Trabalho Prático 2

José Pedro Silva, José Ricardo Cunha, and Válter Carvalho

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a84577,a84302,a84464}@alunos.uminho.pt

Resumo É explicada a resolução dos exercícios de ambas as partes do trabalho prático. Para a primeira, é estudado o protocolo IPv4, através da análise de datagramas e de fragmentações dos mesmos, usando o WireShark, *bash* de Linux e o *CORE*. Na segunda parte, é explorado o endereçamento, sub-redes, máscaras de rede e endereços privados, usando o *CORE* e a *bash* de Linux, novamente.

1 Introdução

Com este trabalho pretendemos mostrar a nossa interpretação dos problemas apresentados, assim como tentar explicitar o nosso raciocínio lógico, de modo a conseguir uma melhor visão da tecnologia em Redes, mergulhando nas ideias intrínsecas por base neste ramo científico.

Tentamos de forma concisa agrupar as ideias e definições num pequeno relatório que apenas contém comentários gerais (mas pertinentes) às vàrias questões apresentadas, tentando detalhar ao máximo os conceitos mais relevantes para cada problema.

Segundo o enumeramento lógico do trabalho prático, este foi dividido em duas partes, ambas realizadas durante as diversas aulas práticas durante o semestre.

Na **Parte 1** (secção 2.1), foi proposta a análise de datagramas, nomeadamente de protocolos ICMP e IPv4, com o objetivo de perceber como estão organizados as estruturas destes pacotes, assim como perceber o propósito das mesmas. Primeiramente, começou-se por analisar um datagrama singular, de seguida, foi analisado o processo de fragmentação.

Na **Parte 2** (secção 2.2), foi proposta a análise e interpretação de sub-redes, assim como modos de endereçamento. Inicialmente foi-nos exposta uma tipologia para analisar e foi-nos pedido para fazer alterações a essa topologia, nomeadamente às tabelas de reencaminhamento, de modo a refletirmos sobre a sua importância no reencaminhamento de pacotes.

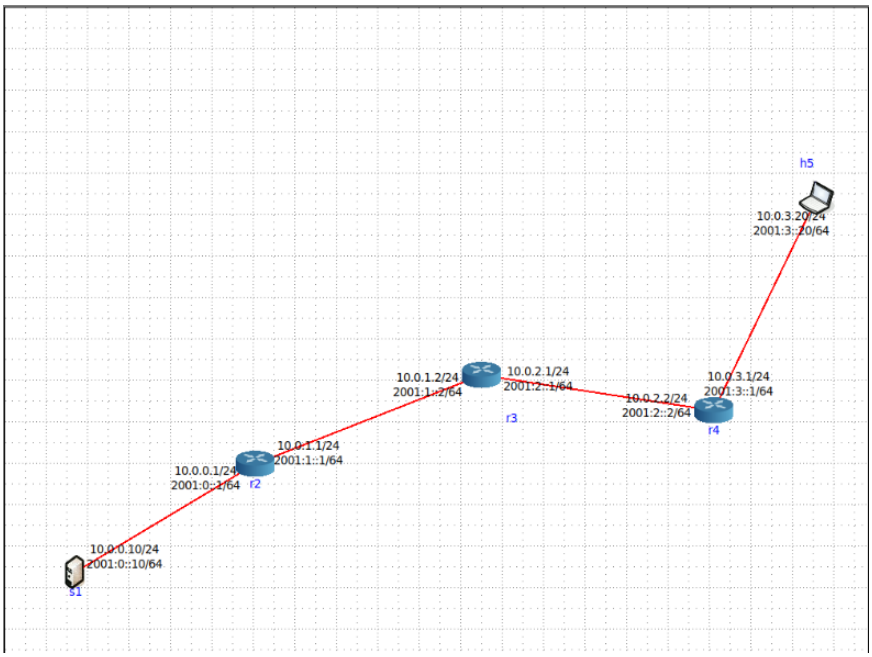
Ambas as partes são extremamente importantes porque fazem ponte com as aulas teóricas, isto é, servem para cimentar conhecimentos já adquiridos e que, certamente, são importantes na formação de futuros engenheiros, não só para este ramo em concreto mas sim como conhecimentos transversais.

2 Questões e Respostas

2.1 Parte 1

Exercício 1 *Prepare uma topologia no CORE para verificar o comportamento do traceroute. Ligue um host (servidor) s1 a um router r2; o router r2 a um router r3, o router r3 a um router r4, que por sua vez, se liga a um host (pc) h5. Ajuste o nome dos equipamentos atribuídos por defeito para a topologia do enunciado.*

Será este o modelo proposto pelo enunciado.



Esquema da topologia no CORE.

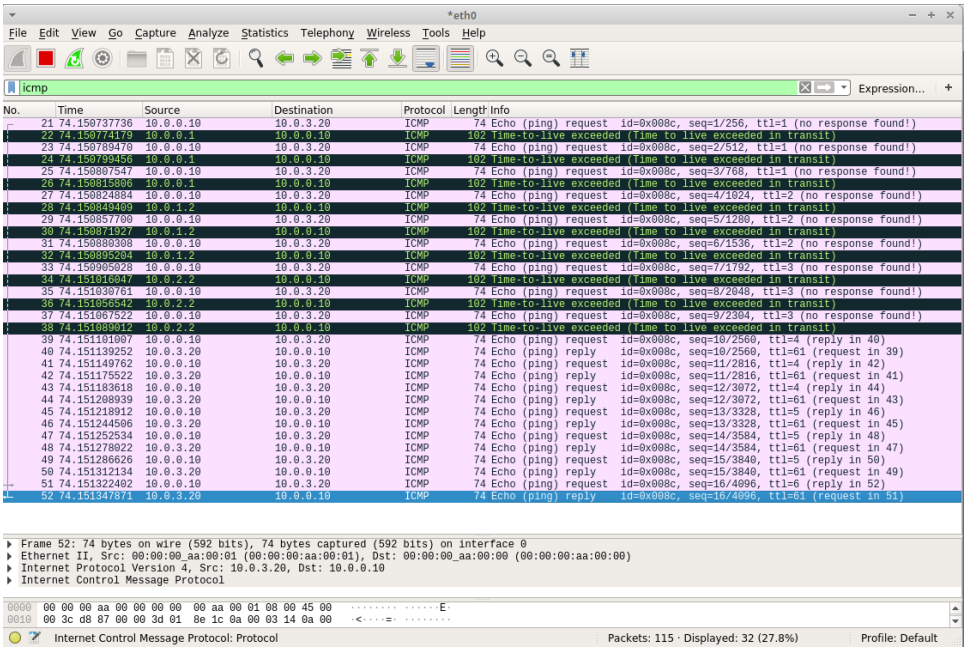
– **a)** *Active o wireshark ou o tcpdump no pc s1. Numa shell de s1, execute o comando traceroute -I para o endereço IP do host h5.*

```
root@s1:/tmp/pycore.45921/s1.conf# traceroute -I 10.0.3.20
traceroute to 10.0.3.20 (10.0.3.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.040 ms  0.011 ms  0.009 ms
 2  10.0.1.2 (10.0.1.2)  0.020 ms  0.013 ms  0.013 ms
 3  10.0.2.2 (10.0.2.2)  0.025 ms  0.017 ms  0.017 ms
 4  10.0.3.20 (10.0.3.20)  0.050 ms  0.021 ms  0.021 ms
```

Execução do comando traceroute.

Deixando o WireShark aberto em plano de fundo, executamos este comando, de modo a servir de apoio nos exercícios que se seguem.

– **b)** *Registe e analise o tráfego ICMP enviado por s1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.*

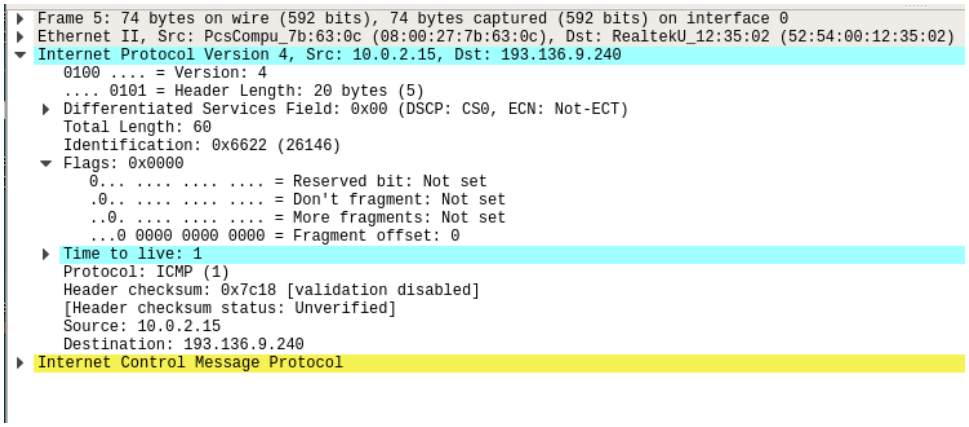


Análise do tráfego em Wireshark.

Verificamos que acontece o esperado, que é ser TTL=4 o número mínimo de saltos para conseguir estabelecer conexão. Primeiro, são enviados datagramas com TTL=1, descartados por r2. Seguidamente, são enviados com TTL=2, que por sua vez são descartados por r3. Novamente, são enviados datagramas com TTL=3, desta vez sendo descartados por r4. Só quando é enviado pelo servidor outros pacotes com TTL=4, é que estes são recebido pelo computador h5. Este resultado é evidente no wireshark, quando se verifica que para TTL<4, estes resultam em "*time-to-live exceeded*",ou seja, só quando TTL=4 é que há um *reply* por parte de h5.

– **c)** *Qual deve ser o valor inicial mínimo do campo TTL para alcançar o destino h5? Verifique na prática que a sua resposta está correta.*
Como visto anteriormente, TTL=4. Vemos pelo *WireShark* que, de facto, se verifica esta resposta.
– **d)** *Qual o valor médio do tempo de ida-e-volta (Round-Trip Time) obtido?*
RTT = (0.05 + 0.021 + 0.021) / 3 = 0.03 ms.
Vemos estes valores pelo *hop* número 4 em **a)**.

Exercício 2 *Execute o seguinte comando na sua máquina nativa e analise o tráfego ICMP: traceroute -I marco.uminho.pt.*



Análise do tráfego em Wireshark do primeiro pacote ICMP.

- **a)** *Qual é o endereço IP da interface ativa do seu computador?*
  O IP da interface ativa do computador é 10.0.2.15.

- **b)** *Qual é o valor do campo protocolo? O que identifica?*
  O protocolo é ICMP (1). Identifica o protocolo de Internet que se refere a algumas situações de tratamentos de erros, como por exemplo, quando um pacote IP não consegue chegar ao destino pretendido.

- **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?*
  Como temos Length = 60 bytes e Header = 20 bytes, então: Payload = 60-20 = 40 bytes, porque o *payload* é dado pela diferença entre dois campos, na sua definição.

- **d)** *O datagrama IP foi fragmentado? Justifique.*
  Não. Dentro do *header*, tem a seguinte flag: *More fragments: Not set*, o que indica a não existência de fragmentos para além do atual. Para além disso, a flag: *Fragment offset: 0* indica que está a apontar para o início do datagrama original, ou seja, não há mais nenhum fragmento sem ser o pacote original, que é o próprio.

- **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.*
  Os únicos campos que mudam são: Identification, Header Checksum, TTL. Isto verifica-se porque o TTL incrementa sempre a cada iteração, o checksum é um controlador de erros, é natural que mude e a identificação muda porque cada pacote é diferente, logo, tem de ser identificado diferenciadamente.

- **f)** *Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?*
  Sim. Ambos os campos são incrementados em cada datagrama, o que segue a lógica de **e)**.

– **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?*

  Sim. O valor permanece constantemente 61 para mensagens de *Time Exceeded*. Como, no *host*, o valor máximo de TTL está definido como 64 por defeito, para se garantir que o datagrama chegue ao destino e como há 3 routers intermédios, o TTL é decrementado por 3, ou seja, fica um TTL final de 61.

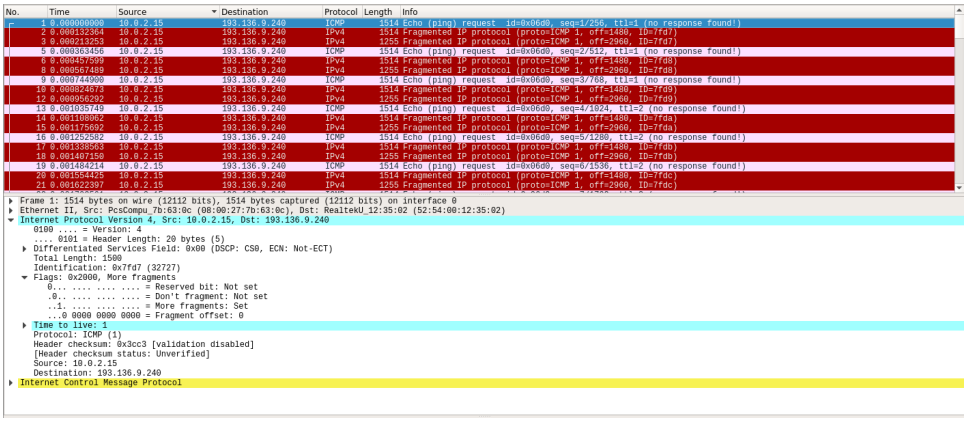
Exercício 3 *Pretende-se agora analisar a fragmentação de pacotes IP. Reponha a ordem do tráfego capturado usando a coluna do tempo de captura. Observe o tráfego depois do tamanho de pacote ter sido definido para 42yy bytes (yy é o número do grupo).*

O nosso grupo é o **01**, pelo que teremos de usar 4201 bytes.

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

  Estamos a tentar enviar 4201 bytes de uma só vez num datagrama. Como a PDU (*Protocol Data Unit*) máxima é 1500 bytes, o pacote terá de ser fragmentado em 3 distintos, de modo a cumprir esta norma.

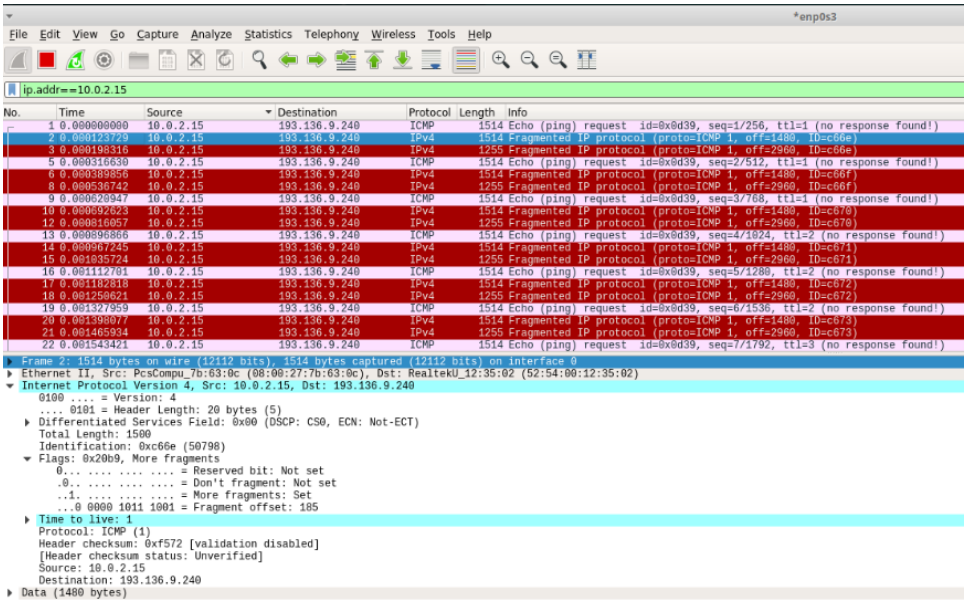
– **b)** *Imprima o primeiro fragmento do datagrama IP segmentado. Que informação no cabeçalho indica que o datagrama foi fragmentado? Que informação no cabeçalho IP indica que se trata do primeiro fragmento? Qual é o tamanho deste datagrama IP?*



Análise do tráfego em Wireshark (fragmento 1).

Vemos pela imagem que nas *flags* do datagrama original temos os seguintes campos: *More Fragments: Set*, que indica que existe fragmentação do datagrama original, *Fragment Offset: 0*, que indica que se trata do primeiro fragmento dos vários que existem do mesmo pacote. Vemos, também, que o tamanho é 1500 (pelo campo *length* do *header*).

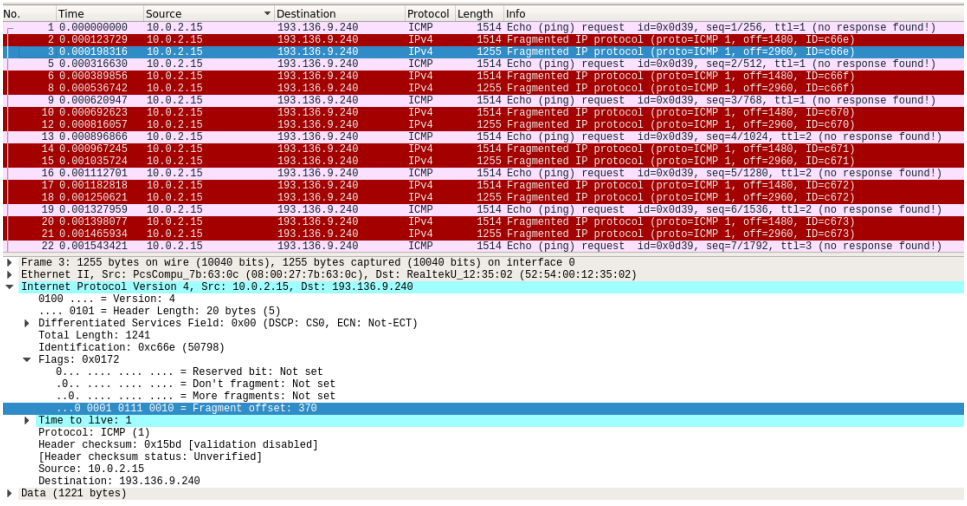
– **c)** *Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1º fragmento? Há mais fragmentos? O que nos permite afirmar isso?*



Análise do tráfego em Wireshark (fragmento 2).

Trata-se do segundo fragmento uma vez que o *Fragment Offset=185*, ou seja, há um deslocamento relativo ao primeiro fragmento (porque é diferente de 0), que coincide com o final do pacote inicial (primeiro fragmento). Porém, há ainda mais fragmentos, que é visível na flag *More Fragments: Set*.

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



Análise do tráfego em Wireshark (fragmento 3).

Foram criados 3 fragmentos distintos, com o mesmo campo de Identificação. Novamente, os campos das flags *Fragment Offset = 370* e *More Fragments: Not Set* indicam

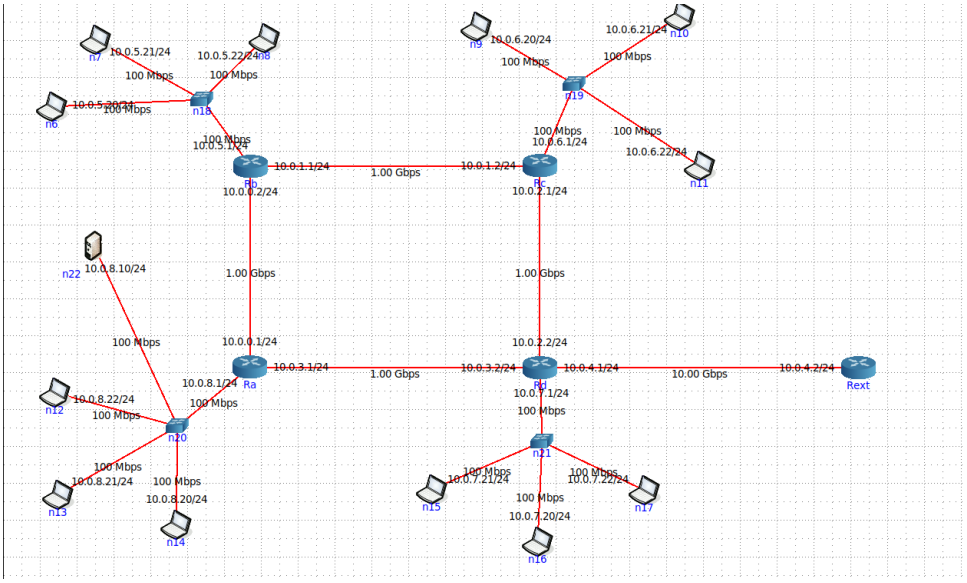
que este é o último fragmento (0 <= 185 <= 370, como está "Not Set"o campo de existirem mais fragmentos, significa que o último termina com *offset* de 370 relativamente ao original).

– **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 campos que mudam são a *Length* do datagrama e as respetivas *flags* relativas à fragmentação. Como a identificação não muda e como temos os *offsets* de cada fragmento, podemos reconstruir o datagrama colocando cada um na "posição"correta (porque sabemos a identificação do original), colocando-os ordenados e conseguindo os 4201 bytes originais.

2.2 Parte 2

Exercício 1 *Considere que a organização MIEI-RC é constituída por quatro departamentos (A, B, C e D) e cada departamento possui um router de acesso à sua rede local. Estes routers de acesso (Ra, Rb, Rc, e Rd) estão interligados entre si por ligações Ethernet a 1 Gbps, formando um anel. Por sua vez, existe um servidor (S1) na rede do departamento A e, pelo menos, três laptops por departamento, interligados ao router respetivo através de um comutador (switch). S1 tem uma ligação a 1 Gbps e os laptops ligações a 100 Mbps. Considere apenas a existência de um comutador por departamento. A conectividade IP externa da organização é assegurada através de um router de acesso Rext conectado a Rd por uma ligação ponto-a-ponto a 10 Gbps.*

– **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.*

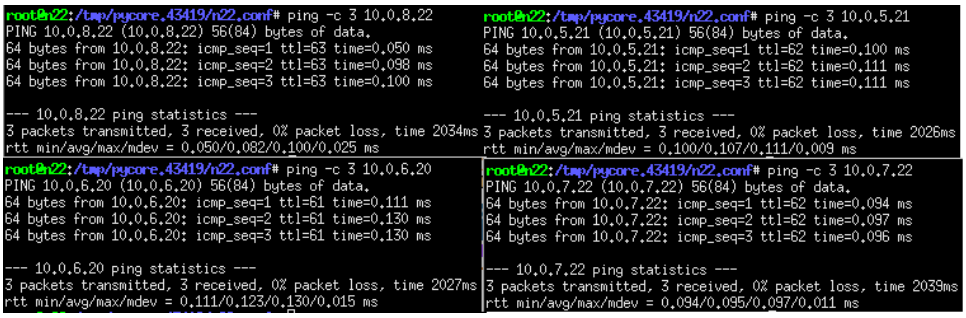


Apresentação dos vários IP e máscaras atribuídos.

– **b)** *Trata-se de endereços públicos ou privados? Porquê?*
São todos endereços privados, porque são todos do domínio 10.0.0.0/8, que fazem parte dos IP's de redes de Classe A definidos para redes privadas pela IANA.

– **c)** *Por que razão não é atribuído um endereço IP aos switches?*
Como estes trabalham num nível inferior ao dos *IP's* (nível 2 das camadas de Internet, isto é, "*Link Layer*"), não possuem IP.

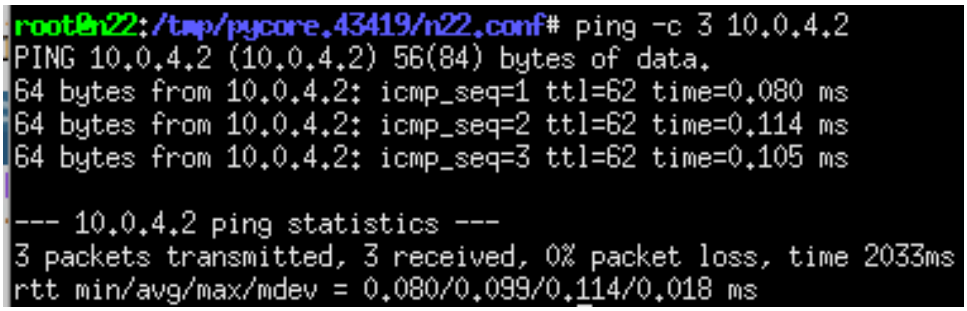
– **d)** *Usando o comando ping certifique-se que existe conectividade IP entre os laptops dos vários departamentos e o servidor do departamento A (basta certificar-se da conectividade de um laptop por departamento).*



Pings de 1 computador de vários departamentos.

Como podemos ver, de facto há conectividade entre os computadores de cada departamento e o servidor.

– **e)** *Verifique se existe conectividade IP do router de acesso Rext para o servidor S1.*
Utilizando o comando ping:

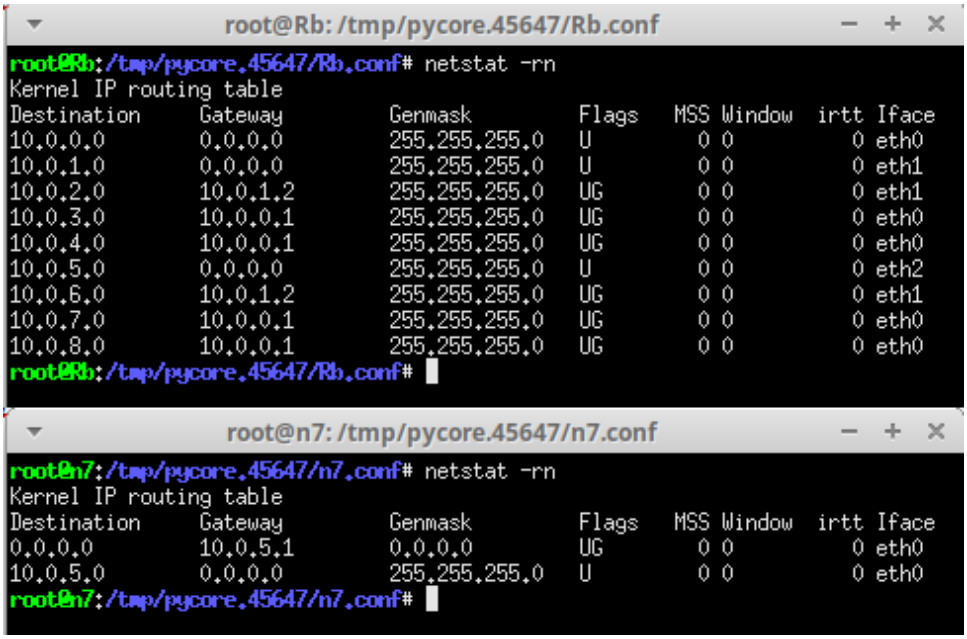


Ping do servidor para o router externo.

Verificamos que de facto a conexão é válida.

Exercício 2 *Para o router e um laptop do departamento B:*

– **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*



Tabelas de Reencaminhamento do Rb (1ª parte) e de um computador dentro do departamento B (2ª Parte)

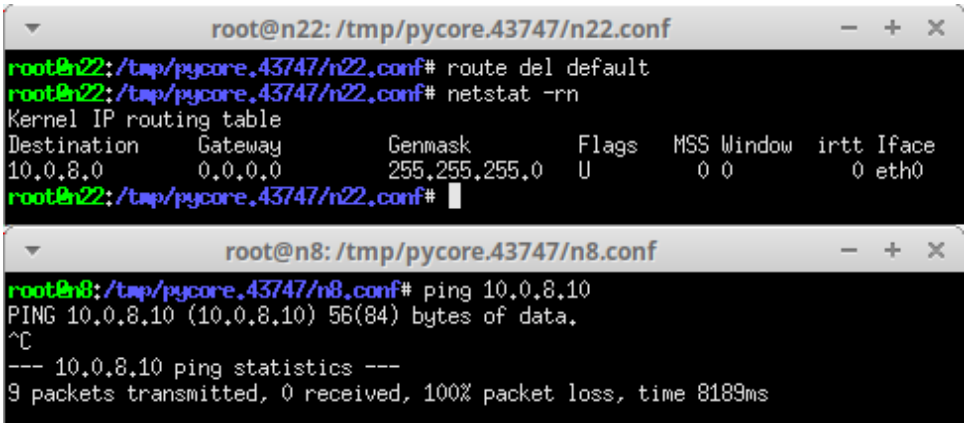
Começando na primeira linha e interpretando cada coluna: Um datagrama que tenha como origem uma rede do tipo 10.0.0.0, com máscara 255.255.255.0, ou seja, redes definidas pelo IP 10.0.0.0/24, com entrada na interface de rede dada por 0.0.0.0, partirá da interface eth0 do Rb. O proceso é análogo para as restantes linhas do router B.

Para o computador, o processo tem exatamente a mesma lógica: datagramas com destino 10.0.5.0/24, partirá da interface eth0 com entrada na *gateway* 0.0.0.0.

– **b)** *Diga, justificando, se está a ser usado encaminhamento estático ou dinâmico (sugestão: analise que processos estão a correr em cada sistema).*

Está a ser usado um reencaminhamento dinâmico. Ao analisarmos os diferentes processos vemos que há alguns processos com siglas *ospf*, que indicam protocolos de reencaminhamento dinâmico.

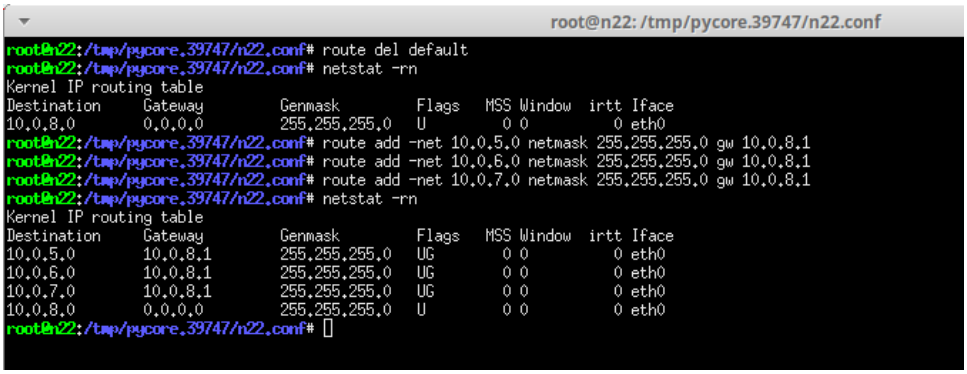
– **c)** *Admita que, por questões administrativas, a rota por defeito (0.0.0.0 ou default) deve ser retirada definitivamente da tabela de encaminhamento do servidor S1 localizado no departamento A. Use o comando route delete para o efeito. Que implicação tem esta medida para os utilizadores da empresa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



Ping de um computador de uma outra subrede para o servidor.

Todos os clientes (exceto os da sua rede local) perderão a conexão ao servidor, como visível pelo ping acima. Como não há rota por defeito, o servidor pede a capacidade de enviar pacotes para computadores fora da sua rede local. Os clientes conseguem transmitir pacotes, mas não recebem resposta do servidor, porque são imediatamente descartados todos os pacotes que não têm como destino a própria rede local (10.0.8.0/24).

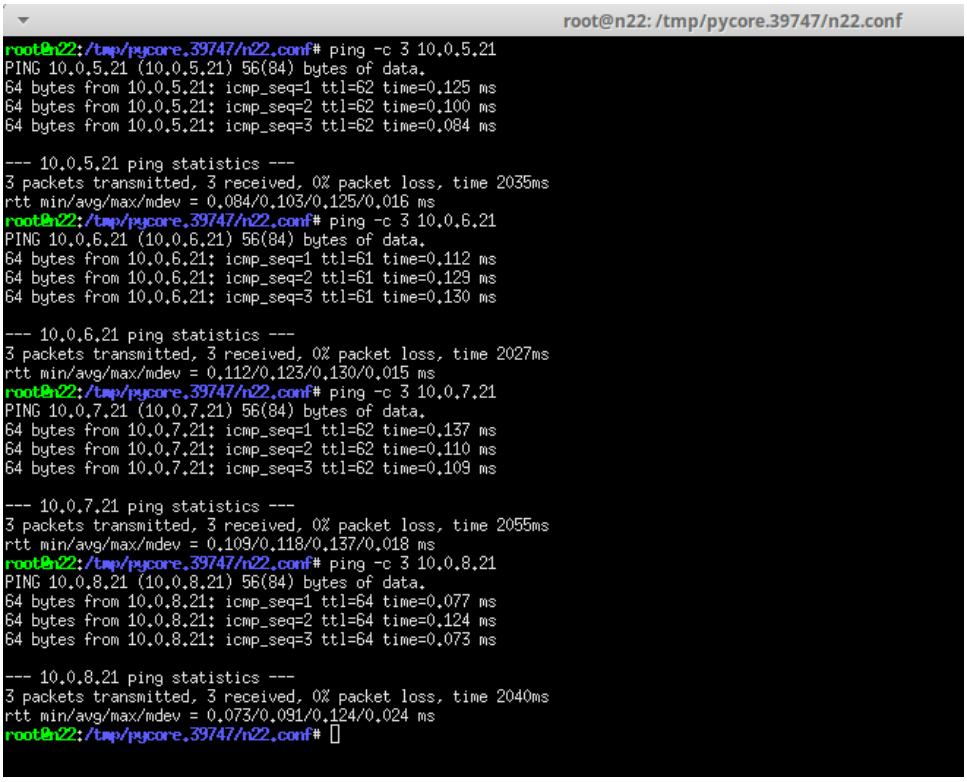
– **d)** *Adicione as rotas estáticas necessárias para restaurar a conectividade para o servidor S1 por forma a contornar a restrição imposta na alínea c). Utilize para o efeito o comando route add e registe os comandos que usou.*



Criação das novas entradas da tabela após eliminar a rota por defeito.

Para restituir a conexão, temos de indicar ao servidor como endereçar as 3 restantes subredes, isto é, dos routers Rb, Rc e Rd. Para tal, adicionamos à tabela os endereços das subredes, pelo que a conexão é reestabelecida. A *Destination* indica a subrede, a *Gateway* como vimos, indica por onde os pacotes têm de passar para chegar a cada subrede, e a *Genmask* será 255.255.255.0, que é a por defeito nesta topologia.

– **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.*



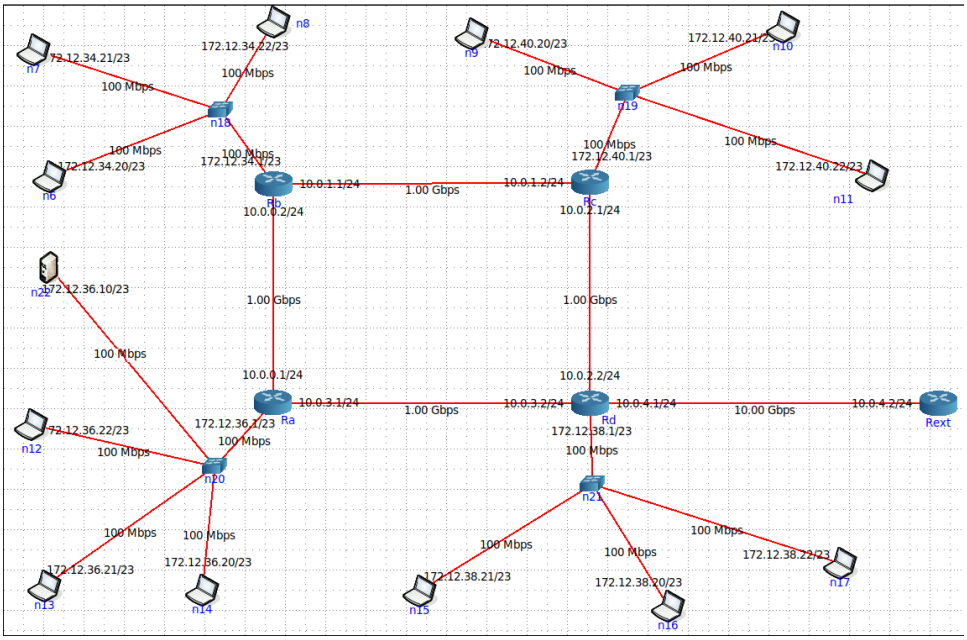
Ping do servidor para um computador em cada uma das subredes.

Vemos pelos pings que há conexão novamente entre os hosts e o servidor. A tabela de encaminhamento está visivel em **d)**, na imagem anexada respetiva, resultante do comando *netstat -rn*.

Exercício 3 *Considere a topologia definida anteriormente. Assuma que o endereçamento entre os routers se mantém inalterado, contudo, o endereçamento em cada departamento deve ser redefinido.*

- **a)** *Considere que dispõe apenas do endereço de rede IP 172.yyx.32.0/20 em que "yy" são os dígitos correspondendo ao seu número de grupo (Gyy) e "x" é o dígito correspondente ao seu turno prático (PLx). 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. Deve justificar as opções usadas.*

  O nosso grupo é o grupo **01** do turno **PL2**, logo, o campo *yyx* = 012 = **12**.



Novos endereçamentos das subredes.

Para determinar estes valores, primeiro começamos por analisar o IP dado.
Vemos que a máscara é 255.255.240.0 (pelo indicador de máscara no IP), pelo que em binário temos a seguinte sequência binária: 1111 1111 . 1111 1111 . 1111 0000 . 0000 0000, o que nos indica os bits que poderemos usar para endereçamentos para os departamentos e respetivos hosts, isto é, os bits contados apartir do primeiro bit nulo na sequência da máscara, são 12. Só precisamos de 3 destes bits para endereçar a subrede, uma vez que são 4 subredes no total ($2^3 - 2 = 6$, subtrai-se 2 porque não podem existir a combinação de todos os bits a 0 ou todos a 1 e obtemos assim o número de redes que podemos endereçar com 3 bits).
Tendo esta informação presente, seja a sequência **XXX** os 3 primeiros bits da máscara nulos. As subredes serão identificadas desta forma:

1. A: 010 ($2^2 = 4$)
2. B: 001 ($2^1 = 2$)
3. C: 100 ($2^3 = 8$)
4. D: 011 ($2^2 + 2^1 = 6$)

Que correspondem, respetivamente, aos octetos de rede 36,34,40 e 38 (somando a sequência anterior ao 32 do IP original). Temos, então, a presença duma nova máscara de rede: 255.255.254.0, porque estamos a utilizar aqueles 3 bits extra. Isto origina, então, os seguintes prefixos para as subredes:

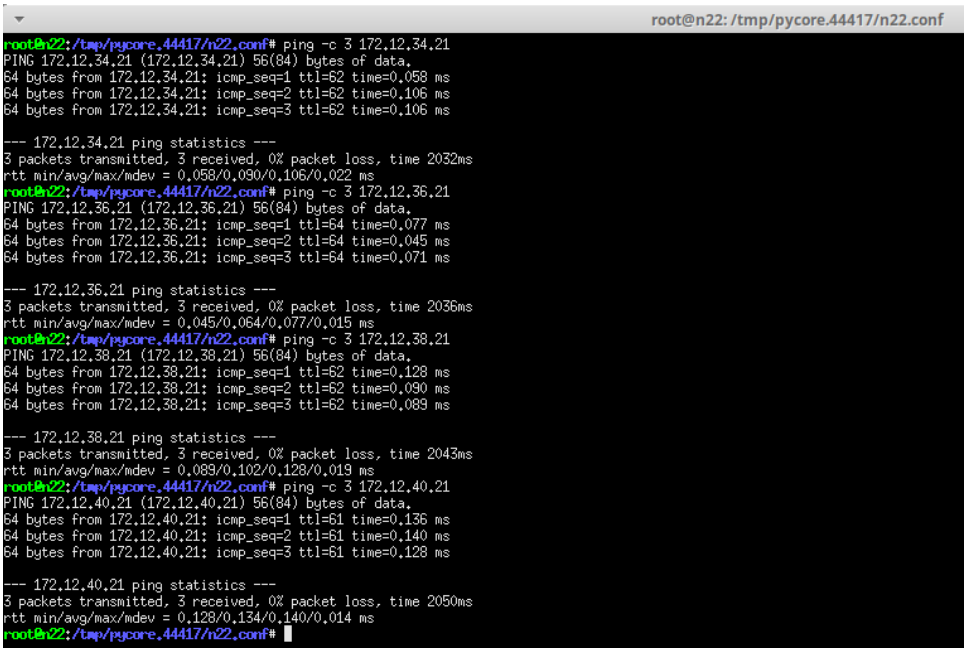
1. A: 172.12.36.0/23
2. B: 172.12.34.0/23
3. C: 172.12.40.0/23
4. D: 172.12.38.0/23

De seguida, mantemos a identificação por defeito das interfaces originais dos hosts, porque não é relevante mudar, obtendo assim o novo esquema de endereçamento visto na figura.

– **b)** *Qual a máscara de rede que usou (em notação decimal)? Quantos interfaces IP pode interligar em cada departamento? Justifique.*
Como visto em **a)**, a máscara de rede é **255.255.254.0**.

Como se verifica a existência de 9 bits para os hosts, podemos ligar a cada subrede $2^9 - 2 = \mathbf{510}$ interfaces distintas (o facto de serem subtraídas 2 unidades deve-se às combinações de todos os bits a 0 e todos os bits a 1 não serem válidas).

– **c)** *Garanta e verifique que a conectividade IP entre as várias redes locais da organização MIEI-RC é mantida. Explique como procedeu.*
Para testar a conectividade, enviamos um ping a uma interface de host de todas as subredes.



Pings para hosts das outras subredes.

Verificamos que, de facto, a conectividade IP é mantida.

3 Conclusões

No final de realizar este trabalho, num contexto geral, sentimos que conseguimos atingir os objetivos pretendidos, pelo que conseguimos por fim estabelecer uma ponte entre o que foi lecionado nas aulas teóricas e as aplicações práticas dos conceitos e temas propostos.

Sentimos que este projeto elucidou-nos bastante acerca do protocolo IPv4 e da noção de *subnetting* principalmente, fazendo com que nós estabelecêssemos bases em temas que previamente não estávamos familiarizados e, portanto, enriqueceu-nos intelectualmente.

Em particular, em cada parte destacaremos conclusões mais particulares.

3.1 Parte 1

O conceito mais importante retirado do exercício 1 é o TTL. Com este exercícios verificamos o que acontece na prática com este campo do IPv4, acerca da sua incrementação por cada iteração de modo a conseguir estabelecer uma rota dum host para outro, assim como os valores que deve tomar (na teoria) para este efeito.

Acerca do exercício 2, após vasculhar internamente um datagrama, apercebemo-nos da forma como devemos interpretar os seus vários campos e a maneira como estão organizados. Existem campos de correção de erros, campos a indicar o seu tamanho, locais que

indicam se há fragmentação ou não do datagrama original, enfim, imensos mais que contribuem para a estrutura lógica e facilidade de leitura, de modo a que seja facilmente estudado e, mais importante, facilmente reencaminhado por máquinas.

Por fim, no exercício 3, "abrimos"novamente o datagrama para descobrir que agora tinha sido fragmentado, porque há um limite estabelecido como norma global e, caso sejam enviados mais bytes do que esta norma suporta, o datagrama tem de ser fragmentado. Analisamos, assim, os vários campos referentes à fragmentação, que são os mesmos do exercício anterior, só que desta vez mostravam o evidente: que o datagrama foi, de facto, fragmentado e a quantidade de vezes que o foi. Por fim, percebemos como se reconstruía o datagrama original apartir dos seus fragmentos.

3.2 Parte 2

No exerício 1, aprendemos noções de endereços públicos e privados, em que concluimos que se retratavam endereços privados na topologia porque assim estava estabelecido pela IANA. Percebemos, também, que por causa da divisão lógica da Internet, um switch não pode nunca ter um endereço IP, porque está num nível inferior a este último. Conseguimos, também, perceber a noção de subredes com esta topologia que, analisando instruções de ping, verificamos que todas as interfaces estão conectadas e é possível estabelecer conexões entre elas, usando esta técnica de endereçamento.

Para o exercício 2, aprofundamos a noção de tabelas de reencaminhamento, analisando campo a campo e percebendo a forma como os datagramas sao redirecionados a cada iteração, isto é, como é que cada linha desta tabela se comportava na prática e qual o seu significado. Verificamos que após remover o endereçamento por defeito, não havia qualquer tipo de conexão para computadores fora da própria subrede do servidor, o que se traduzia na impossibilidade de resposta por parte deste para os clientes, mas que conseguia ler os datagramas normalmente que foram enviados por todos os hosts. Assim, aprendemos que se eliminássemos esta rota, tínhamos de adicionar manualmente os endereços de todas as outras subredes, para que se reestabelecesse as conexões iniciais.

Por fim, no exercício 3, aprendemos como funciona a criação de novos endereços para redes já existentes apartir dum IP base. Conseguimos perceber que o processo depende da Classe de rede que estamos a utilizar (máscaras) e o número de subredes que existem na nossa topologia, de modo a garantir o desperdício mínimo de bytes, que podem ser usados para anexar interfaces de hosts dentro destas redes. Através dum processo de atribuição de endereços, garantimos a existência de novas subredes perfeitamente funcionais e facilmente alteráveis (eventualmente).

Referências

1. Internetworking - Protocolo IP (Notas de Apoio das Aulas Teóricas)
2. traceroute: http://tools.ietf.org/html/rfc2151 (secção 3.4)
3. Internet Protocol (IP): http://tools.ietf.org/html/rfc791
4. Internet Message Control Protocol (ICMP): http://tools.ietf.org/html/rfc792