Universidade do Porto
FEUP Faculdade de Engenharia

2º Trabalho Laboratorial

Relatório Final

Redes de Computadores

Henrique Miguel Bastos Gonçalves up201608320
Tomás Viana Coutinho de Oliveira Figueiredo up201607932

Índice:

Introdução

O trabalho foi dividido em duas partes: o desenvolvimento de uma aplicação de download e a configuração de uma rede utilizando um *router* e um *switch*.

A aplicação de download, desenvolvida de acordo com o protocolo *FTP*, realiza uma ligação *TCP* através de *sockets* que fará download de um ficheiro. Para a configuração da rede, foram criadas duas *VLANs* dentro do *switch* às quais ligaram-se 3 *tuxs* e o *router*.

O relatório tem como objetivo a exposição e explicação do desenvolvimento destas duas partes e está estruturado da seguinte forma:

- **Parte 1: Development of a download application**
  Arquitetura e resultados da aplicação de download.

- **Parte 2: Configuration and Study of a Network**
  Análise de cada experiência.

- **Conclusões**
  Análise e reflexão da informação exposta.

Parte 1: Development of a download application

Como primeira parte deste projeto tivemos de desenvolver um programa em C que usasse o protocolo FTP de forma a conseguirmos fazer download de um ficheiro. Para isto, o programa recebe como argumento uma string do tipo: **ftp://[<user>:<password>@]<host>/<url-path>** e separa-a de forma a obter a informação necessária para conectar-se com um servidor e começar o download.

Arquitetura

Como dito anteriormente, o primeiro passo é a obtenção da informação necessária a partir do argumento inicial. Para isto, é criada uma estrutura ***Info*** que guardará toda essa informação e duas funções, ***parseArgument*** e ***getFilename***, que trataram de desmontar o argumento dado de forma a obter o ***username***, ***password***, ***server***, ***path*** e o ***filename*** a partir do último.

Após a obtenção da informação que precisamos, começamos por interagir com o servidor, no qual, após envio de quaisquer comandos, nos responde com um código composto por três dígitos. Para isto, é utilizada a função ***getAnswer*** que vai receber, guardar e interpretar os três dígitos de forma a saber como proceder. Apesar de todos os dígitos darem informação diferente, usamos apenas o significado do primeiro para sabermos como continuar. Essa interpretação é feita da seguinte forma: o primeiro dígito é um valor de 1 a 5, sendo a reposta positiva se for de 1 a 3 e negativa de 4 a 5. Caso o primeiro dígito seja 1, é feita uma nova leitura da resposta do servidor, caso seja 2 ou 3 o programa continua, caso seja 4 repete-se a última acção realizada e caso seja 5 o programa termina.

Assim, após ser aberto o ***socket*** por onde se fará a conexão entre o servidor e o cliente, são enviados 4 comandos. Primeiro, sendo necessário fazer login, manda-se o comando ***USER*** seguindo do utilizador e o

comando ***PASS*** seguido da password. Depois de o login ter sido bem sucedido, manda-se o comando ***PASV*** de forma a entrar em modo passivo, no qual o servidor retorna também uma nova porta necessária para abrir um novo ***socket*** que servirá para troca de dados. Após a abertura do novo ***socket***, é enviado o comando ***RETR*** seguido do nome do ficheiro começando assim a realizar o download com recurso à função ***makeFile.*** Terminado o download, são fechadas as duas conexões e o programa termina.

Resultados

Testamos o programa com diferentes tipos de ficheiros de diferentes tamanhos, no qual retornava sempre caso o ficheiro não existisse, acabando por ser sempre bem sucedido. Esta interação pode ser vista na figura 1 nos anexos.

Parte 2: Configuration and Study of a Network

Experiência 1 – Configure an IP Network

Esta experiência teve como objetivo a ligação do tux1 ao tux4 utilizando o *switch*.

1) What are thee ARP packets and what are they used for?

O ARP (*Address Resolution Protocol*) é um protocolo de comunicação que serve para descobrir o endereço físico das placas de rede das máquinas (o MAC) através do endereço IP fornecido. O protocolo contrário de através de um MAC descobrir o IP associado chama-se RARP(*Reverse Address Resolution Protocol*).

2) What are the MAC and IP addresses of ARP packets and why?

Após *ping* do tux1 para o tux4, o tux4 envia um pacote a perguntar qual é o tux com aquele IP, ou seja, a perguntar que endereço MAC tem o tux que está a tentar comunicar com ele. Esta pergunta vem na forma de um pacote ARP com o endereço IP e endereço MAC do tux4 (172.16.60.254 e 00:21:5a:c5:61:bb respetivamente) e com o endereço IP do tux target, ou seja, que se quer saber o MAC (172.16.60.1). Como não se sabe o MAC do tux target este está registado como 00:00:00:00:00:00. Exemplificado na figura 2 nos anexos.

De seguida, o tux1 responde a dizer que é ele que tem aquele IP enviando o seu endereço MAC.

No pacote de resposta presente na figura 3, o endereço IP e MAC da origem são o do tux1 (172.16.60.1 e 00:0f:fe:8c:af:71 respetivamente) e o endereço IP e MAC do destino são o do tux4 (172.16.60.254 e 00:21:5a:c5:61:bb).

3) What packets does the ping command generate?

Como referenciado na pergunta anterior o comando *ping* gera primeiramente pacotes ARP para obter os endereços MAC e de seguida gera pacotes ICMP (*Internet Control Message Protocol*).

4) What are MAC and IP addresses of the ping packets?

Quando damos *ping* ao tux4 a partir do tux1 os endereços (origem e destino) IP e MAC dos pacotes vão ser os destes tuxs. Endereço de rede: 172.16.60.0/24.

Pacote de pedido (figura 4 no anexo):

Endereço MAC da origem do pacote: 00:0f:fe:8c:af:71 (tux1).

Endereço MAC do destino do pacote: 00:21:5a:c5:61:bb (tux4).

Endereço IP da origem do pacote: 172.16.60.1 (tux1).
Endereço IP do destino do pacote: 172.16.60.254 (tux4).

Pacote de resposta (figura 5 no anexo):
Endereço MAC da origem do pacote: 00:21:5a:c5:61:bb (tux4).
Endereço MAC do destino do pacote: 00:0f:fe:8c:af:71 (tux1).
Endereço IP da origem do pacote: 172.16.60.254 (tux4).
Endereço IP do destino do pacote: 172.16.60.1 (tux1).

5) How to determine if a receiving Ethernet frame is ARO, IP, ICMP?

Inspecionando o Ethernet header de um pacote conseguimos determinar o tipo da trama. Caso o tipo tiver o valor 0x0800, significa que o tipo da trama é IP, depois conseguimos analisar o IP header. Se o IP header tiver o valor 1 isso significa que o tipo de protocolo é ICMP. No entanto, se o tipo tiver o valor 0x0806, significa que o tipo da trama é ARP. Deverão ser consultadas as figuras 6 e 7 no anexo.

6) How to determine the length of a receiving frame?

O comprimento de uma trama recetora pode ser consultado no **wireshark** como mostra a figura 8.

7) What is the loopback interface and why is it important?

A interface *loopback* é uma interface virtual da rede que permite ao computador receber respostas de si mesmo. É usada para testar se a placa de rede está configurada corretamente, como se observa na figura 9 no anexo.

Experiência 2 – Implement two virtual LANs in a switch

Nesta experiência foram criadas duas virtual LANs (vlan60 e vlan61, no nosso caso) às quais foram ligados os tux1 e tux4, e tux2, respetivamente.

1) How to configure vlany0?

Na régua 1, a porta T4 tem que estar ligada à porta do *switch* na régua 2. A porta T3, da régua 1, vai estar ligada à porta S0 do tux que se deseja estar ligado ao *switch*. Assim, para criar a *vlan,* tal como referenciado nos slides do guião, utilizam-se estes comandos no **GTKTerm** do tux escolhido:

- configure terminal
- vlan y0
- end

Depois deverá adicionar-se as portas dos tux1 e 4:

- configure terminal
- interface fastethernet 0/[nº da porta]
- switchport mode access
- switchport access vlan y0
- end

2) How many broadcast domains are there? How can you conclude it from the logs?

Existem dois domínios de transmissão, visto que o tux1 recebe resposta do tux4 quando faz *ping broadcast*, mas não do tux2 pois este não está ligado à vlan0. O tux2 não recebe resposta de ninguém quando faz *ping broadcast* porque se encontra isolado na vlan1.

Experiência 3 – Configure a Router in Linux

Nesta experiência o tux4 foi configurado de forma a simular um router, estabelecendo uma ligação entre as duas VLANs criadas anteriormente.

1) What routes are there in the tuxes? What are their meaning?

As rotas para as vlans associadas:

a. Tux1 tem uma rota para a vlan0 (172.16.y0.0) pela gateway 172.16.y0.1.
b. Tux4 tem uma rota para a vlan0 (172.16.y0.0) pela gateway 172.16.y0.254 e uma rota para a vlan1 (172.16.y1.0) pela gateway 172.16.y1.253.
c. Tux2 tem uma rota para a vlan1 (172.16.y1.0) pela gateway 172.16.y1.1.

As rotas que foram criadas durante a experiência:

a. Tux1 tem uma rota para a vlan1 (172.16.y1.0) pela gateway 172.16.y0.254.
b. Tux2 tem uma rota pata a vlan0 (172.16.y0.0) pela gateway 172.16.y1.253.

O destino das rotas é até onde o tux que está na origem da rota consegue chegar.

2) What information does an entry of the forwarding table contain?

Destination*:* o destino da rota.
Gateway: o ip do próximo ponto por onde passará a rota.
Netmask: usado para determinar o ID da rede a partir do endereço IP do destino. É dado depois do '/' no IP da rede que indica quantos bits(da esquerda para a direita, ou seja, os mais significativos) representam a subnet onde se encontra um certo host na rede.
Flags: dá-nos informações sobre a rota.
Metric: o custo de cada rota.

Ref: número de referências para esta rota (não usado no *kernel* do Linux).
Use: contador de pesquisas pela rota, dependendo do uso de -F ou -C isto vai ser o número de falhas da cache (-F) ou o número de sucessos (-C).
Interface: qual a placa de rede responsável pela *gateway* (eth0/eth1).

3) What ARP messages, and associated MAC addresses, are observed and why?

Quando um tux dá *ping* a outro e o tux que recebeu o *ping* não conhece o MAC *address* do que enviou o *ping*, pergunta qual o MAC *address* do tux com aquele IP. E faz isso enviando uma mensagem ARP.

Essa mensagem vai ter o MAC do tux de origem associado e 00:00:00:00:00:00 (mensagem enviada em modo de broadcast) pois ainda não sabe qual o tux de destino. De seguida, o tux de destino reponde uma mensagem ARP a dizer o seu MAC *address.*

Esta mensagem vai ter associado tanto o MAC *address* do tux de destino como o de origem.

4) What ICMP packets are observed and why?

São observados pacotes ICMP de *request* e *reply*, pois depois de serem adicionadas as rotas todos os tuxs se conseguem ver uns aos outros. Se não se conseguissem ver, seriam enviados os pacotes ICMP de *Host Unreachable*.

5) What are the IP and MAC addresses associated to ICMP packets and why?

Os endereços IP e MAC associados com os pacotes ICMP são os endereços IP e MAC dos tuxs de origem e destino. Por exemplo, quando se faz *ping* do tux1 para o tux4 (.253) os endereços de origem vão ser 172.16.60.1 (IP) e 00:0f:fe:8c:af:71 (MAC) e o de destino 172.16.61.253 (IP) e 00:21:5a:c5:61:bb (MAC).

Experiência 4 – Configure a Commercial Router and Implement NAT

O objetivo desta experiência foi configurar primeiramente um router comercial sem NAT ligando-o à rede do laboratório. De seguida foi configurar o router com NAT permitindo assim o acesso dos computadores da rede à internet.

1) How to configure a static route in a commercial router?

De forma a configurar o *router*, foi necessário ligar a porta T4, da régua 1, à porta do *router*, da régua 2. Relativamente à porta T3, da régua 1, esta vai estar ligada à porta S0 do TUX que se pretende que esteja ligado ao *router*. Quanto à criação da VLAN, invocam-se os seguintes comandos no **GTKTerm** do TUX escolhido:

- configure terminal
- ip route [ip rota de destino] [máscara] [ip gateway]
- exit

2) What are the paths followed by the packets in the experiments carried out and why?

No caso de a rota existir, os pacotes usam essa mesma rota. Caso contrário, os pacotes vão ao *router* (rota *default*), o *router* informa que o TUX 4 existe, e deverá ser enviado pelo mesmo.

3) How to configure NAT in a commercial router?

De forma a configurar o *router*, foi necessário configurar a interface interna no processo de NAT, que foi feito seguindo o guião fornecido para a dada experiência. A partir do **GTKTerm**, foram inseridos os comandos presentes na figura 10 presente nos anexos.

4) What does NAT do?

O NAT (*Network Address Translation*) tem como objetivo a conservação de endereços IP. Assim, permite que as redes IP privadas que usem endereços IP não registados se conectem à Internet ou uma rede pública. O NAT opera num *router*, onde conecta duas redes e traduz os endereços privados, na rede interna, para endereços legais, antes que os pacotes sejam encaminhados para outra rede. Adicionalmente, o NAT oferece também funções de segurança e é implementado em ambientes de acesso remoto.

Em suma, permite que os computadores de uma rede interna, como a que foi criada, tenham acesso ao exterior, sendo que, um único endereço IP é exigido para representar um grupo de computadores fora da sua própria rede.

Experiência 5 – DNS

Nesta experiência foi necessário configurar o DNS (*Domain Name System*) nos tuxs 1, 2 e 4. Um servidor de DNS, neste caso, **services.netlab.fe.up.pt**, contém uma base de dados dos endereços IP públicos e dos seus respetivos *hostnames*. É usado para traduzir os *hostnames* para os seus respetivos endereços de IP.

1) How to configure the DNS service at an host?

De forma a configurar o serviço DNS, é necessário mudar o ficheiro **resolv.conf** que se localiza em **/etc** no *host* tux. Esse ficheiro tem de conter a seguinte informação:

- search netlab.fe.up.pt
- nameserver 172.16.1.1

Onde **netlab.fe.up.pt** é o nome do servidor DNS e 172.16.1.1 o seu endereço de IP. Após esta experiência, é possível aceder à internet nos tuxs.

2) What packets are exchanged by DNS and what information is transported?

Em primeiro lugar, temos um pacote enviado do *Host* para o *Server* (linha 6) que contém o *hostname* desejado, pedindo o seu endereço de IP. Deverá ser consultada a figura 11.
O servidor responde (linha 7) com um pacote que contem o endereço IP do *hostname*.

Experiência 6 – TCP connections

O objetivo desta experiência foi observar o comportamento do protocolo TCP utilizando para isso a aplicação desenvolvida na primeira parte do trabalho.

1) How many TCP connections are opened by your ftp application?

A aplicação FTP abriu 2 conexões TCP, uma para mandar os comandos FTP ao servidor e receber as respostas e outra para receber os dados enviados pelo servidor e enviar as respostas do cliente.

2) In what connection is transported the FTP control information?

O controlo de informação é transportado na conexão TCP responsável pela troca de comandos.

3) What are the phases of a TCP connection?

Uma conexão TCP tem três fases: o estabelecimento da conexão, troca de dados e encerramento da conexão.

4) How does the ARQ TCP mechanism work? What are the relevant TCP fields? What relevant information can be observed in the logs?

O TCP (*Transmission Control Protocol*) utiliza o mecanismo ARQ (*Automatic Repeat Request*) com o método da janela deslizante. Este consiste no controlo de erros na transmissão de dados. Para tal, utiliza ***acknowledgment numbers***, que estão num campo das mensagens enviadas pelo recetor que indicam que a trama foi recebida corretamente, ***window size*** que indica a gama de pacotes que o emissor pode enviar e o ***sequence number***, o número do pacote a ser enviado.

5) How does the TCP congestion control mechanism work? What are the relevant fields? How did the throughput of the data connection evolve along the time? Is it according the TCP congestion control mechanism?

O mecanismo de controlo de congestão é feito quando o TCP mantém uma janela de congestão que consiste numa estimativa do número de octetos que a rede consegue encaminhar, não enviando mais octetos do que o mínimo da janela definida pelo recetor e pela janela de congestão.

Registamos que no início do primeiro download no tux1, a taxa de transferência aumentou, chegando esta taxa a um pico perto dos 7 segundos. Após o início do segundo download verificamos uma descida a pique seguida de uma subida a pique que estabilizou relativamente (apesar de ainda ter alguns picos) num nível mais baixo do que quando apenas o primeiro download estava a ser feito.

O fluxo de dados de conexão está de acordo com o mecanismo de controlo de congestão pois quando a rede estava mais congestionada tinha um bitrate menor.

6) Is the throughput of a TCP data connections disturbed by the appearance of a second TCP connection? How?

Com o aparecimento de uma segunda conexão TCP, a existência de uma transferência de dados em simultâneo pode levar a uma queda na taxa de transmissão, uma vez que a taxa de transferência é distribuída de igual forma para cada ligação.

Conclusões

Apesar das dificuldades encontradas no início do desenvolvimento da parte dois deste trabalho, a sua realização deu-nos uma noção mais clara sobre o funcionamento de redes, algo tão presente no nosso dia-a-dia, como também fomos capazes de mais facilmente consolidar os conhecimentos adquiridos nas aulas teóricas.

Podemos concluir que o trabalho foi realizado com sucesso uma vez que conseguimos alcançar todos os objetivos pretendidos.

Anexos

Imagens

```
tux61:~/Downloads/RCOM-master/proj2/src# ./exec ftp://anonymous:1@speedtest.tele
2.net/1GB.zip
anonymous
1
speedtest.tele2.net
1GB.zip
1GB.zip

220 (vsFTPd 2.3.5)
 > Connection made.
 > Logging in.
331 Please specify the password.
230 Login successful.
227 Entering Passive Mode (90,130,70,73,82,181)
.
150 Opening BINARY mode data connection for 1GB.zip (1073741824 bytes).
 > File download finish.
226 Transfer complete.
tux61:~/Downloads/RCOM-master/proj2/src#
```

Figura 1

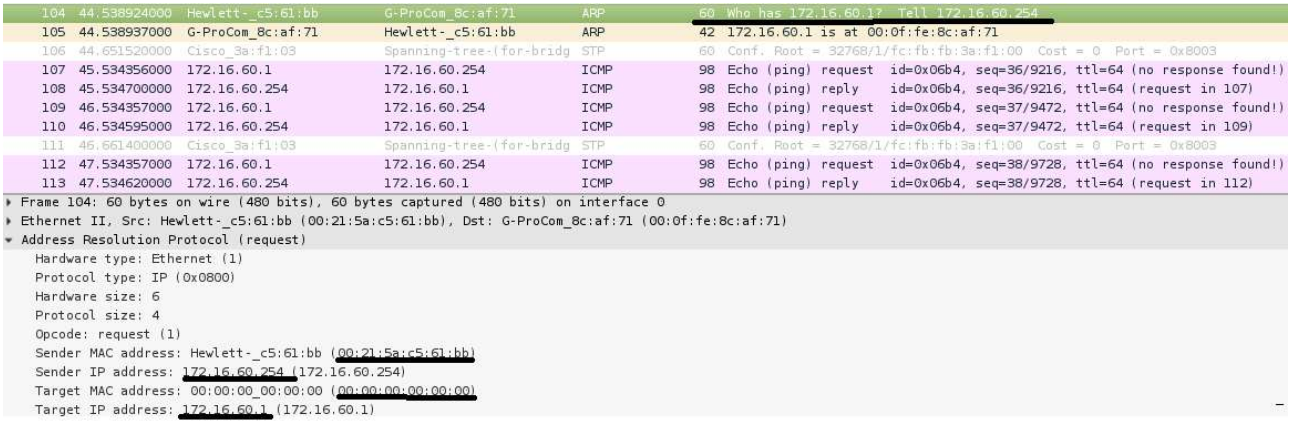

Figura 2

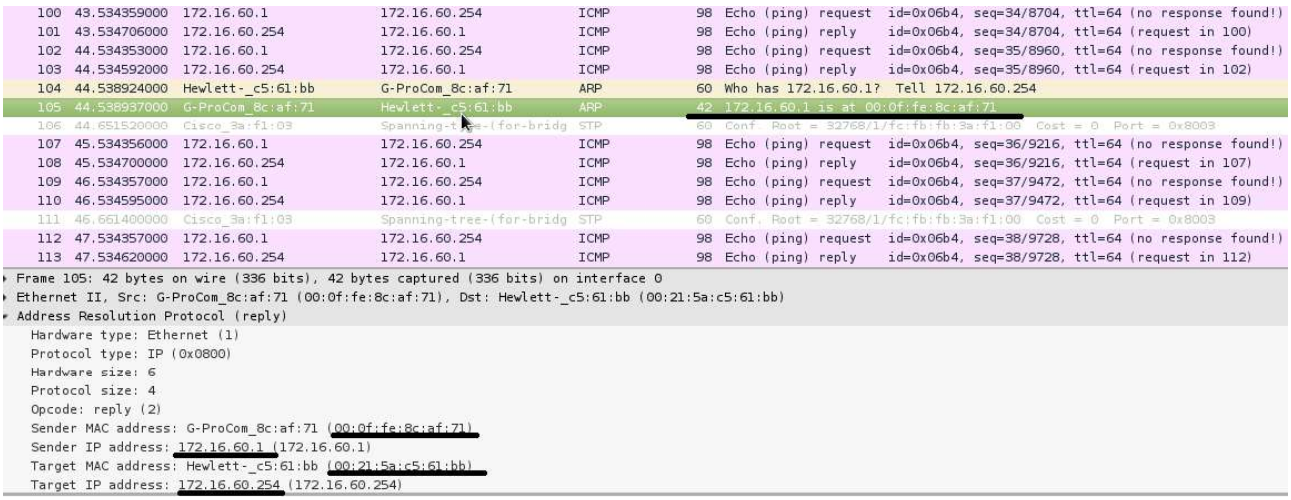

Figura 3

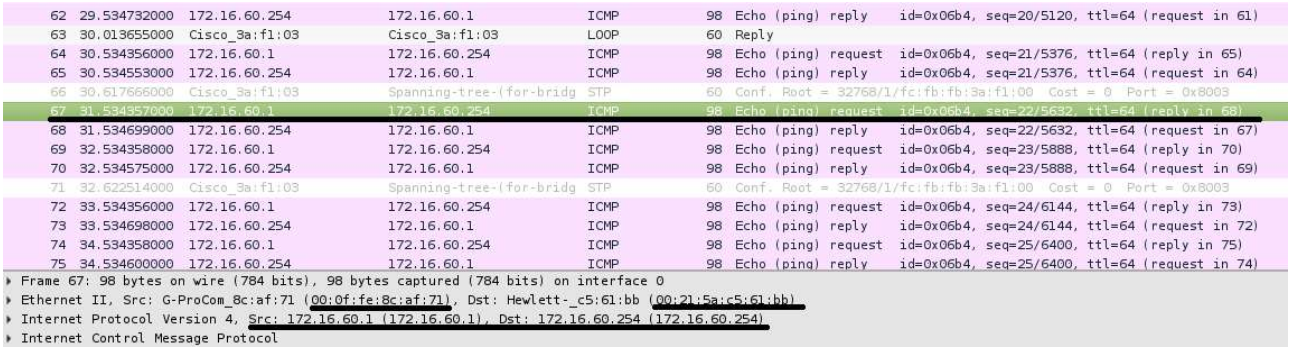

Figura 4

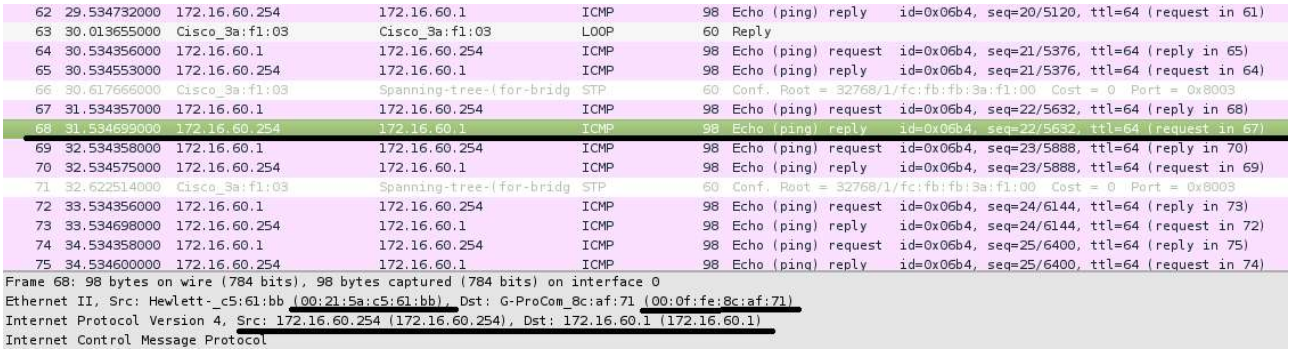

Figura 5

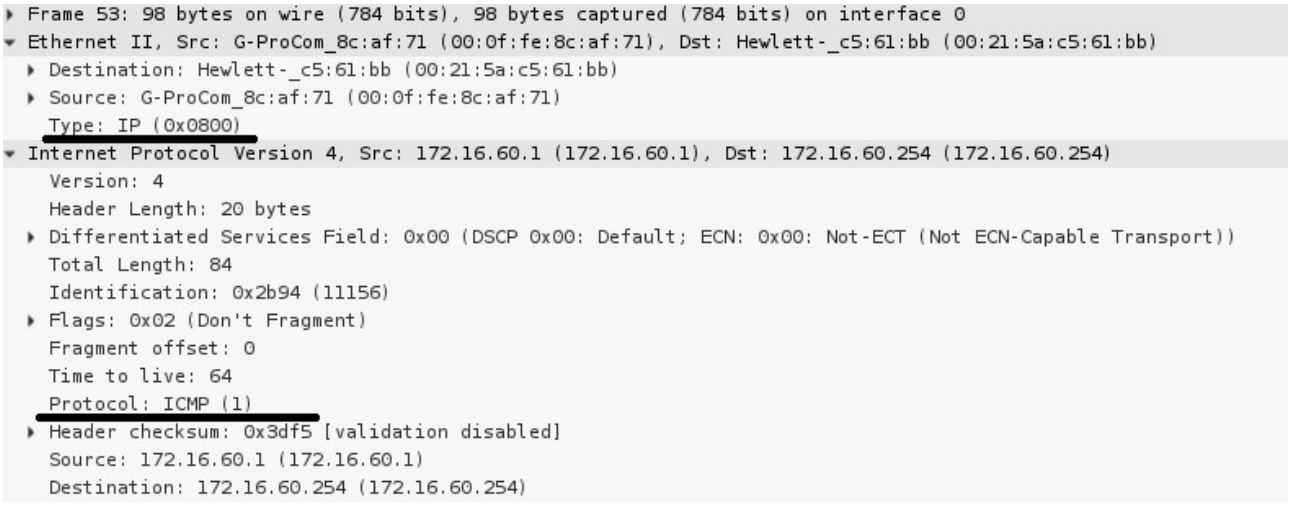

Figura 6

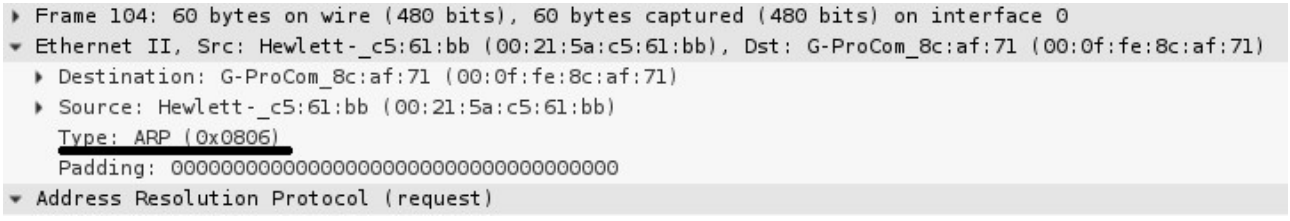

Figura 7

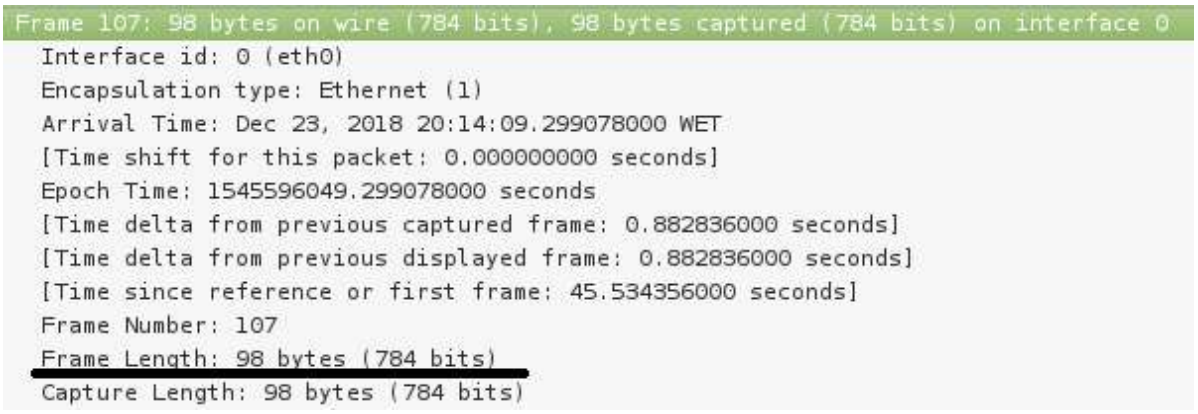

Figura 8

Figura 9

Configuração do Router Cisco com NAT

- Cisco NAT
  http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094e77.shtml

```
conf t
interface gigabitethernet 0/0 *
ip address 172.16.y1.254 255.255.255.0
no shutdown
ip nat inside
exit

interface gigabitethernet 0/1*
ip address 172.16.1.y9 255.255.255.0
no shutdown
ip nat outside
exit

ip nat pool ovrld 172.16.1.y9 172.16.1.y9 prefix 24
ip nat inside source list 1 pool ovrld overload

access-list 1 permit 172.16.y0.0 0.0.0.7
access-list 1 permit 172.16.y1.0 0.0.0.7

ip route 0.0.0.0 0.0.0.0 172.16.1.254
ip route 172.16.y0.0 255.255.255.0 172.16.y1.253
end
```

* In room I320 use interface fastethernet

46

Figura 10

4	5.939469000	172.16.60.1	172.16.1.1	DNS	70	Standard query 0x78e3 A google.com
5	5.942397000	172.16.1.1	172.16.60.1	DNS	334	Standard query response 0x78e3 A 172.217.168.174
6	5.942624000	172.16.60.1	172.217.168.174	ICMP	98	Echo (ping) request id=0x0fb5, seq=1/256, ttl=64 (reply in 7)
7	5.957892000	172.217.168.174	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0fb5, seq=1/256, ttl=50 (request in 6)
8	5.958047000	172.16.60.1	172.16.1.1	DNS	88	Standard query 0x9fad PTR 174.168.217.172.in-addr.arpa
9	5.959661000	172.16.1.1	172.16.60.1	DNS	385	Standard query response 0x9fad PTR mad07s10-in-f14.1e100.net

Figura 11