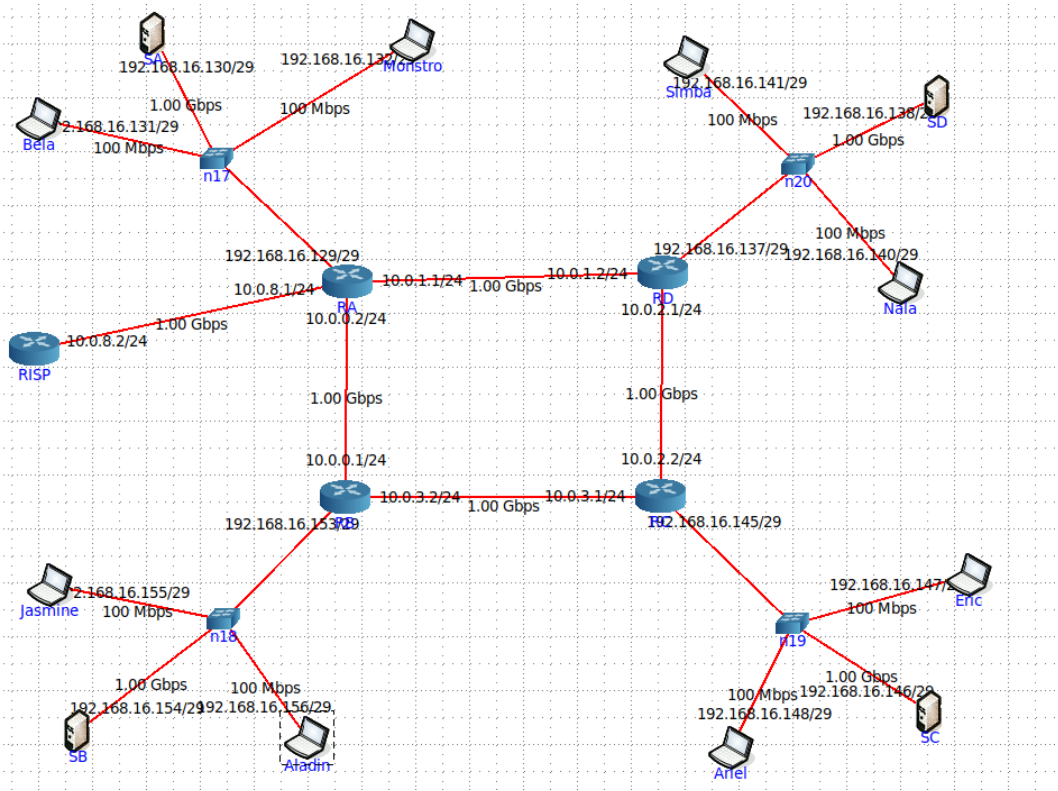


Figura 26: Core de acordo com as novas alterações na topologia

2. Qual a máscara de rede que usou (em formato decimal)? Quantos hosts IP podem interligar em cada departamento? Quantos prefixos de sub-rede ficam disponíveis para uso futuro? Justifique.

   R: Foi utilizada máscara de rede /29 que em formato decimal é 255.255.255.248. O número de hosts possíveis é 6 (2^3 -2=6), porque há 3 bits para host e dois deles são reservados. Já a nível de sub-redes há 16 (2^4=16) disponíveis, pois são usados 4 bits para identificar as mesmas.

3. Verifique e garanta que a conectividade IP interna na rede local LEI-RC é mantida. No caso de não existência de conetividade, reveja a atribuição de endereços efetuada e eventuais erros de encaminhamento por forma a realizar as correções necessárias. Explique como procedeu.

   R: Tal como se pode observar nas imagens seguintes, há conectividade entre os computadores dos diversos departamentos e o *router* RISP. Por exemplo, há conexão entre o *host* Aladin do departamento B e o *router* RISP, entre outros.

Figura 27: Conectividade entre computador Aladin e router RISP

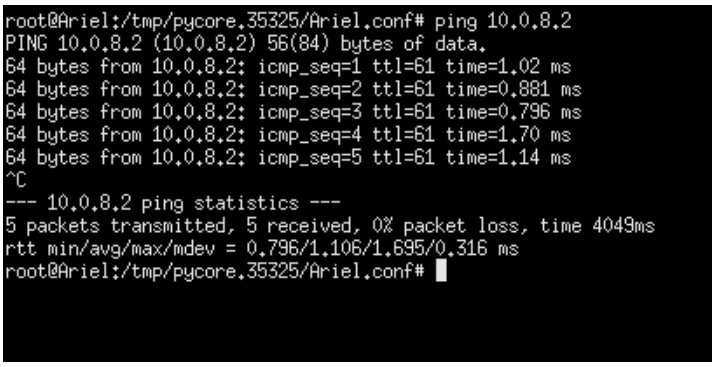


Figura 28: Conectividade entre computador Ariel e router RISP

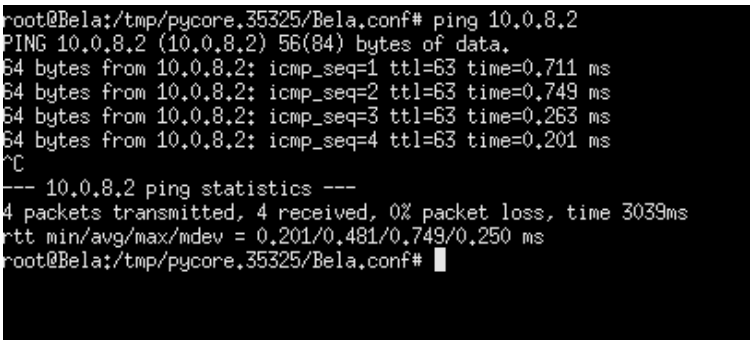


Figura 29: Conectividade entre computador Bela e router RISP

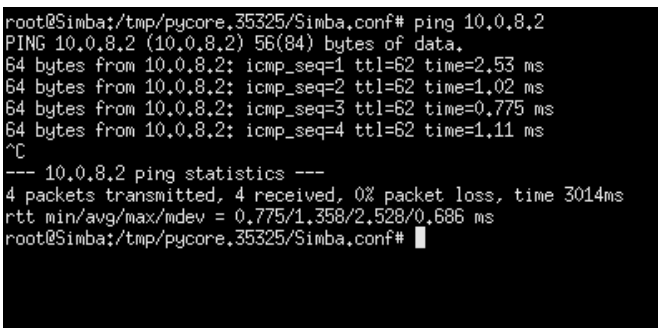


Figura 30: Conectividade entre computador Simba e router RISP

Já nas imagens que se seguem pode-se verificar que existe conectividade entre os *routers* dos diversos departamentos. Por exemplo, há conexão entre o *router* A e o *router* D, entre outros.

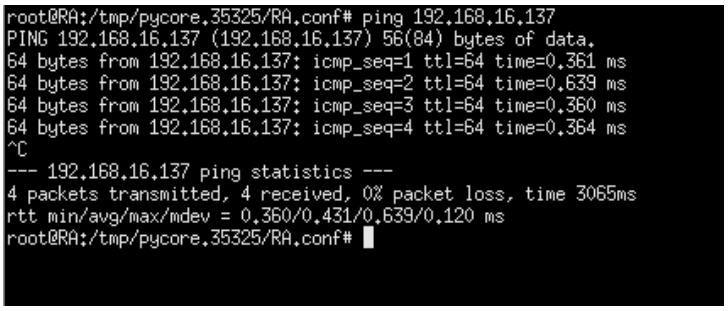


Figura 31: Conectividade entre router A e router D

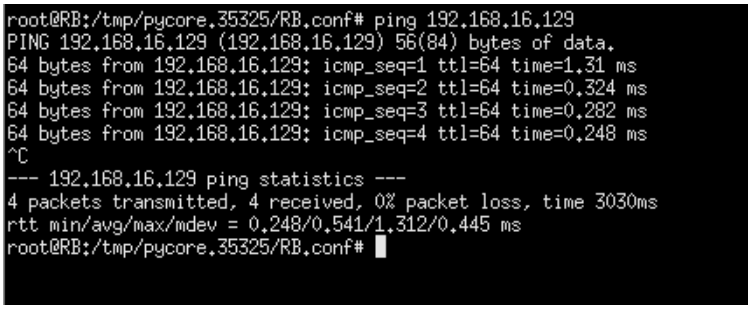


Figura 32: Conectividade entre router B e router A

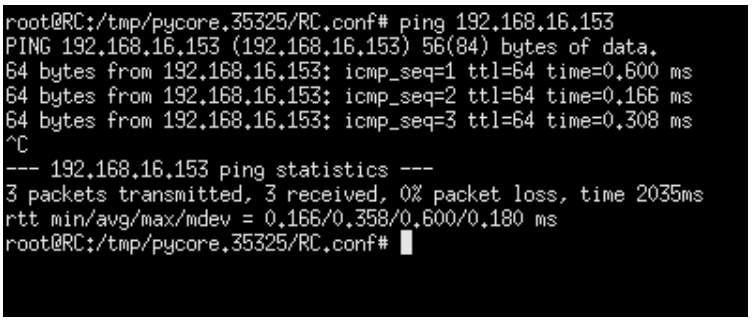


Figura 33: Conectividade entre router C e router B

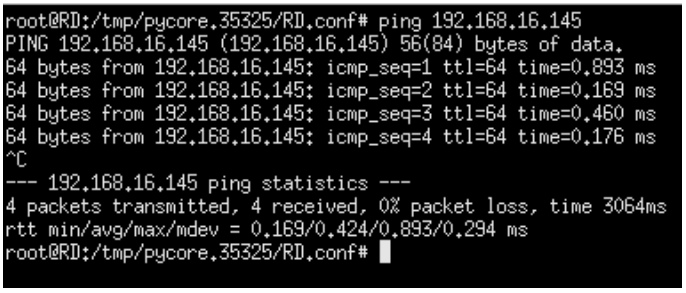


Figura 34: Conectividade entre router D e router C

Conclusão

Neste trabalho aprendemos a usar softwares de controle e administração de redes, como o Wireshark e o Core. Fundamentamos conceitos importantes, que são um objetivo final da aprendizagem desta unidade curricular como o processo de fragmentação, o endereçamento, sub-redes entre outros. Tivemos algumas dificuldades no utilizamento do Core mas com algum esforço conseguimos superar os percalços. Acreditamos que este trabalho foi relevante para nos tornarmos melhores profissionais e pessoas mais cultas.