TP2: Protocolo IP

Henrique Neto, Sara Marques e Tiago Gomes

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal
e-mail: {a89618,a89477,a78141}@alunos.uminho.pt

Resumo O principal objetivo deste trabalho é o estudo do Internet Protocol (IP) nas suas principais vertentes, nomeadamente: estudo do formato de um pacote ou datagrama IP; fragmentação de pacotes IP; endereçamento IP; e encaminhamento IP.

1 Datagramas IP e Fragmentação

1.1 Topologia Core

Neste exercício pretendeu-se estudar a **Topologia Core** da figura 1 com o fim de verificar o comportamento do *traceroute*.

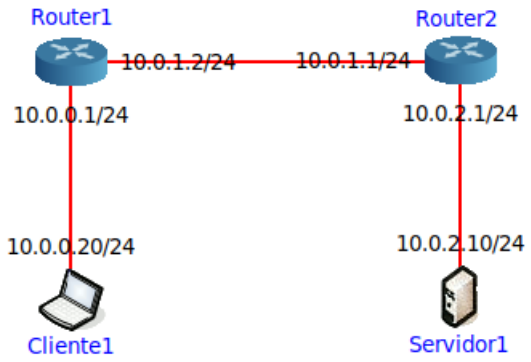

Figura 1. Topologia Core 1

a) **Numa shell do Cliente1, execute o comando traceroute -I para o endereço IP do *Servidor1*.**

```
root@Cliente1:/tmp/pycore.42381/Cliente1.conf# traceroute -I 10.0.2.10
traceroute to 10.0.2.10 (10.0.2.10), 30 hops max, 60 byte packets
1  10.0.0.1 (10.0.0.1)  0.038 ms  0.008 ms  0.005 ms
2  10.0.1.1 (10.0.1.1)  0.013 ms  0.005 ms  0.005 ms
3  10.0.2.10 (10.0.2.10)  0.014 ms  0.009 ms  0.007 ms
```

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=1/256, ttl=1 (no response found!)
2	0.000026327	10.0.0.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
3	0.000037946	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=2/512, ttl=1 (no response found!)
4	0.000042723	10.0.0.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
5	0.000046258	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=3/768, ttl=1 (no response found!)
6	0.000049281	10.0.0.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
7	0.000052999	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=4/1024, ttl=2 (no response found!
8	0.000064461	10.0.1.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
9	0.000067520	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=5/1280, ttl=2 (no response found!
10	0.000071855	10.0.1.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
11	0.000074513	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=6/1536, ttl=2 (no response found!
12	0.000078689	10.0.1.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
13	0.000081948	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=7/1792, ttl=3 (reply in 14)
14	0.000094390	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=7/1792, ttl=62 (request in 13)
15	0.000098277	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=8/2048, ttl=3 (reply in 16)
16	0.000105608	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=8/2048, ttl=62 (request in 15)
17	0.000108547	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=9/2304, ttl=3 (reply in 18)
18	0.000114360	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=9/2304, ttl=62 (request in 17)
19	0.000117751	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=10/2560, ttl=4 (reply in 20)
20	0.000123459	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=10/2560, ttl=62 (request in 19)
21	0.000126198	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=11/2816, ttl=4 (reply in 22)
22	0.000132090	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=11/2816, ttl=62 (request in 21)
23	0.000134814	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=12/3072, ttl=4 (reply in 24)
24	0.000140581	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=12/3072, ttl=62 (request in 23)
25	0.000144024	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=13/3328, ttl=5 (reply in 26)
26	0.000149750	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=13/3328, ttl=62 (request in 25)
27	0.000152503	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=14/3584, ttl=5 (reply in 28)
28	0.000158922	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=14/3584, ttl=62 (request in 27)
29	0.000161842	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=15/3840, ttl=5 (reply in 30)
30	0.000168207	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=15/3840, ttl=62 (request in 29)
31	0.000171761	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0033, seq=16/4096, ttl=6 (reply in 32)
32	0.000178015	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0033, seq=16/4096, ttl=62 (request in 31)

- Como esperado, o comando traceroute enviou uma série de mensagens com o TTL crescente, que provocou que cada router em que o TTL excedia devolvesse uma mensagem de erro até que o TTL atingisse um valor alto o suficiente para chegar ao destino. Desta forma ficamos com o conhecimento da distância de saltos a que se encontra o Servidor e que interfaces percorre até chegar lá.

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

- No contexto desta topologia o TTL de uma mensagem entre o *Cliente1* e o *Servidor1* é decrementado duas vezes ao ser encaminhada pelo *router* 1 e 2. Assim, concluímos para que a mensagem não seja rejeitada, ou seja, para que o valor de TTL não chegue a 0 este tem de ter um valor inicial maior ou igual a 3. Isto é visível no tráfego capturado pelo *Wireshark*: a partir do momento em que as mensagens do traceroute atingem o valor 3, o cliente deixa de receber mensagens de "Time-to-live exceeded"das interfaces intermédias e começa a receber "Echo (ping) reply" proveniente dos servidor.

d) **Calcule o valor médio do tempo de ida-e-volta (Round-Trip Time) obtido?**

- A partir do output do comando *traceroute* podemos observar que o RTT médio de ida-e-volta tem o valor de 0.007 milisegundos.

1.2 *Traceroute* para a interface "marco.uminho.pt"

Neste exercício pretendeu-se estudar o *traceroute* na nossa interface nativa até ao host "marco.uminho.pt".

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

- O endereço de IP da interface é 172.26.20.50 .

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

- O campo de protocolo do cabeçalho do IP tem o valor 1, traduzindo que se trata de uma mensagem ICMP (Internet Control Message Protocol).

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?**

- O cabeçalho IPv4 tem 20 bytes de tamanho segundo o *header length*. Tendo em conta que a mensagem tem 56 bytes de tamanho (length = 0x38) podemos concluir que o tamanho do payload é 36.

d) **O datagrama IP foi fragmentado?**

- Não, visto que as flags de fragmentação e mais fragmentos têm o valor 0.

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.**

- A Identificação (id) e o "Time to Leave"(TTL) mudam de mensagem para mensagem.

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

- Ambos são incrementados a cada mensagem.

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?**

- Neste tipo de mensagens ICMP existem dois headers IP. O primeiro correspondente à mensagem "TTL exceeded"em si e o segundo correspondente à mensagem ao qual o TTL foi excedido.
  No primeiro header o TTL toma os valores de 255, 254 e 253 respetivamente. Isto deve-se ao facto de que à medida que o *traceroute* envia mensagens ICMP que não conseguem chegar ao destino final, os routers intermédios enviam sempre a mensagem ICMP do tipo "TTL exceeded"com o TTL inicializado no seu valor máximo (255). Dado a natureza do *traceroute* estes routers encontraram-se a uma distancia de saltos cada vez maior do destino e daí o TTL recebido pelo cliente decrementa à medida que as mensagens chegam.
  No segundo header o TTL mantém-se sempre constante e com o valor de 1. Obviamente isto é o esperado visto que estes cabeçalhos estão a ser comunicadas de volta pelo facto do seu TTL ter chegado a esse valor.

1.3 Fragmentação de Pacotes IP

Neste exercício pretendeu-se analisar a fragmentação de pacotes IP. Para tal foi estudado um ambiente semelhante ao exercício anterior, no qual a única alteração era o tamanho do pacote ser 3267 bytes.

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

- Tendo em conta que o *Maximum Transmission Unit* (MTU) da rede indica o tamanho máximo do pacote é 1500 bytes e que a mensagem tem 3267 bytes, esta obrigatoriamente terá de ser fragmentada em pelo menos 3 fragmentos.

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?**

- Segundo o cabeçalho IP a mensagem tem 1500 bytes de tamanho e foi fragmentada visto que a flag de fragmentação não é zero. Trata-se do primeiro fragmento visto que o *offset* de fragmentação tem o valor zero.

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?**

- Segundo o cabeçalho IP a mensagem está fragmentada visto que a flag de fragmentação não é zero e possui mais fragmentos para além deste, uma vez que a flag de mais fragmentos tem o valor 1. Podemos também observar que não se trata do primeiro fragmento, visto que o offset tem o valor de 1480.

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

- Foram criados 3 fragmentos, entre os quais o último pode ser diferenciado por ter a flag de mais fragmentos com o valor 0.

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.**

- Os únicos valores que mudam entre os diagramas são o tamanho das mensagens, o *checksum*, o offset do fragmento e a flag de mais fragmentos. O tamanho das mensagens e checksum mudam devido a cada pacote possuir payloads e características diferentes e não participam na reconstrução do datagrama original. Por outro lado, as flags de mais fragmentos permitem identificar o último elemento e o offset de cada fragmento permite identificar qual a distância que cada fragmento se encontra do início da mensagem e consecutivamente descobrir o fragmento inicial. Sendo assim, é possível reconstruir os vários fragmentos e formar o datagrama inicial.

2 Endereçamento e Encaminhamento IP

2.1 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.**

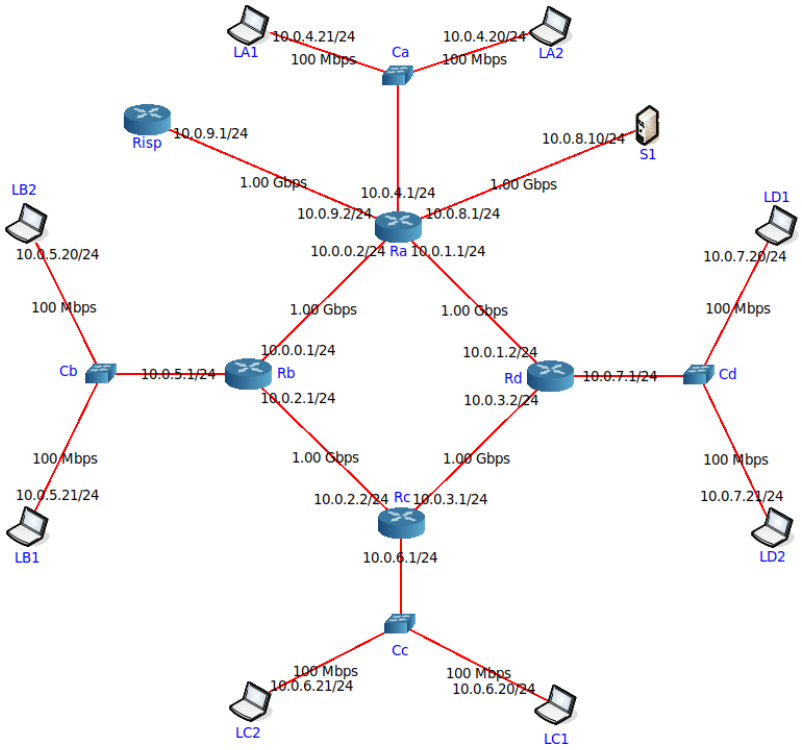

Figura 2. Topologia Core 2

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

- Tratam-se de endereços IP privados, pois pertencem à gama de endereços 10.0.0.0 - 10.255.255.255 /8.

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

- Porque os switches apenas redirecionam a ligação, não precisando por isso de um endereço IP próprio.

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

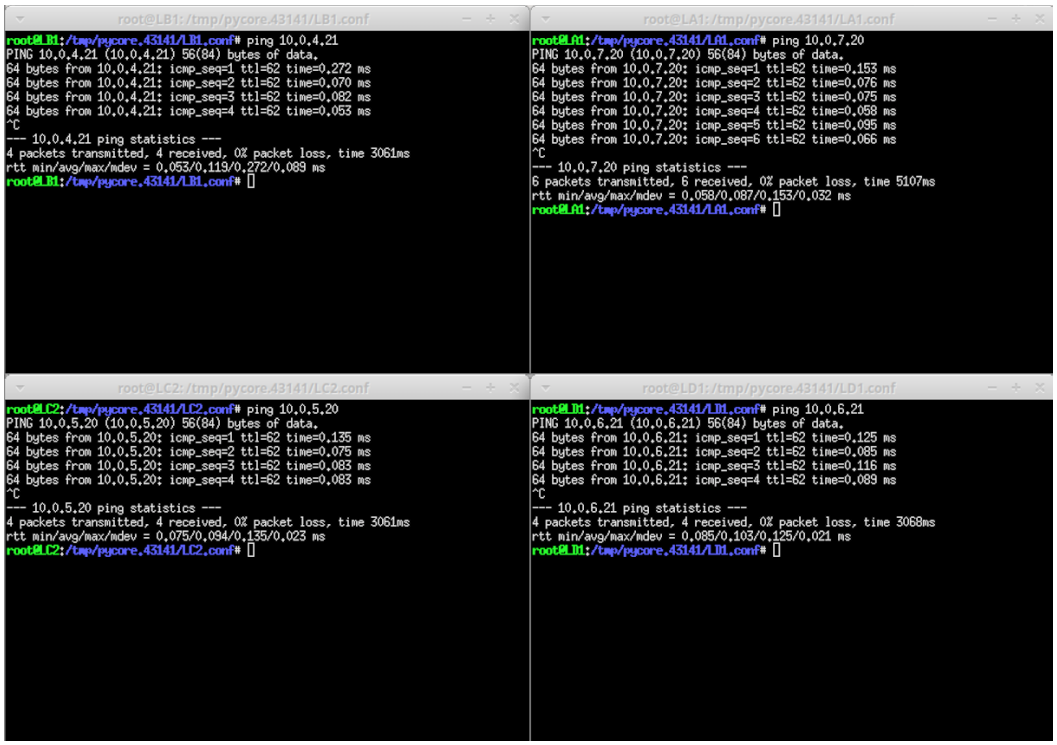

Figura 3. *Pings* feitos em cada departamento.

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

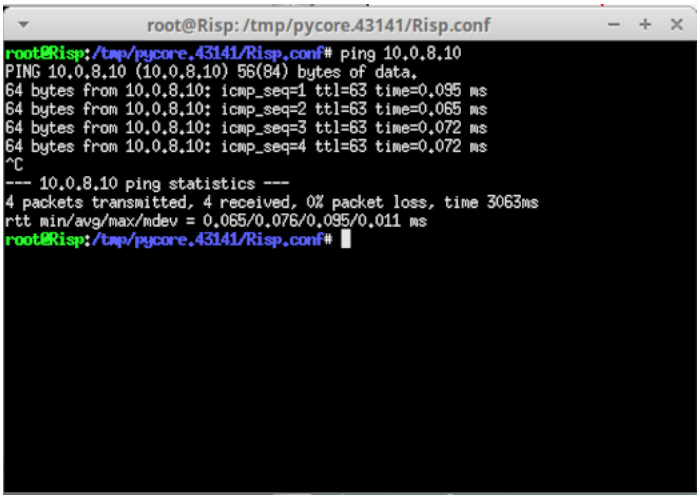

Figura 4. *Ping* feito para o servidor através do RISP.

2.2 Para o router e um laptop do departamento C:

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.**

Para um laptop do departamento C:

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
0.0.0.0	10.0.6.1	0.0.0.0	UG	0	0	0	eth0
10.0.6.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

– Nesta tabela podemos observar que a rota por defeito deste Laptop é o IP 10.0.6.1 (correspondente ao *RouterC*) e podemos observar também a linha que permite o endereçamento local.

Para o router do departamento C:

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
10.0.0.0	10.0.2.1	255.255.255.0	UG	0	0	0	eth0
10.0.1.0	10.0.3.2	255.255.255.0	UG	0	0	0	eth1
10.0.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
10.0.3.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
10.0.4.0	10.0.2.1	255.255.255.0	UG	0	0	0	eth0
10.0.5.0	10.0.2.1	255.255.255.0	UG	0	0	0	eth0
10.0.6.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
10.0.7.0	10.0.3.2	255.255.255.0	UG	0	0	0	eth1
10.0.8.0	10.0.2.1	255.255.255.0	UG	0	0	0	eth0
10.0.9.0	10.0.2.1	255.255.255.0	UG	0	0	0	eth0

– Nesta tabela podemos observar o encaminhamento feito pelo router do departamento C para os restantes equipamentos da topologia CORE. Vemos que, como esperado, as gamas de IPs locais ao router tem o próximo nó não especificado (o gateway corresponde ao ip 0.0.0.0) e que apenas mudam as interfaces *ethernet*. Da mesma forma, os restantes IPs da topologia encontram-se encaminhados para os routers vizinhos ao *RouterC*. Note-se que não existe rota por defeito e, por isso, esta tabela não tem a capacidade de encaminhar IPs desconhecidos a esta organização.

b) **Diga, justificando, se está a ser usado encaminhamento estático ou dinâmico.**

– Ao analisarmos os processos a correr nos routers da rede podemos observar que cada um deles possui um processo `/usr/sbin/ospfd -d` que representa uma *daemon* do protocolo OSPF. Ora sendo o OSPF um protocolo de encaminhamento dinâmico, podemos concluir que está a ser utilizado encaminhamento dinâmico.

```
root@Ra:/tmp/pycore.43141/Ra.conf# ps -ax

  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:00 /usr/local/bin/vnoded -v -c /tmp/pycore.43141/Ra -l
/tmp/pycore.43141/Ra.log -p /tmp/pycore.43141/Ra.pid -C /tmp/pycore.43141/Ra.conf
   72 ?        Ss     0:00 /usr/sbin/zebra -d
   78 ?        Ss     0:00 /usr/sbin/ospf6d -d
   82 ?        Ss     0:00 /usr/sbin/ospfd -d
   90 pts/3    Ss     0:00 /bin/bash
  102 pts/3    R+     0:00 ps -ax
```

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.**

- Ao remover a rota por defeito do servidor, este fica deixa de ser capaz de enviar mensagens para todos os IPs que não conhece localmente. Ora tendo em conta que todas as interfaces desta organização são endereços estrangeiros à rede do servidor, podemos concluir que o servidor não conseguirá comunicar com nenhuma interface do Sistema MIEI-RC.

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.**

```
route add -net 10.0.0.0 gw 10.0.8.1 netmask 255.255.248.0
```

- A partir deste comando, devido a mascara de rede usada (/21) foi possível restaurar conectivade a todos os endereços no intervalo 10.0.0.0 até 10.0.7.255. Estes endereços englobam todas as interfaces e routers de todos os departamentos, com a exceção do router *Risp* e da subrede que o interliga ao *routerA*.

```
route add -net 10.0.9.0 gw 10.0.8.1 netmask 255.255.255.0
```

- A partir deste comando, devido a mascara de rede usada (/25) foi possível restaurar conectivade a todos os endereços no intervalo 10.0.9.0 até 10.0.9.255. Estes endereços correspondem à subrede que contém o router *Risp*, e sendo assim permite reestabelecer as comunicações com o mesmo.

e) **Teste a nova política de encaminhamento garantindo que o servidor está novamente acessível, utilizando para o efeito o comando ping.**

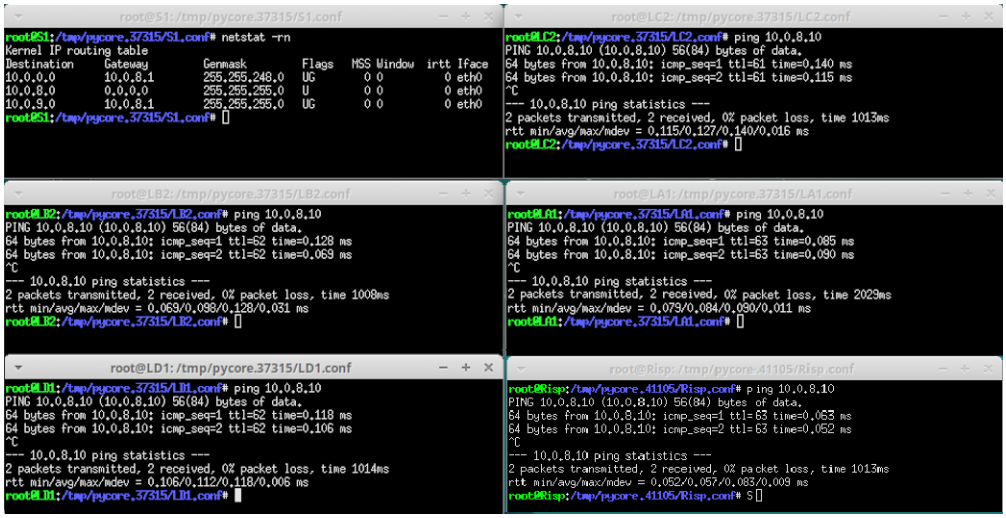

Figura 5. *Pings* feitos ao servidor.

3 Definição de sub-redes

a) **Considere que dispõe apenas do endereço de rede IP 130.67.96.0/19. 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.**

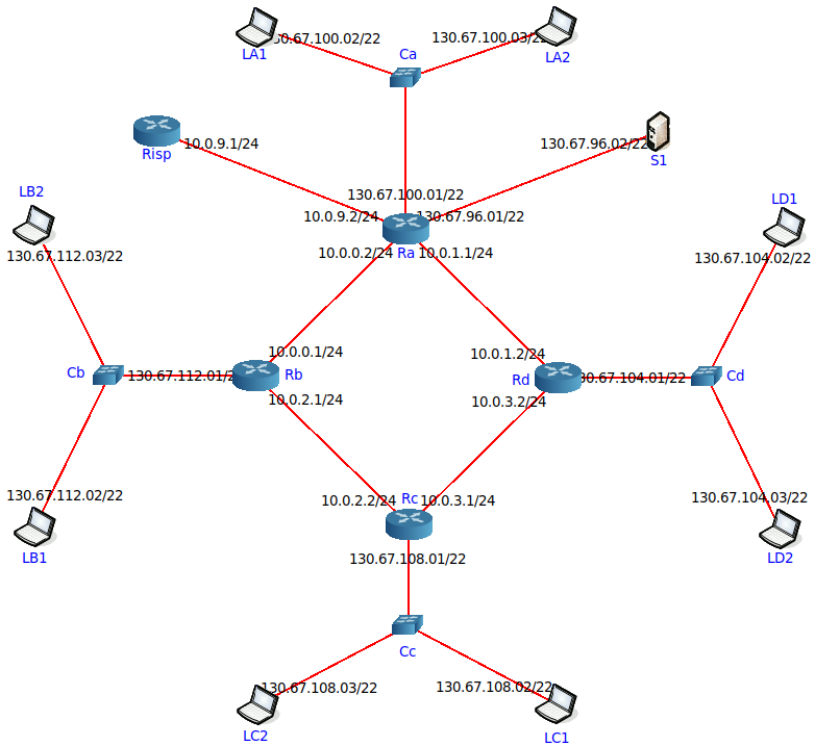

Figura 6. Esquema de Subneting

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

- Foi utilizada a máscara com valor decimal 22. Como existem 3 bits para definir as sub-redes, restam mais 10 bits para os endereços em si. Assim, cada sub-rede terá 1024 interfaces disponíveis (por exemplo: o departamento A terá 2048 interfaces pois existem duas sub-redes neste local, e os restantes departamentos terão 1024 interfaces em cada). Foram utilizadas 5 sub-redes das 8 permitidas por esta máscara.

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

- Mais uma vez, foi utilizado o comando *ping* para verificar as conexões de todos os computadores em cada departamento. Desta maneira, foi verificado que, de facto, esta solução é viável e que existe conectividade entre todos os locais. Na imagem seguinte demonstramos a conectividade do departamento C com o servidor e com os restantes departamentos.

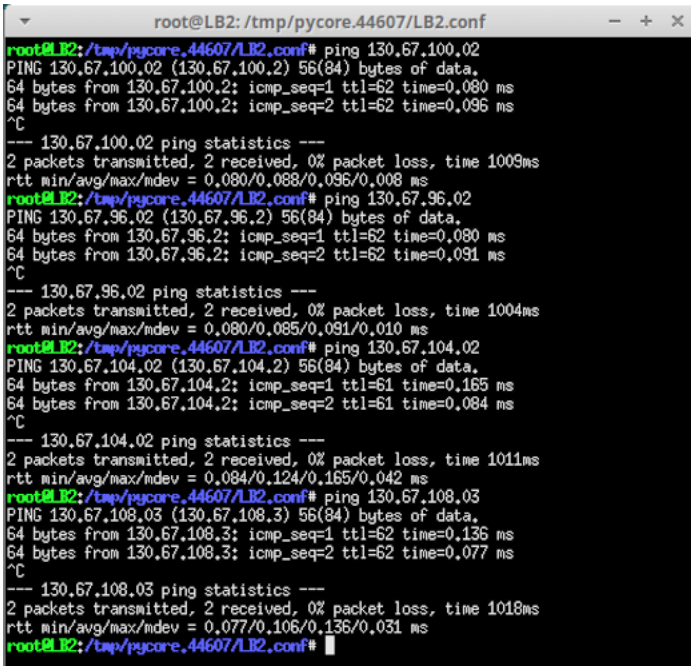

Figura 7. *Pings* feitos aos diversos departamentos a partir do *Laptop 2* do departamento *B*.

4 Conclusão

Em suma, este relatório apresenta as nossas respostas e estratégias para realizar as duas partes deste trabalho prático sobre Protocolo IP.

A primeira parte, relativamente aos *Datagramas IP e Fragmentação*, permitiu introduzir-nos aos conceitos básicos de um datagrama IP. Já na segunda parte do trabalho, fomos apresentados com noções de *Endereçamento e Encaminhamento IP*, onde foram propostas diversas técnicas para aumentar a escalabilidade do protocolo IP.

Para este efeito, foram utilizadas as ferramentas didáticas propostas pela equipa docente. Estas ferramentas concederam-nos uma maneira simples e intuitiva de captar o tráfego de pacotes em diversos tipos de redes, tal como analisar a estrutura de cada pacote.

Numa fase posterior, também nos permitiram definir endereços de sub-rede, de modo a ser possível analisar o encaminhamento do tráfego de dados entre diferentes tipos de sub-redes.

Deste modo, utilizando as ferramentas propostas para a realização deste trabalho, acreditamos que foi uma mais valia para o nosso estudo e aprofundamento destes conceitos, e que tornará os conceitos nos futuros projetos práticos mais intuitivos.