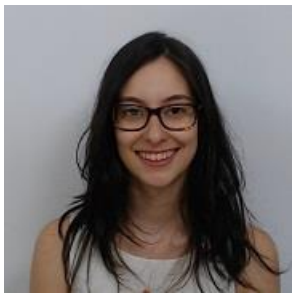
Relatório do Trabalho Prático 2

Redes de Computadores 2020/2021

a89506

Luís Miguel Lopes Pinto



a89599

Maria Beatriz Cardoso Gonçalves Barbosa e Moreira

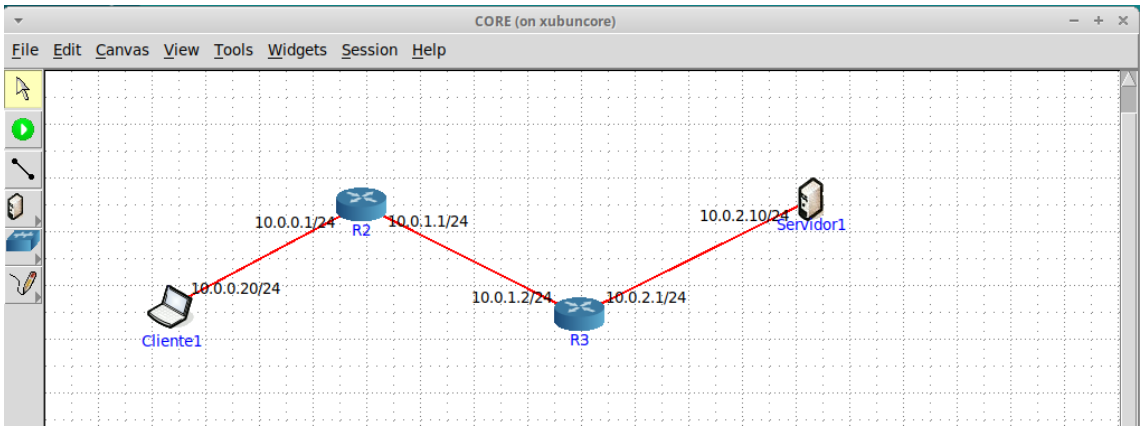
a89574

Pedro Almeida Fernandes

Conteúdo

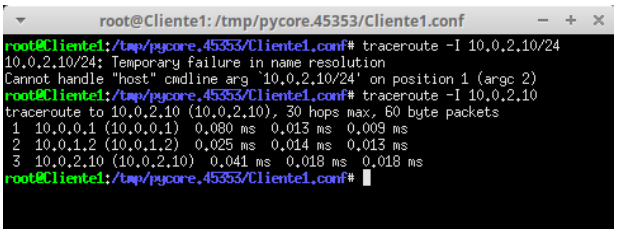
Parte 1

Captura de Tráfego IP



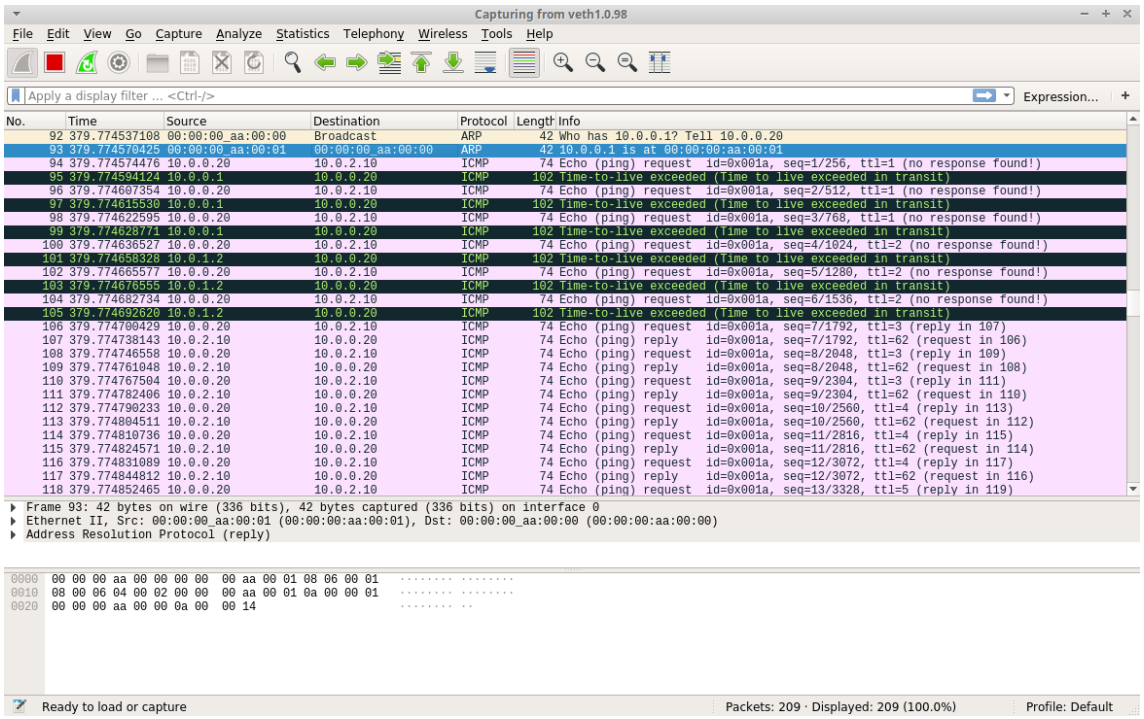
1 a) Active o wireshark ou o tcpdump no Cliente1. Numa shell doCliente1, execute o comando traceroute -I para o endereço IP do Servidor1.

R:.



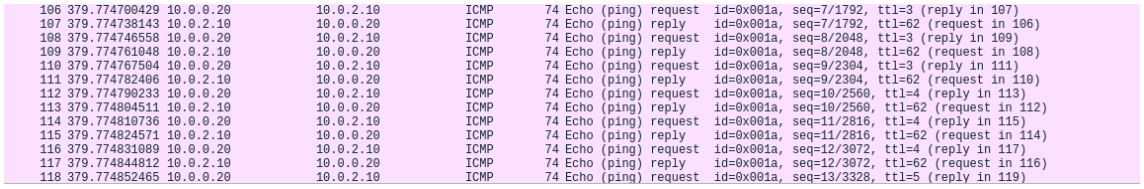
1 b) Registe e analise o tráfego ICMP enviado pelo Cliente1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.

R:. Observamos que do servidor1 foram enviados vários pacotes. Os primeiros pacotes tinham TTL = 1 e foram descartados por R2. De seguida foram enviados pacotes com TTL = 2 que também foram descartados, desta vez por R3. Por fim foram enviados pacotes com TTL = 3 que finalmente conseguiram chegar ao destino. Para cada pacote descartado foi recebida uma resposta proveniente do router que o descartou “Time to live exceeded”.

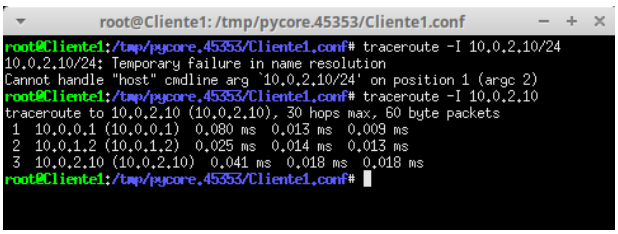


1 c) Qual deve ser o valor inicial mínimo do campo TTL para alcançar o Servidor1? Verifique na prática que a sua resposta está correta.

R:. O valor mínimo deverá ser TTL = 3. (confirmado na figura 1)



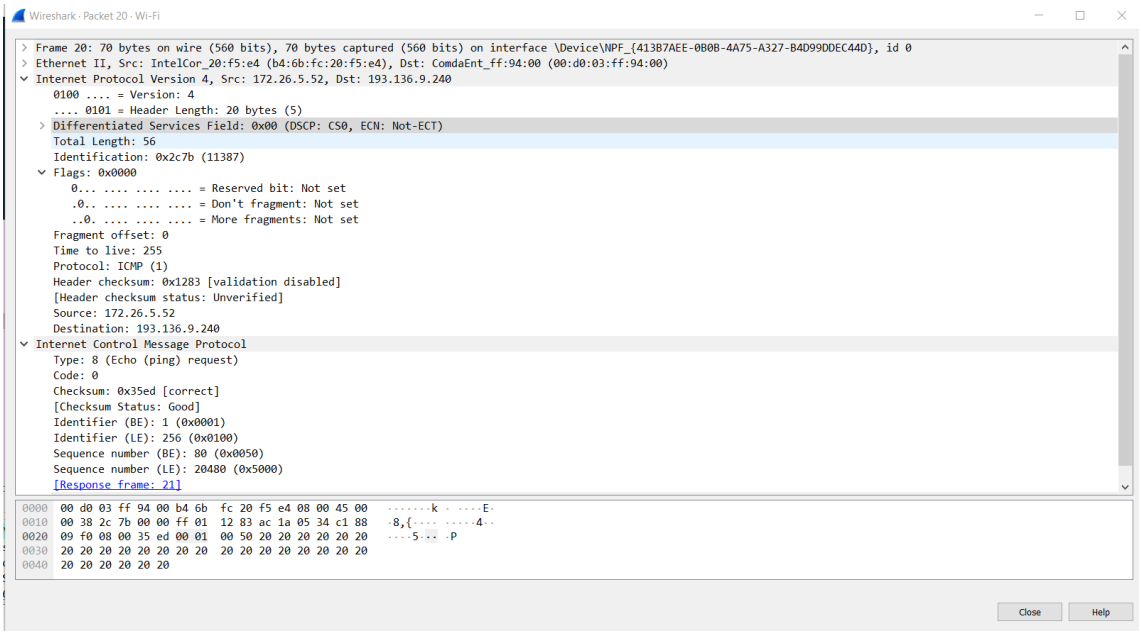
1 d) Calcule o valor médio do tempo de ida-e-volta (Round-TripTime) obtido?



R:. (0.041ms + 0.018ms + 0.018ms) / 3 = 0.026ms

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

R:. IP Source: 172.26.89.142



2 b) Qual é o valor do campo protocolo? O que identifica?

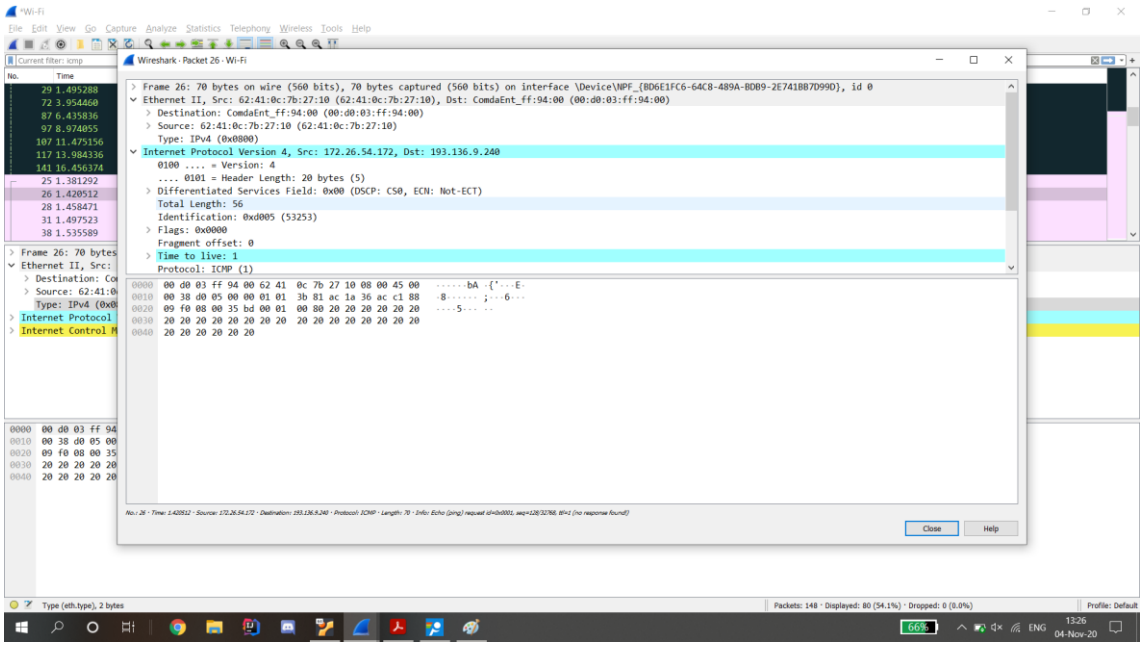
R:. IP Protocol: ICMP(1). O valor é 01. Identifica o ICMP. Confirmar na imagem da alínea anterior.

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

R:. Header Length: 20 bytes.

Total Length: 56 bytes.

Payload Length = Total Length – Header Length = 56 – 20 = 36 bytes.

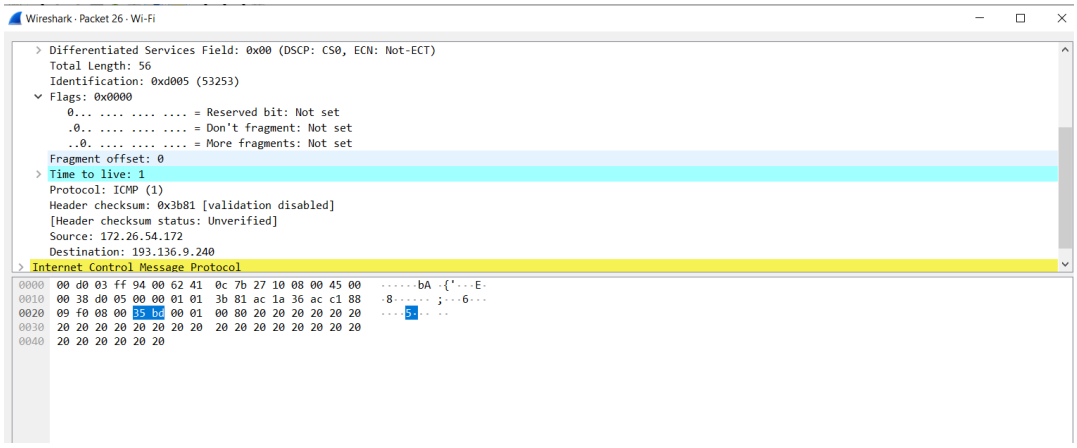


2 d) O datagrama IP foi fragmentado? Justifique.

R:. Fragment offset = 0 , prova que estamos no inicio do pacote.

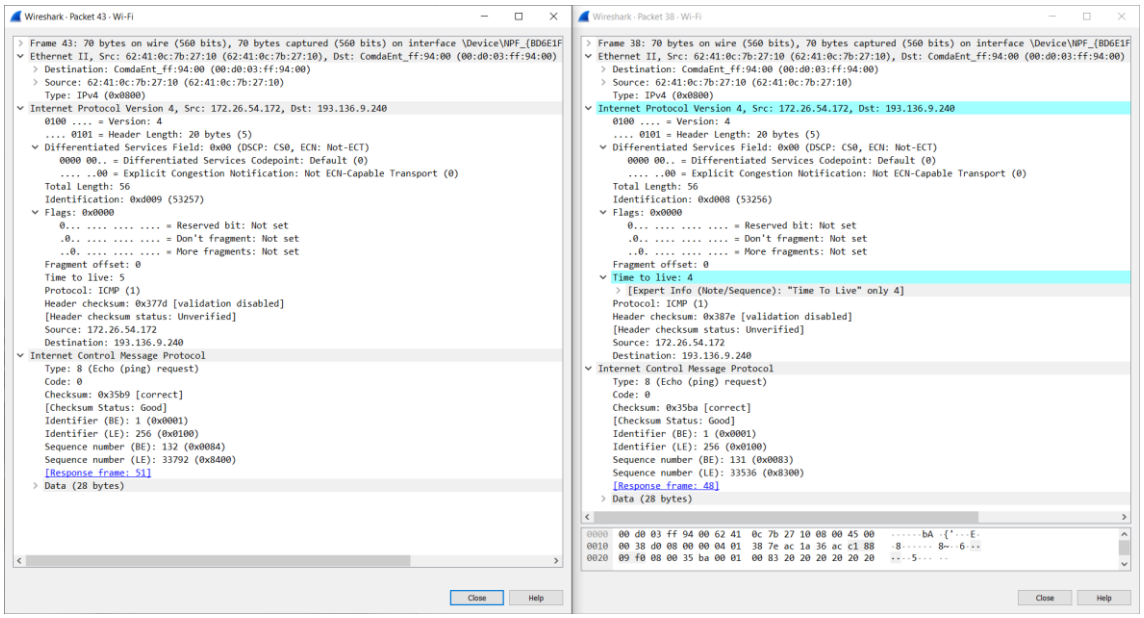
More fragments: Not set , prova que é a ultima parte do pacote.

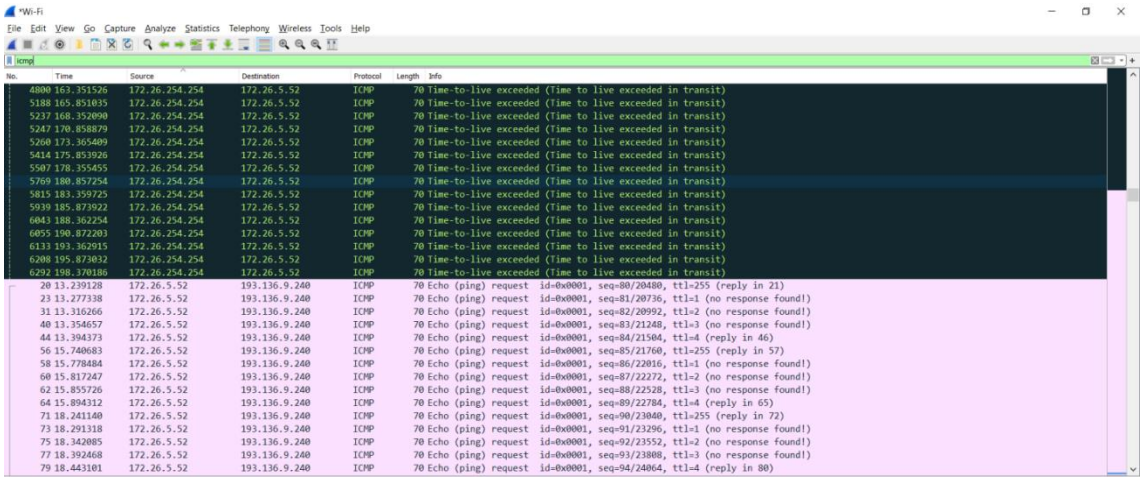
Logo, o datagrama IP não foi fragmentado.



2 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:. Os campos que variam conforme pacote IP são: Identification, TTL e Header checksum.

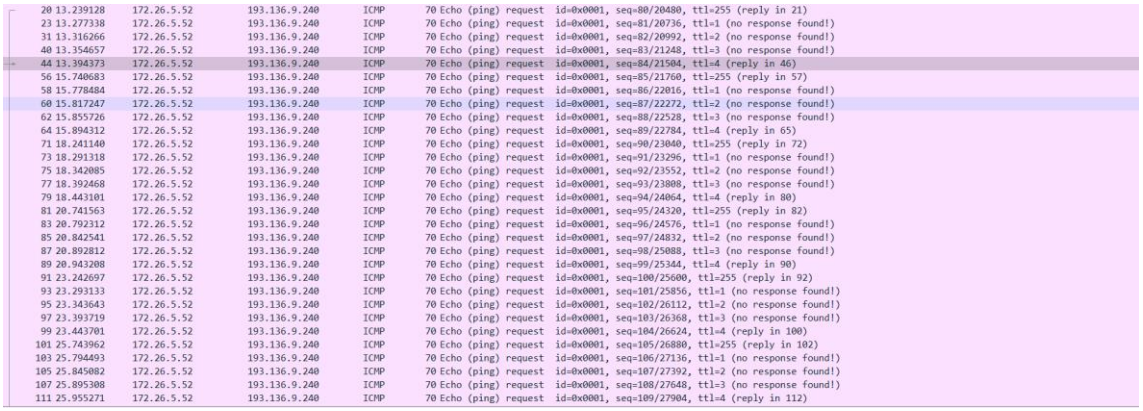




2 f) Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?

R:. O campo de identificação do datagrama IP tem os 8 bits iguais.

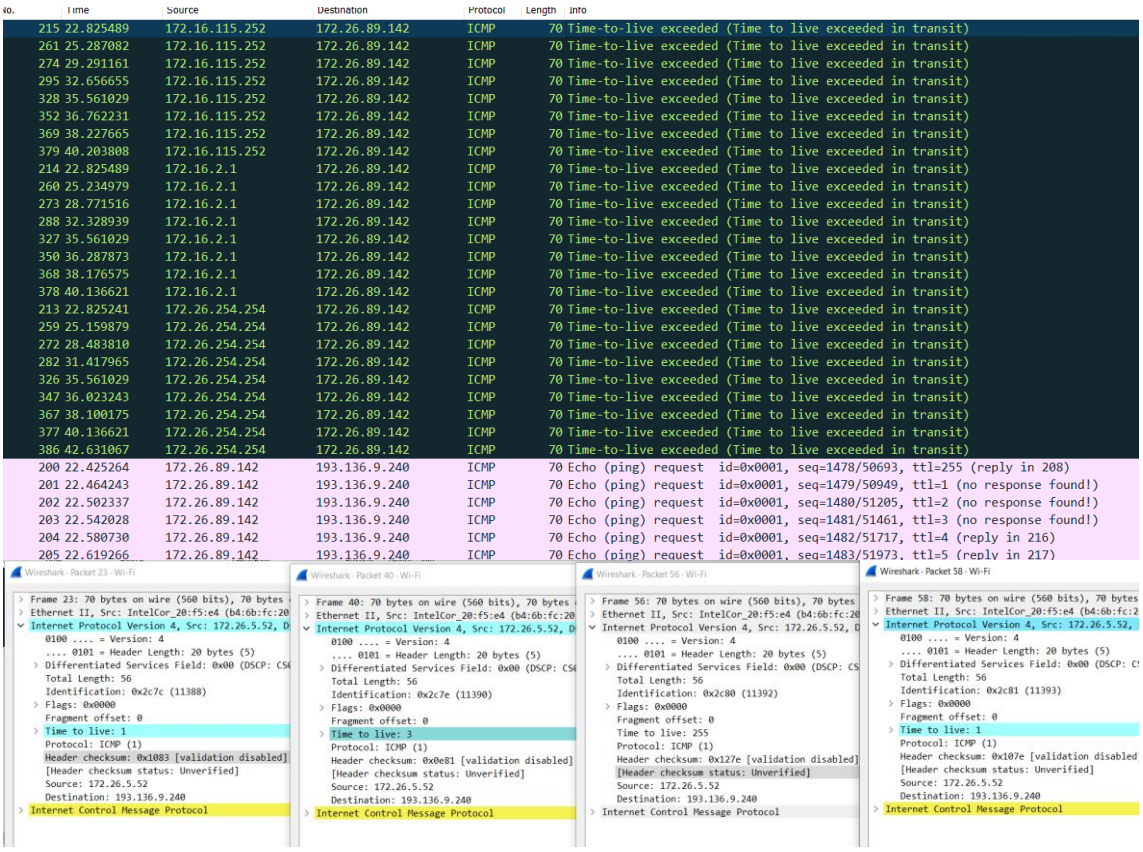
Vão sendo lançados pacotes com TTL sucessivamente maiores até que algum chegue ao destino.



20	13.239128	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=80/20480, ttl=255 (reply in 21)
23	13.277338	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=81/20736, ttl=1 (no response found!)
31	13.316266	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=82/20992, ttl=2 (no response found!)
40	13.354657	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=83/21248, ttl=3 (no response found!)
44	13.394373	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=84/21504, ttl=4 (reply in 46)
56	15.740683	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=85/21760, ttl=255 (reply in 57)
58	15.778484	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=86/22016, ttl=1 (no response found!)
60	15.817247	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=87/22272, ttl=2 (no response found!)
62	15.855726	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=88/22528, ttl=3 (no response found!)
64	15.894312	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=89/22784, ttl=4 (reply in 65)
71	18.241140	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=90/23040, ttl=255 (reply in 72)
73	18.291318	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=91/23296, ttl=1 (no response found!)
75	18.342085	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=92/23552, ttl=2 (no response found!)
77	18.392468	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=93/23808, ttl=3 (no response found!)
79	18.443101	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=94/24064, ttl=4 (reply in 80)
81	20.741563	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=95/24320, ttl=255 (reply in 82)
83	20.792312	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=96/24576, ttl=1 (no response found!)
85	20.842541	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=97/24832, ttl=2 (no response found!)
87	20.892812	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=98/25088, ttl=3 (no response found!)
89	20.943208	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=99/25344, ttl=4 (reply in 90)
91	23.242697	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=100/25600, ttl=255 (reply in 92)
93	23.293133	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=101/25856, ttl=1 (no response found!)
95	23.343643	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=102/26112, ttl=2 (no response found!)
97	23.393719	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=103/26368, ttl=3 (no response found!)
99	23.443701	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=104/26624, ttl=4 (reply in 100)
101	25.743962	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=105/26880, ttl=255 (reply in 102)
103	25.794493	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=106/27136, ttl=1 (no response found!)
105	25.845082	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=107/27392, ttl=2 (no response found!)
107	25.895308	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=108/27648, ttl=3 (no response found!)
111	25.955271	172.26.5.52	193.136.9.240	ICMP	70 Echo (ping) request id=0x0001, seq=109/27904, ttl=4 (reply in 112)

2 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:. TTL = 1. O TTL é constante. Deste modo, inferimos que a mensagem não conseguiu atingir o destino, o TTL foi decrementado de 1 para 0 e depois a mensagem foi descartada, e obviamente transmitiu essa mensagem novamente até a origem.



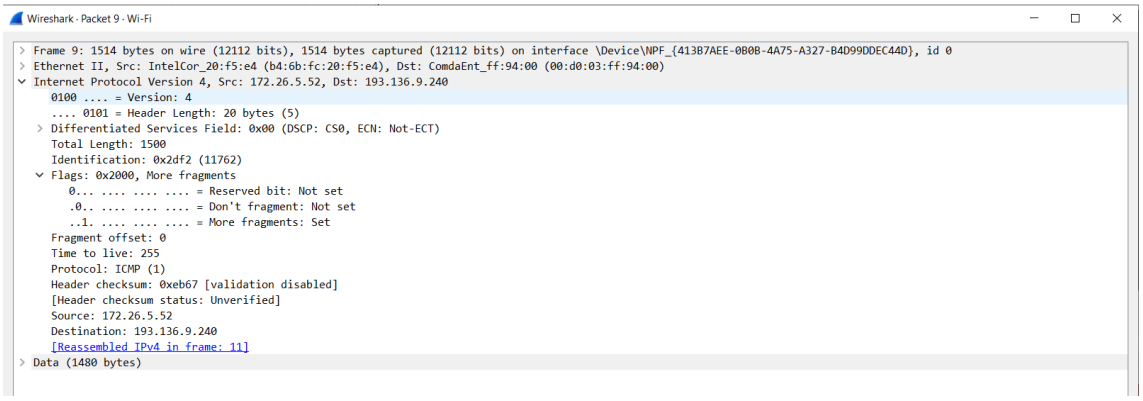
No.	Time	Source	Destination	Protocol	Length	Info
215	22.825489	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
261	25.287082	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
274	29.291161	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
295	32.656655	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
328	35.561029	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
352	36.762231	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
369	38.227665	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
379	40.203808	172.16.115.252	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
214	22.825489	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
260	25.234979	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
273	28.771516	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
288	32.328939	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
327	35.561029	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
350	36.287873	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
368	38.176575	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
378	40.136621	172.16.2.1	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
213	22.825241	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
259	25.159879	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
272	28.483810	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
282	31.417965	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
326	35.561029	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
347	36.023243	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
367	38.100175	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
377	40.136621	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
386	42.631067	172.26.254.254	172.26.89.142	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
200	22.425264	172.26.89.142	193.136.9.240	ICMP	70	Echo (ping) request id=0x0001, seq=1478/50693, ttl=255 (reply in 208)
201	22.464243	172.26.89.142	193.136.9.240	ICMP	70	Echo (ping) request id=0x0001, seq=1479/50949, ttl=1 (no response found!)
202	22.502337	172.26.89.142	193.136.9.240	ICMP	70	Echo (ping) request id=0x0001, seq=1480/51205, ttl=2 (no response found!)
203	22.542028	172.26.89.142	193.136.9.240	ICMP	70	Echo (ping) request id=0x0001, seq=1481/51461, ttl=3 (no response found!)
204	22.580730	172.26.89.142	193.136.9.240	ICMP	70	Echo (ping) request id=0x0001, seq=1482/51717, ttl=4 (reply in 216)
205	22.619266	172.26.89.142	193.136.9.240	ICMP	70	Echo (ping) request id=0x0001, seq=1483/51973, ttl=5 (reply in 217)

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

R:. O pacote enviado passou a ser de 3243 bytes. Mas, como Max Transfer Unit (MTU) é 1500 (constituído por 20 bytes de header e 1480 de dados) podemos concluir que não foi possível transferir o pacote de uma só vez. Deste modo, o pacote teve de ser fragmentado de modo a que o seu transporte fosse possibilitado.

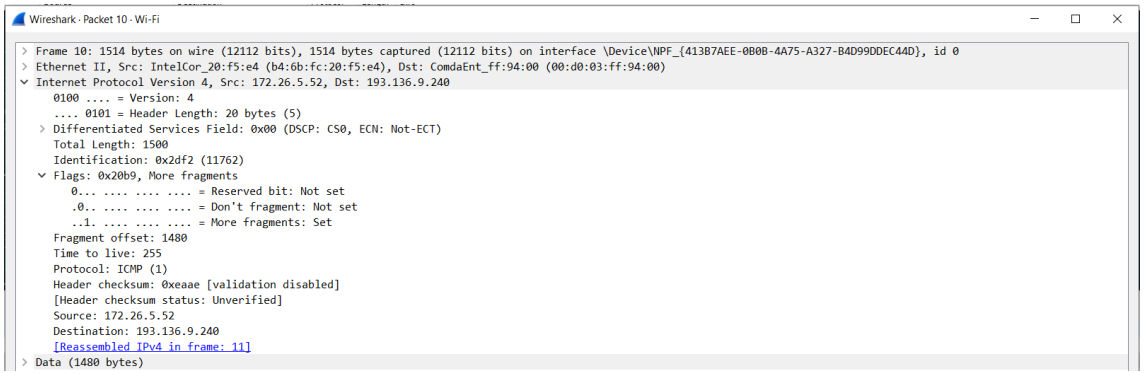
3 b) Imprima o primeiro fragmento do datagrama IP segmentado. Que informação no cabeçalho indica que o datagrama foi fragmentado? Que informação no cabeçalho IP indica que se trata do primeiro fragmento? Qual é o tamanho deste datagrama IP?

R:. No cabeçalho temos o offset a 0 (pelo que temos a certeza que estamos no primeiro fragmento do pacote) e a flag More fragments a 1 (o que nos indica que este é seguido de mais fragmentos do pacote que lhe deu origem). O datagrama em si é constituído por um header de 20 bytes e uma parte de dados de 1480 bytes, resultando assim em datagramas de 1500 bytes.



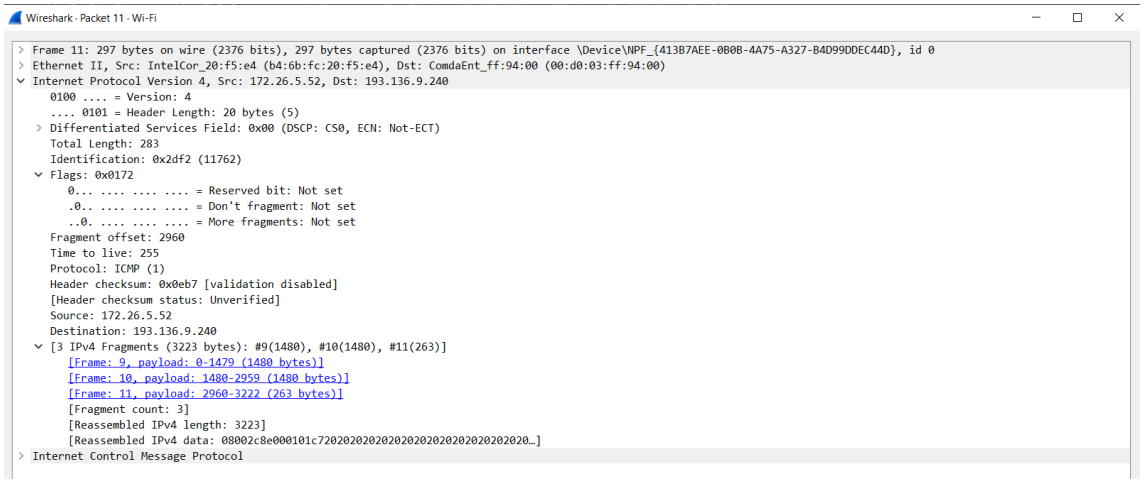
3 c) Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1ºfragmento? Há mais fragmentos? O que nos permite afirmar isso?

R:. Podemos concluir que não estamos no primeiro fragmento, uma vez que o offset é 1480 e não zero, ou seja, já houve a transferência de 1480 bytes do pacote inicial. Sabemos também que há mais fragmentos, uma vez que a flag More fragments estar a 1.



3 d) Quantos fragmentos foram criados a partir do datagrama original? Como se deteta o último fragmento correspondente ao datagrama original?

R:. Foram criados 3 fragmentos a partir do datagrama original. O último fragmento tem a flag More fragments a 0 (o que indica que não há mais fragmentos a seguir deste) (e como o offset não está a 0 sabemos que este não é um pacote não fragmentado), pelo que sabemos que este pertence a um datagrama que foi fragmentado. Podemos concluir que é o último do datagrama que estávamos a analisar, uma vez que é o ultimo fragmento com o id do datagrama inicial.



3 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 vão mudando entre diferentes fragmentos são:

- a flag More fragments - que será 1 em todos menos no ultimo, pelo que nos permite distinguir de forma direta o ultimo fragmento;

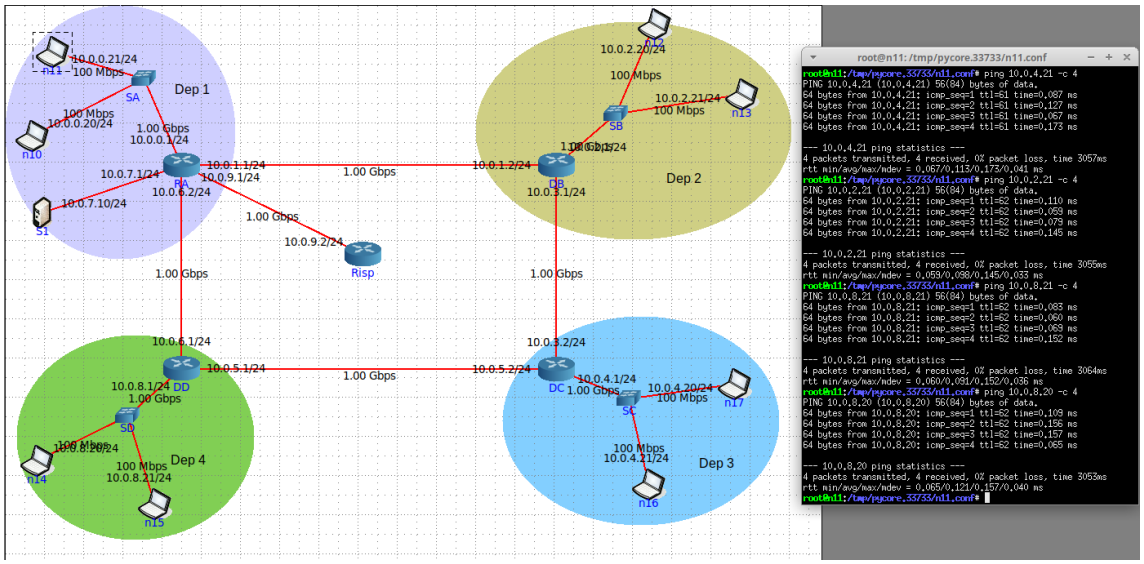
- o offset - que corresponde ao número de bytes do datagrama inicial que já foram enviados, pelo que se ordenarmos os fragmentos por ordem crescente de offset podemos obter o datagrama ordenado pela ordem em que os dados estavam no datagrama original;
- a length - será garantidamente máxima em todos os fragmentos, exceto no último, em que pode não chegar a esse valor, pelo que é mais um fator que nos pode permitir distinguir o último fragmento dos restantes.

Parte 2

Endereço e Encaminhamento IP

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

R:.



1 b) Tratam-se de endereços públicos ou privados? Porquê?

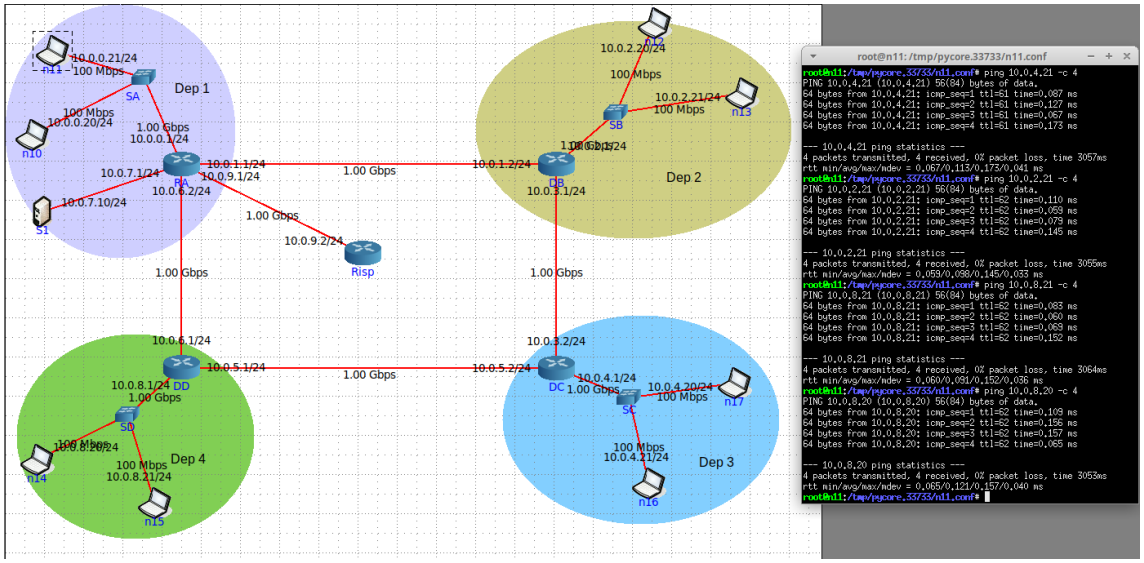
R:. Privado - porque segundo o protocolo/norma RFC 1918 o prefixe 10.0 é identificador de redes privadas

1 c) Porque razão não é atribuído um endereço IP aos switches?

R:. Porque estão numa camada a baixo (Layer 2 - Link Layer)

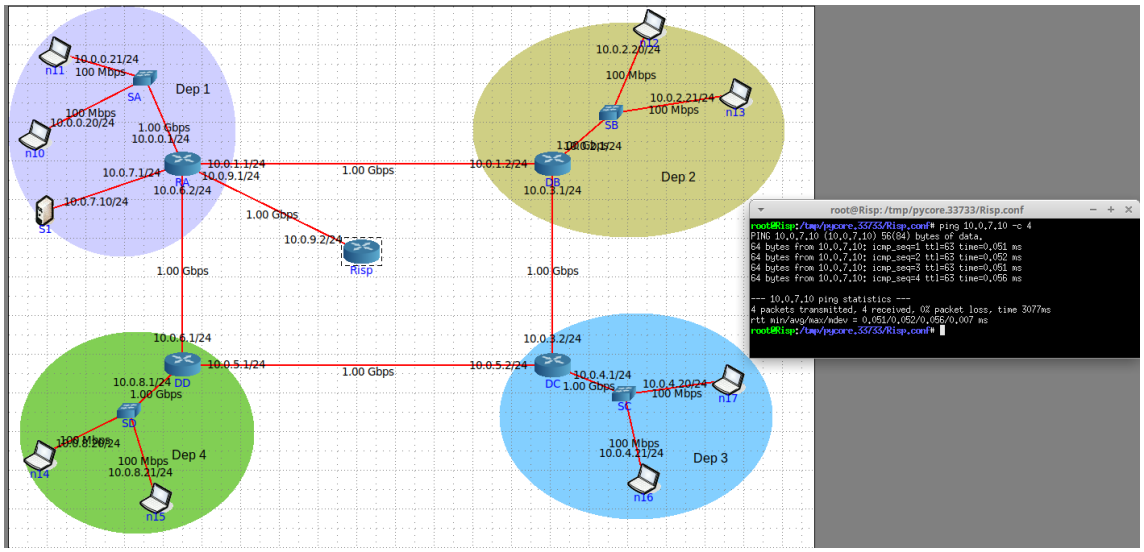
1 d) Usando o comando ping certifique-se que existe conectividade IP entre os laptops dos vários departamento se o servidor do departamento A (basta certificar-se da conectividade de um laptop por departamento).

R:.



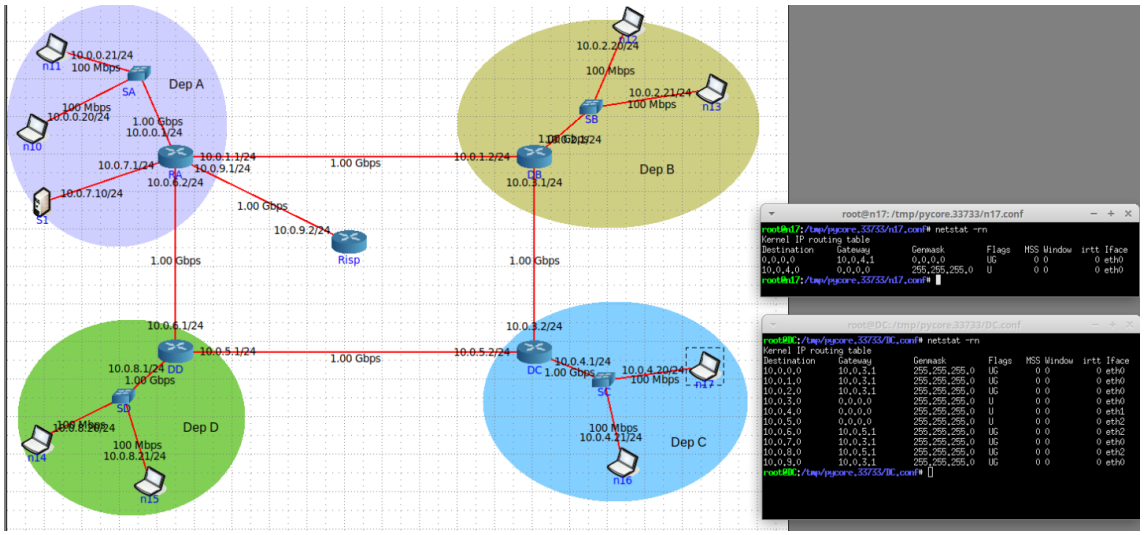
1 e) Verifique se existe conectividade IP do router de acesso RISP para o servidor S1.

R:.



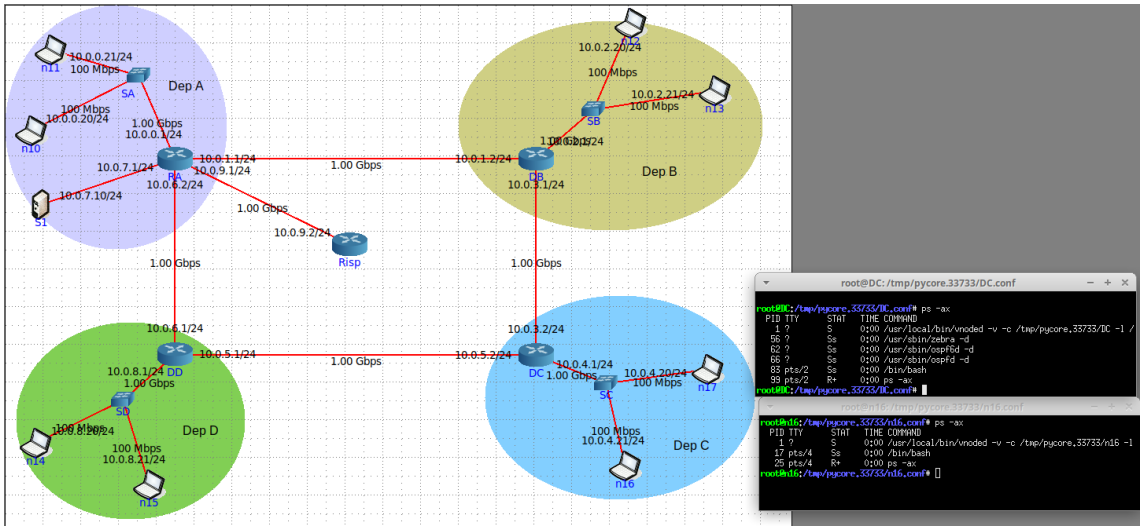
2 a) Execute o comando netstat –rn por forma a poder consultara tabela de encaminhamento unicast (IPv4). Inclua no seu relatório as tabelas de encaminhamento obtidas; interprete as várias entradas de cada tabela. Se necessário, consulte o manual respetivo (man netstat).

R:.



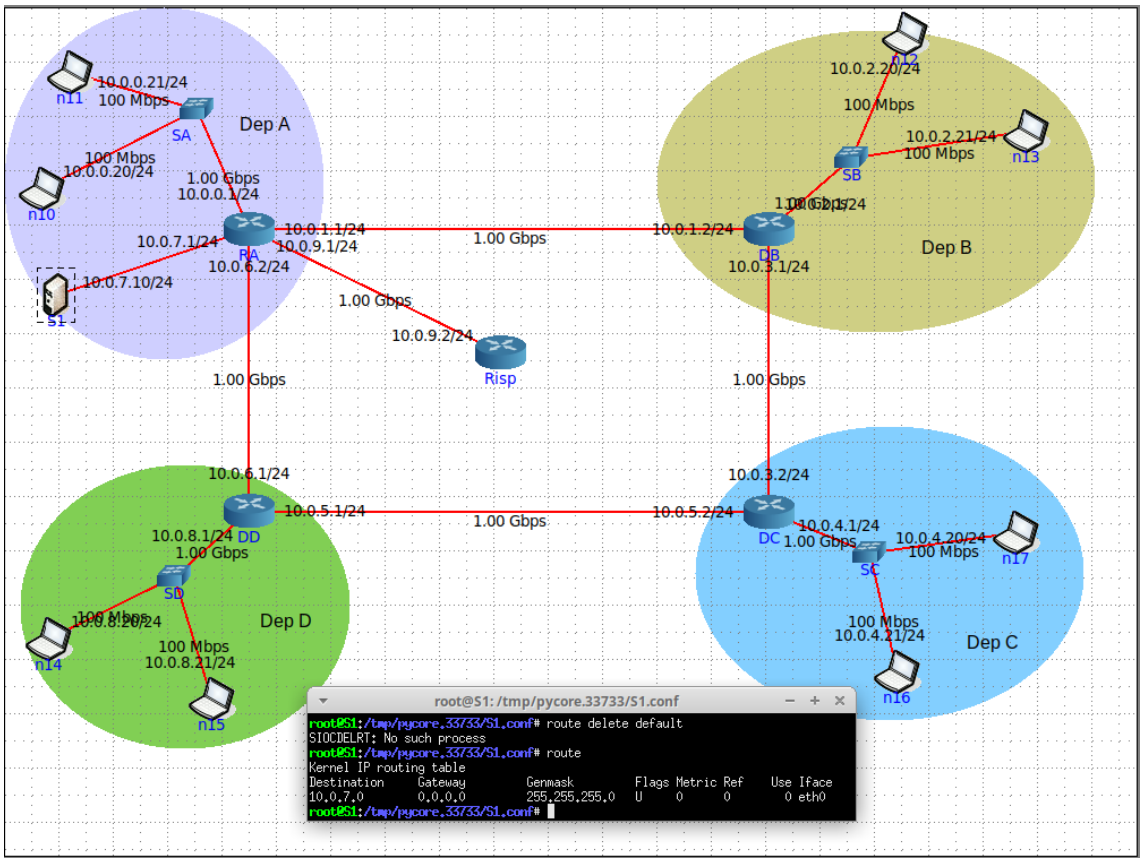
2 b) Diga, justificando, se está a ser usado encaminhamento estático ou dinâmico (sugestão: analise que processos estão a correr em cada sistema, por exemplo, ps -ax).

R:. Dinâmicos- daemon cria a rota.



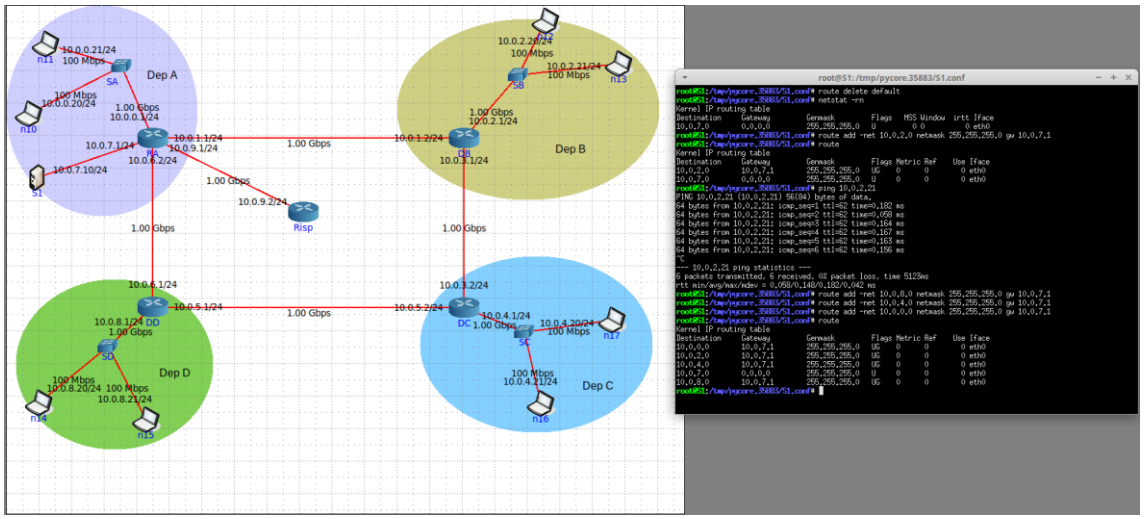
2 c) Admita que, por questões administrativas, a rota por defeito(0.0.0.0 ou default) deve ser retirada definitivamente da tabela de encaminhamento do servidor S1 localizado no departamento A. Use o comando route delete para o efeito. Que implicações tem esta medida para os utilizadores da organização MIEI-RC que acedem ao servidor. Justifique.

R:. S1 perde conectividade com todos os hosts não presentes na rede local. Removendo a rede default de S1 não tem rota de trafego definida para hosts não locais, assim consegue enviar dados, mas não consegue receber. Consegue receber informação dos outros departamentos (porque as outras redes sabem onde ele está), mas não consegue enviar, pois não sabe para onde enviar a mensagem (porque a rota default foi retirada). Assim este é apenas capaz de comunicar com o router do departamento A (não conseguindo estabelecer comunicação nem com os hosts deste departamento nem com hosts ou routers dos restantes).



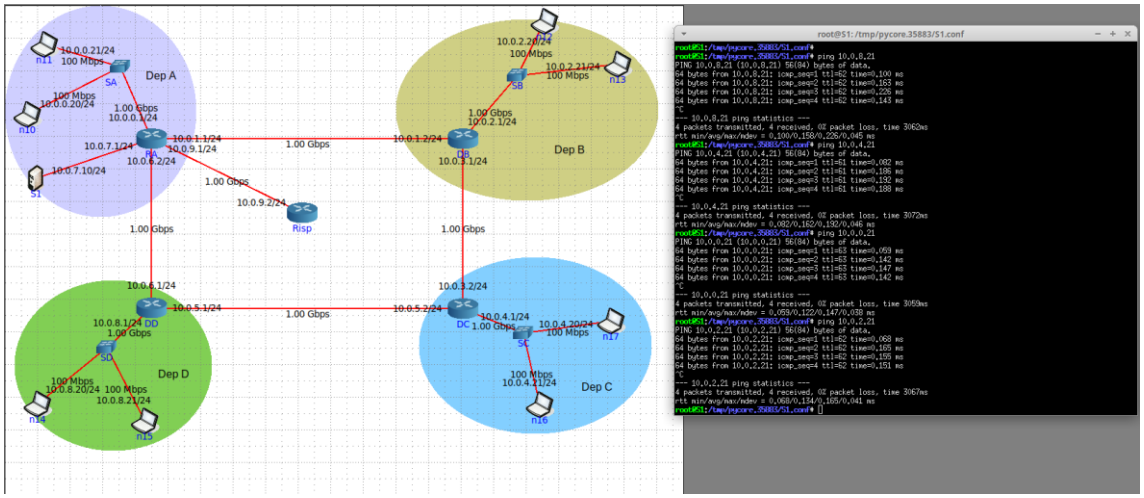
2 d) Adicione as rotas estáticas necessárias para restaurar a conectividade para o servidor S1, por forma a contornar a restrição imposta na alínea c). Utilize para o efeito o comando route add e registe os comandos que usou.

R:.



2 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:.

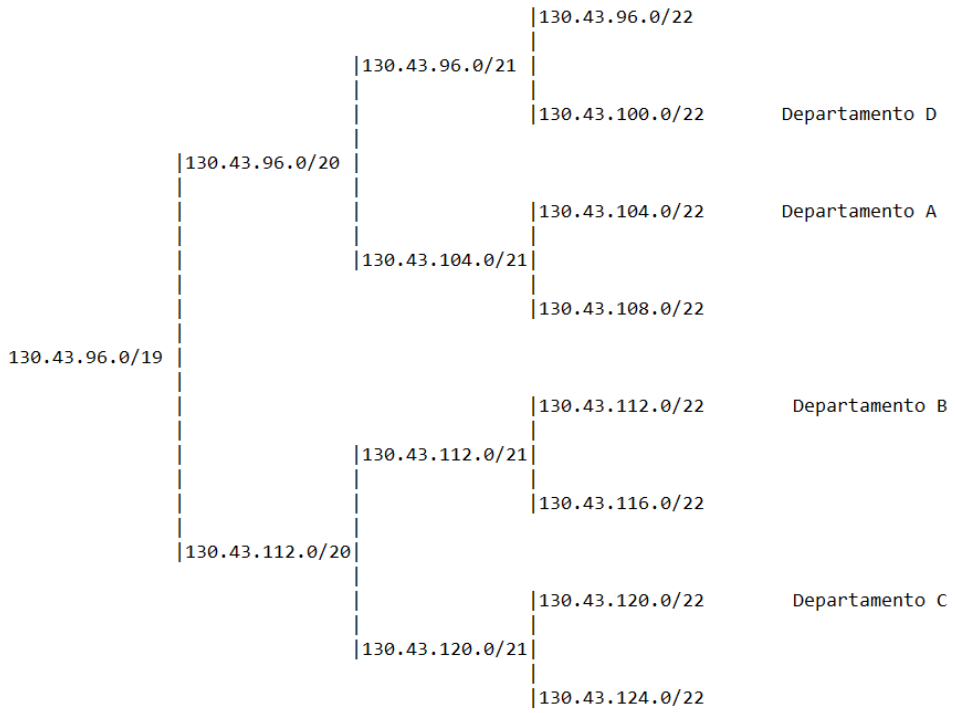


Definição de Sub-redes

3 1) Considere que dispõe apenas do endereço de rede IP130.XX.96.0/19, em que XX é o decimal correspondendo ao seu número de grupo (PLXX). Defina um novo esquema de endereçamento para as redes dos departamentos (mantendo a rede de acesso e core inalteradas) e atribua endereços às interfaces dos vários sistemas envolvidos. Assuma que todos os endereços de sub-redes são usáveis. Deve justificar as opções usadas.

R:. 10000010.00101011.011|000|00.00000000

3 bits para sub-rede uma vez que 4 são necessárias e assim ainda sobrariam outras tantas para possíveis expansões futuras (sendo que mesmo que considerássemos o 1º e o ultimo como reservados ainda teríamos mais duas sub-redes disponíveis). E assim sobram 10 bits para hosts. Se forem atribuídos como demonstrado no diagrama os departamentos A e B têm a possibilidade de diminuir o tamanho da mascara (passando a 21) e assim poder aceitar mais hosts. Isto deve-se ao facto de as sub-redes cuja mascara dista um bit não estar atribuída. (como isso não ocorre para C e D, estes têm de manter o tamanho da sua mascara, visto que estamos a considerar que estamos a reservar o primeiro e o ultimo endereços).

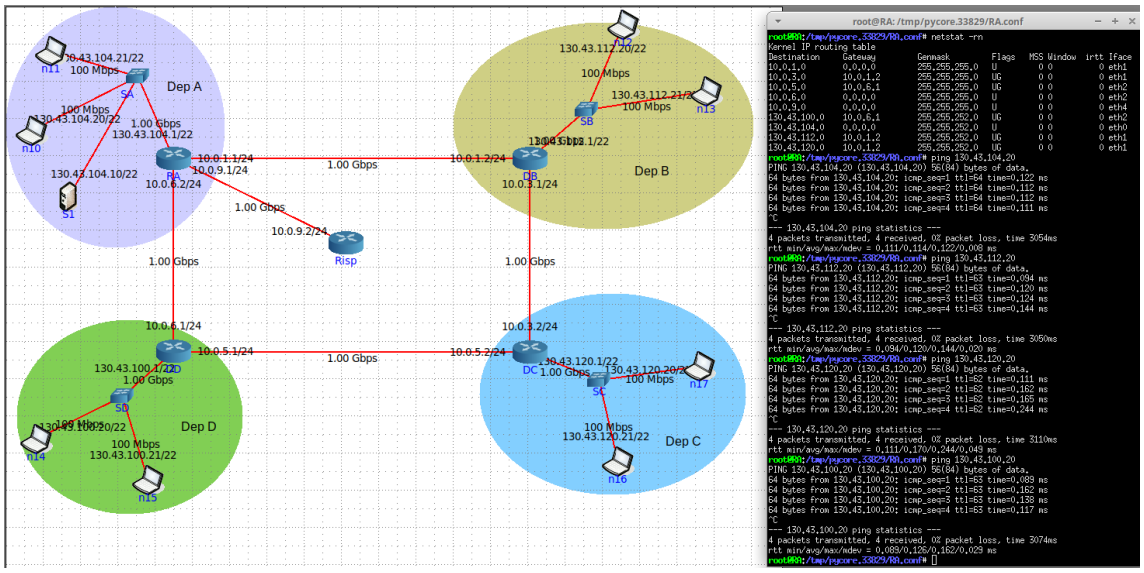


3 2) Qual a máscara de rede que usou (em formato decimal)? Quantos hosts IP pode interligar em cada departamento? Justifique.

R:. Mascara = 22; Número de Hosts da Rede = (2^(10)-1)= 1023.

3 3) Garanta e verifique que conectividade IP entre as várias redes locais da organização MIEI-RC é mantida. Explique como procedeu.

R:. Para esta alínea mudamos manualmente os endereços (e respetivas mascaras) de cada uma das sub-redes, de modo a ficarem endereçadas como o esquematizado na alínea anterior. De seguida, executamos o comando netstat -rn para verificarmos que na tabela de endereçamento apareciam caminhos para todos os departamentos e demos ping a um host de cada departamento para verificar que a ligação era efetivamente estabelecida.



Conclusão

Ao longo da realização deste trabalho prático fomos capazes de solidificar vários dos conhecimentos que nos tinham sido passados nas aulas teóricas desta cadeira no âmbito do Internet Protocol (com o endereçamento IPv4), CIDR e subnetting.

Começamos por perceber melhor como funciona o TTL (Time To Live) através dos exercícios propostos, tendo chegado à conclusão de que este representa o número de saltos que o pacote tem que dar antes de chegar ao destino, pelo que deve ser sempre o menor possível. Constatamos também que para evitar que pacotes fiquem presos em ciclos existe um valor máximo de saltos que este pode executar. O TTL assume esse valor e vai sendo decrementado a cada salto, sendo descartado caso chegue a zero.
No que toca ao formato do datagrama IP, verificamos que é constituído por duas partes (heather e payload). Fomos capazes de verificar que a analise cuidada do cabeçalho permite-nos detetar se o pacote foi fragmentado e em caso afirmativo ser capazes de o reconstruir.

De seguida, foram-nos relembrados os conceitos de endereços públicos, privados (baseado no norma RFC 1918) routers e switches (e a utilidade destas ultimas para evitar a criação de uma subnet por host, como ocorreria se estes estivessem diretamente ligados ao router).

Posteriormente analisamos tabelas de endereçamento. Vimos que caso a rota default a um servidor fosse retirada este deixaria de ser capaz de comunicar com routers e hosts que não estivessem na sua rede local. Para resolver este problema (através de encaminhamento estático) recorremos à inserção das rotas de encaminhamento especificas para cada departamento (não local).

Por ultimo, estudamos a temática do subnetting. Este permitiu-nos que a partir de um único endereço inicial criássemos diversas sub-redes para alimentar diversos departamentos. Concluímos também que a sua realização cuidada permite que esta esteja preparada para possíveis expansões futuras (sacrificando para isso o número máximo de hosts que se podem ligar a cada uma).