**Gonçalo Santos up202003537 1MEEC\_T02**

**Guilherme Moreira up201806631 1MEEC\_T02**

**26 de janeiro de 2022**

**Mestrado em Engenharia Eletrotécnica e de Computadores  
Redes de Computadores 2021/2022**

**2º Trabalho Laboratorial**

***Computer Networks***



***Índice***

[Introdução 2](#_Toc94523643)

[1. Parte 1 – Aplicação FTP 2](#_Toc94523644)

[2. Parte 2 – Configuração e Estudo de uma Rede 3](#_Toc94523645)

[2.1. Exp 1 – *Configure an IP Network* 3](#_Toc94523646)

[2.1.1. Arquitetura da rede 3](#_Toc94523647)

[2.1.2. Análise e discussão dos resultados 3](#_Toc94523648)

[2.2. Exp 2 – *Implement two virtual LANs in a switch* 4](#_Toc94523649)

[2.2.1. Arquitetura da rede 4](#_Toc94523650)

[2.2.2. Análise e discussão dos resultados 4](#_Toc94523651)

[2.3. Exp 3 – *Configure a Router in Linux* 5](#_Toc94523652)

[2.3.1. Arquitetura da rede 5](#_Toc94523653)

[2.3.2. Análise e discussão dos resultados 5](#_Toc94523654)

[2.4. Exp 4 – *Configure a Commercial Router and Implement NAT* 6](#_Toc94523655)

[2.4.1. Arquitetura da rede 6](#_Toc94523656)

[2.4.2. Análise e discussão dos resultados 7](#_Toc94523657)

[2.5. Exp 5 – DNS 7](#_Toc94523658)

[2.5.1. Arquitetura da rede 7](#_Toc94523659)

[2.5.2. Análise e discussão dos resultados 8](#_Toc94523660)

[2.6. Exp 6 – TCP *connections* 8](#_Toc94523661)

[2.6.1. Arquitetura da rede 8](#_Toc94523662)

[2.6.2. Análise e discussão dos resultados 9](#_Toc94523663)

[Conclusão 9](#_Toc94523664)

[Anexos 10](#_Toc94523665)

# Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Redes de Computadores visando implementar uma aplicação que recorresse ao FTP (*File Transfer Protocol*), tal como a realização de diversas experiências em ambiente laboratorial consistindo na configuração e estudo de uma rede de computadores.

Neste relatório iremos analisar em partes distintas as funcionalidades implementadas da aplicação, apelidada *download.c*, tal como os passos e resultados das experiências de modo a explicar como as realizamos.

# Parte 1 – Aplicação FTP

A aplicação FTP faz a transferência de um ficheiro, adota o protocolo FTP como descrito no RFC959 e recebe como argumento o endereço do ficheiro com a sintaxe *ftp://[<user>:<password>@]<host>/<url-path>* como descrito no RFC1738.

Inicialmente, é feita a análise do endereço passado na linha de comandos pelo utilizador invocando a função *url\_parsing()* de modo a retirar o *host,* o diretório do ficheiroe o nome do ficheiro. Caso, sejam indicados um utilizador e uma palavra-passe na linha de comandos, estes também são retirados. Também é convertido o *host* através da função *gethostbyname()* obtendo assim uma struct do tipo *hostent*, que contêm dados do *host*,nomeadamente, o endereço IPv4 e o nome do servidor FTP.

Deste modo, é possível criar a ligação TCP *(Transmission Control Protocol)* abrindo um *socket* na porta 21. Para tal, é usada a função *open\_socket()*, que abre um *socket,* na porta especificada em parâmetro e faz a conexão ao servidor. É ainda usada a função *fdopen()* para receber a *stream* de *bytes* pelo *socket*.

Na verdade, este *socket* é usado para enviar e receber comandos, para tal usamos as funções *write\_to\_socket() e* *read\_from\_socket()*, respetivamente. Caso não seja especificado um utilizador e uma palavra-passe é feito o *login* em modo anónimo, ou seja, enviando os comandos *user anonymous* e *pass anonymous*. Caso seja especificado, ambos os comandos anteriores são enviados, mas com o respetivo nome de utilizador e palavra-passe. Se o servidor só aceitar o *login* em modo anónimo, a ligação é cancelada se este não for o *login* utilizado.

Após o *login* ser efetuado, é iniciada a ligação em modo passivo com recurso ao comando *pasv* gerando uma resposta do servidor. Essa resposta é tratada pela função *check\_port()* com fim a retirar os dois últimos valores que vão ser necessários para calcular a porta essencial na criação de um segundo *socket* que irá tratar da transferência do ficheiro. Esta nova porta é calculada na função *calc\_port()* através da equação .

Este segundo *socket* é aberto nesta nova porta, de modo semelhante ao anterior. Paralelamente é enviado o comando *retr* com o diretório do ficheiro no primeiro *socket*, tal como, em caso de sucesso, é criado o ficheiro no diretório da aplicação. O ficheiro criado é preenchido com o que receber no segundo *socket* e à medida que é preenchido uma barra de progresso é atualizada. Esta barra de progresso compara o tamanho de *bytes* recebidos com o tamanho total do ficheiro que é previamente indicado na resposta do servidor ao comando *retr*.

Por último, são fechadas as *streams* de *bytes* dos *sockets* e consequentemente os *sockets* através da função *fclose().*

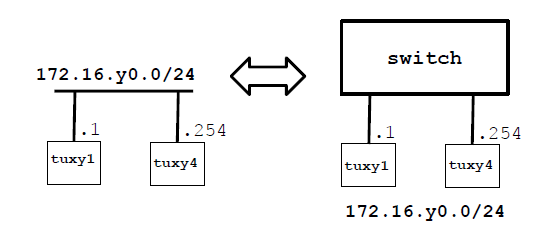
A aplicação foi testada tanto em ambiente laboratorial como em máquinas virtuais, sendo visíveis no anexo 1 o resultado no terminal de comandos de um utilizador e no anexo 2 os comandos FTP enviados ao servidor e alguns pacotes de dados da transmissão do ficheiro.

# Parte 2 – Configuração e Estudo de uma Rede

# Exp 1 – *Configure an IP Network*

# Arquitetura da rede

Primeiramente, foi desconectado o switch da porta 5.1 (*netlab*) e conectados os cabos conforme o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*) e *tux* 53 (*tuxy1*) da bancada 5.



A seguir, foi necessário configurar o *eth0* do *tux* 53 com o IP 172.16.50.1/24 (máscara de 255.255.255.0) e o *eth0* do *tux* 54 com o IP 172.16.50.254/24, recorrendo aos comandos presentes no anexo 3.

# Análise e discussão dos resultados

* ***What are the ARP packets and what are they used for?***

ARP (*Address Resolution Protocol*) é um protocolo da camada 2 utilizado para converter endereços IP em endereços físicos MAC (*Media Access Control*).

O ARP transmite um pacote para todos os *hosts* conectados à *Ethernet.* Este pacote contém o endereço IP com que o transmissor está interessado em comunicar.

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

Na verdade, os pacotes ARP contêm os endereços MAC e IP tanto do transmissor tanto do recetor.

Quando otransmissor quer iniciar uma transmissão e só sabe o endereço IP do recetor, o pacote ARP contem o endereço MAC e IP deste, mas só o endereço IP do recetor, pois como o seu endereço MAC é desconhecido é ignorado.

* ***What packets does the ping command generate?***

O comando *ping* gera pacotes ICMP *(Internet Control Message Protocol)*. O pacote enviado do transmissor é chamado *ICMP\_echo\_request* e o pacote enviado do recetor *ICMP\_echo\_reply.*

Além disso, este comando identifica o alcance de um *host*, indicando a existência de erros na rede estabelecida, perda de pacotes e ainda estatísticas dos resultados.

* ***What are the MAC and IP addresses of the ping packets?***

Os pacotes ICMP, enviados aquando de um comando *ping*, contêm os endereços MAC e IP do transmissor e do recetor.

* ***How to determine if a receiving Ethernet frame is ARP, IP, ICMP?***

A trama *Ethernet* contem no seu *header* um campo que identifica o tipo de protocolo dentro da trama (*EtherType*). Este campo tem 2 *bytes* (16 bits). No caso do protocolo IP o valor deste campo é 0x0800*,* e caso o campo de protocolo da trama IP tiver o valor 1 o protocolo internet é do tipo ICMP. No caso do protocolo ARP o valor é 0x0806.

* ***How to determine the length of a receiving frame?***

O tamanho da trama recebida está indicado no campo *length* do header da trama *Ethernet.*

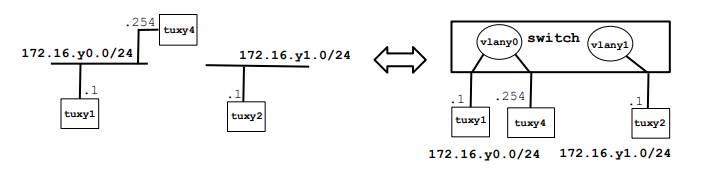
* ***What is the loopback interface and why is it important?***

A *interface* *loopback* é responsável por enviar pacotes *LOOP*, de volta ao transmissor para testar a ligação. Cada pacote *LOOP* tem um espaçamento de 10s.

# Exp 2 – *Implement two virtual LANs in a switch*

# Arquitetura da rede

Primeiramente, foram conectados os cabos consoante o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 1.



Do mesmo modo que demonstrado na experiência anterior, foi configurado o *eth0* do *tux* 52 com o IP 172.16.51.1/24. Depois, foi usada a porta 2 para a conectar o *tux* 52 à *vlan* 51, recorrendo aos comandos no anexo 4 no GTKTerm.

De igual modo, foram configuradas a porta 13 para conectar o *tux* 53 à *vlan* 50 e a porta 14 para conectar o *tux* 54 à *vlan* 50.

# Análise e discussão dos resultados

* ***How to configure vlany0?***

A ideia por detrás da configuração das *vlan* 50 já foi referida na parte da arquitetura da rede desta experiência, estando os comandos da configuração identificados no anexo 4.

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

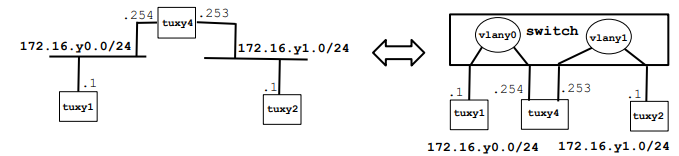
Ao fazer um *ping* em *broadcast* do *tux* 53, recebemos resposta do *tux* 54, mas não do *tux* 52, pelo que existem dois domínios *broadcast*. Infelizmente não foi possível guardar os *logs* dos *pings* em modo *broadcast* porque a *pen* USB não era reconhecida.

# Exp 3 – *Configure a Router in Linux*

Devido a um imprevisto no processo de configuração da bancada 5 foi optado por realizar esta experiência na bancada 6 pelo que nos anexos a nomenclatura considerada é a desta. No entanto, a nomenclatura usada a seguir considerará a bancada 5.

# Arquitetura da rede

Primeiramente, foram conectados os cabos segundo o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 2.



Do mesmo modo que demonstrado na experiência anterior, foi configurado o *eth1* do *tux* 54 com o IP 172.16.51.253/24.

Tal como na experiência anterior, foi criada a *vlan* 50, tendo associada a porta 13 (*tux* 53) e a porta 14 (*eth0* do *tux* 54) e a *vlan* 51, tendo associada a porta 2 (*tux* 52) e a porta 4 (*eth1* do *tux* 54).

Além disso, foram configuradas as *gateways* do *tux* 53 e do *tux* 52 com recurso aos comandos no anexo 5.

Para terminar também foi ativado o IP *forwarding* e desativado o *echo-ignore-broadcast* do ICMP no *tux* 54 (anexo 6), tendo em atenção que estas características foram definidas na configuração inicial das experiências posterior a esta.

# Análise e discussão dos resultados

* ***What routes are there in the tuxes? What are their meaning?***

O *tux* 53 contem a rota para a *vlan* 51 através da *gateway* da *vlan* 50 (.254) e o *tux* 52 contem a rota para a *vlan* 50 pela *gateway* da *vlan* 51 (.253). Estas rotas permitem a comunicação entre *hosts* nas diferentes *vlans*. A tabela de roteamento do *tux* 52 encontra-se no anexo 7.

* ***What information does an entry of the forwarding table contain?***

Uma tabela de encaminhamento contém o endereço de destino, a *gateway, a* rota da qual o destino pode ser alcançado e a *interface* de rede utilizada pela rede.

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

Como ambas as *gateways* estão no *tux* 54 são verificadas mensagens ARP entre o *tux* 53e o *tux* 54 e entre o *tux* 54 e o *tux* 52. Estas mensagens são do seguinte tipo: o recetor envia a mensagem ARP: *“Who has <<endereço IP>>”* e o transmissor envia a mensagem ARP: *<<endereço IP>> is at <<endereço MAC>>* e são necessárias para ser possível estabelecer a ligação entre o *tux* 53 e o *tux* 52. Uma captura das mensagens ARP entre o tux 54 e o tux 53 encontra-se no anexo 8.

* ***What ICMP packets are observed and why?***

Os pacotes ICMP observados são os pacotes *ICMP\_echo\_request* e *ICMP\_echo\_reply* resultantes do ping entre o *tux* 53 e o *tux* 52.

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

Analisando os logs realizados no *tux* 54 (anexos 9 e 10) é possível observar que aquando de um *ping* do *tux* 53 para o *tux* 52 o endereço IP de origem é o do *tux* 53 e o de destino é o do *tux* 52 tanto no *eth0* (*vlan* 50) como no *eth1* (*vlan* 51).

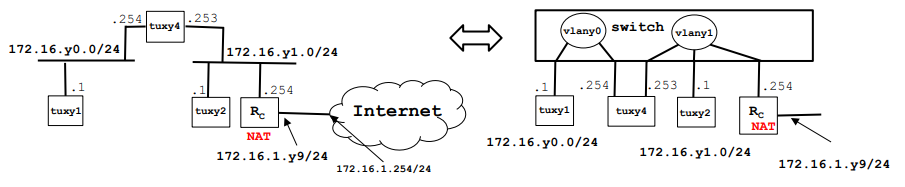
No entanto, no *eth0* o endereço MAC de origem é o do *tux* 53 e o de destino o do *eth0* do *tux* 54 a *gateway* da vlan50, enquanto no *eth1* o endereço de origem é o do *eth1* do *tux* 54 a *gateway* da vlan51 e o de destino o do *tux* 52.

Na verdade, como previamente é feita uma troca de mensagens ARP entre o tux 53 e o *tux* 54 e entre o *tux* 52 e o *tux* 54, os endereços MAC do *tux* 54 são usados para redirecionar os pacotes, pois nem o tux 53 nem o tux 52 sabem o endereço MAC um do outro.

# Exp 4 – *Configure a Commercial Router and Implement NAT*

# Arquitetura da rede

Primeiramente, foram conectados os cabos conforme o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 3.



A seguir, foi necessário configurar o *router* da CISCO (Rc), recorrendo aos comandos do anexo 11 no GTKTerm para definir a *interface inside* com o IP 172.16.51.254/24 e a *interface outside* com o IP 172.16.1.59/24. Também são definidas as rotas estáticas dos endereços de origem e de destino. É importante realçar que, inicialmente, não foi ativada a funcionalidade NAT (*Network Address Translation*) e os comandos relativos a esta componente foram só usados após o 6º passo do guião.

Tal como na experiência anterior, foi criada a *vlan* 50, tendo associada a porta 13 (tux 53) e a porta 14 (*eth0* do *tux* 54) e a *vlan* 51, tendo associada a porta 2 (*tux* 52), a porta 4 (*eth1* do *tux* 54) e a porta 7 (*eth0* do Rc).

Além disso, foram configuradas as *gateways* do *tux* 53, do *tux* 52 e do *tux* 54 com recurso aos comandos no anexo 12.

# Análise e discussão dos resultados

* ***How to configure a static route in a commercial router?***

A ideia por detrás da configuração de rotas estáticas no *router* já foi referida na parte da arquitetura da rede desta experiência, estando os comandos da configuração identificados no anexo 11.

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

Inicialmente, com a rota 172.16.50.0/24 via *tux* 54 definida e com o redirecionamento ICMPativo é possível realizar o *ping* entre os *tux* 52 e o *tux* 53. No entanto, sem a rota definida e com o redirecionamento ICMP ativo já é possível, sendo que em caso deste não estar ativo já não é.

Além disso, realizando o *traceroute* é possível verificar que quando a rota está definida, o pacote inicialmente vai para o *tux* 54 e, a seguir, para o *tux* 53. Enquanto, quando a rota não está definida e o redirecionamento ICMP está ativo, o pacote vai inicialmente para o *router* e só depois para o *tux* 54*.*

* ***How to configure NAT in a commercial router?***

A ideia por detrás da configuração do *router* com a funcionalidade de NAT já foi referida na parte da arquitetura da rede desta experiência, estando os comandos da configuração identificados no anexo 11.

* ***What does NAT do?***

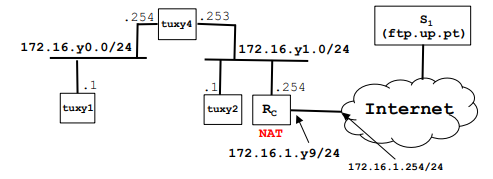
O NAT *(Network Adress Translation)* é uma técnica utilizada para converter endereços IP privados, internos a uma rede, em endereços IP públicos, permitindo que estes consigam comunicar com outras redes.

Este comportamento foi verificado na última fase desta experiência, visto que antes de adicionar a NAT ao *router* (Rc)não era possível comunicar com o router da sala e, após esta ser adicionada, já era possível.

# Exp 5 – DNS

# Arquitetura da rede

Primeiramente, foram conectados os cabos consoante com o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 4.



A seguir, foi verificado que o *tux* 54, o *tux* 53 e o *tux* 52 já tinham o DNS configurado, recorrendo ao comando no anexo 13.

# Análise e discussão dos resultados

* ***How to configure the DNS service at a host?***

Para configurar o servidor DNS no *host* é necessário aceder ao ficheiro *etc/resolv.conf* e adicionar o nome do servidor e o endereço IP deste.

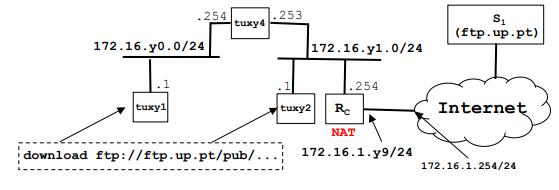
* ***What packets are exchanged by DNS and what information is transported***

Ao realizar um *ping* do *tux* 52 para a *google.com,* o *tux* 52 envia dois pacotes para o servidor DNS da FEUP (172.16.1.1)a “perguntar” quais os endereços IP (IPv4 e IPv6) do domínio *google.com.*  De seguida, recebe do servidor DNS da FEUP dois pacotes de resposta com os endereços IP. No anexo 14 conseguimos observar os pacotes DNS.

# Exp 6 – TCP *connections*

# Arquitetura da rede

Primeiramente, foram conectados os cabos segundo o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 5.



# Análise e discussão dos resultados

* ***How many TCP connections are opened by your ftp application?***

São abertas *2* conexões, uma na porta 21 para a transferência de comandos e respostas FTP e uma na porta 59220, definida pela aplicação após entrar em modo passivo, para a transferências de dados. No anexo 15 conseguimos observar essas conexões e as portas associadas.

* ***In what connection is transported the FTP control information?***

A informação de controlo é transportada na conexão de transferência de comandos e respostas FTP na porta 21.

* ***What are the phases of a TCP connection?***

As fases de uma conexão TCP são 3. A fase de estabelecimento da conexão, a fase de troca de dados e a fase de encerramento da conexão. A fase de estabelecimento e de encerramento encontram-se nos anexos 15 e 16.

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

O mecanismo ARQ *(Automatic Repeat Request),* nomeadamente, o método *Go Back N,* é utilizado pelo protocolo TCP para controlo de erros na transmissão de dados. Este controlo de erros é efetuado por pacotes ACK *(acknowledgment),* que, através do número de sequência, que indica o número do pacote, o número ACK, que indica se o pacote foi corretamente recebido e o *window size,* que indica a quantidade de informação recebida que pode ser armazenada,controla a transferência de dados. Estes campos encontram-se demonstrados no anexo 17.

* ***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 to the TCP congestion control mechanism?***

O TCP usa uma janela de congestionamento que controla a quantidade de bytes enviados na rede para que esta não fique sobrecarregada e neste caso diminui a velocidade de transmissão.

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

O *throughput* de uma conexão TCP irá sofrer uma redução caso surja uma segunda conexão TCP, pois a largura de banda disponível também é diminuída.

# Conclusão

Assim sendo, a aplicação FTP implementada cumpriu os objetivos principais deste trabalho, tendo passado com sucesso os testes efetuados na experiência 6, tal como, explorados conhecimentos de configuração e estudo de uma rede de computadores em ambiente laboratorial.

# Anexos

**Anexo 1 – Aplicação FTP**

**Uma imagem com texto, captura de ecrã, computador

Descrição gerada automaticamente**

**Uma imagem com texto

Descrição gerada automaticamente**

**Uma imagem com texto

Descrição gerada automaticamente**

**Anexo 2 – Captura Wireshark da aplicação FTP**

**Uma imagem com mesa

Descrição gerada automaticamente**

**Anexo 3 – Configuração IPs**

ifconfig eth0 up (tux 53)

ifconfig eth0 172.16.50.1/24 (tux 53)

ifconfig eth0 (tux 53)

ifconfig eth0 up (tux 54)

ifconfig eth0 172.16.50.254/24 (tux 54)

ifconfig eth0 (tux 54)

**Anexo 4 – Configuração *VLANs***

configure terminal

vlan 51

end

show vlan id 51

configure terminal

interface fastethernet 0/2

switchport mode access

switchport access vlan 51

end

show running-config interface fastethernet 0/2

show interfaces fastethernet 0/2 switchport

**Anexo 5 – Configuração rotas Exp 3**

route add -net 172.16.51.0/24 gw 172.16.50.254 (tux 53)

route add -net 172.16.50.0/24 gw 172.16.51.253 (tux 52)

**Anexo 6 – Ativação IP *forwarding* e desativação do *echo-ignore-broadcast* do ICMP**

echo 1 > /proc/sys/net/ipv4/ip\_forward

echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts

**Anexo 7 – Exp 3 - captura no *tux* 52 - *routing table***

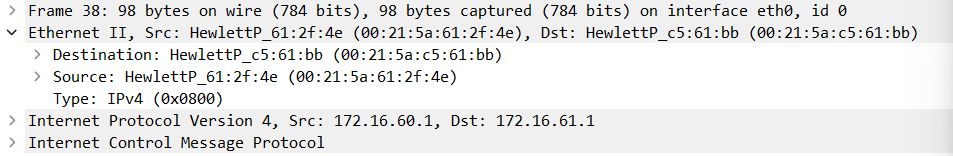
**Uma imagem com texto

Descrição gerada automaticamente**

**Anexo 8 – Exp 3 - captura no *tux* 54 - *mensagens* ARP**

****

**Anexo 9 – Exp 3 - captura no *tux* 54 no *eth0* - *ping* do *tux* 53 para o *tux* 54**



**Anexo 10 – Exp 3 - captura no *tux* 54 *no* eth1 - *ping* do *tux* 53 para o *tux* 54**

Uma imagem com texto

Descrição gerada automaticamente

**Anexo 11 – Configuração *router* com NAT**

configure terminal

interface gigabitethernet 0/0

ip address 172.16.51.254 255.255.255.0

no shutdown

ip nat inside (NAT)

exit

configure terminal

interface gigabitethernet 0/1

ip address 172.16.1.59 255.255.255.0

no shutdown

ip nat outside (NAT)

exit

ip nat pool ovrld 172.16.1.59 172.16.1.59 prefix 24 (NAT)

ip nat inside source list 1 pool ovrld overload (NAT)

access-list 1 permit 172.16.50.0 0.0.0.7 (NAT)

access-list 1 permit 172.16.51.0 0.0.0.7 (NAT)

ip route 0.0.0.0 0.0.0.0 172.16.1.254

ip route 172.16.50.0 255.255.255.0 172.16.51.253

end

**Anexo 12 – Configuração rotas Exp 4**

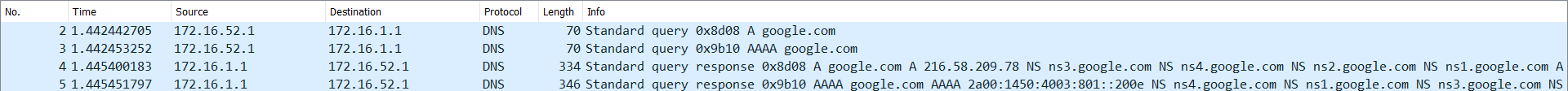
route add default gw 172.16.50.254 (tux 53)

route add default gw 172.16.51.254 (tux 52)

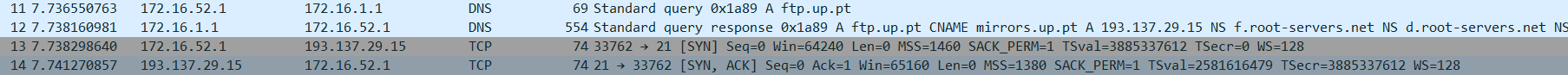
route add default gw 172.16.51.254 (tux 54)

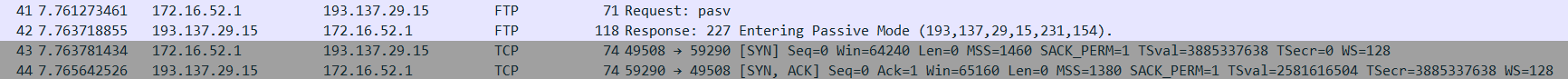
**Anexo 13 – Verificação DNS**

vi /etc/resolv.conf

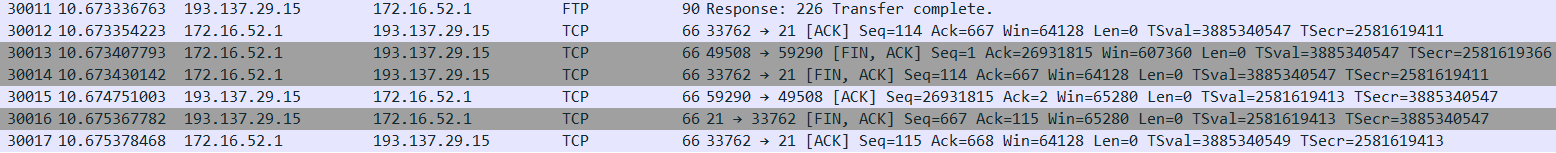
**Anexo 14 – Exp 5 - pacotes DNS**

**Anexo 15 – Exp 6 - Conexões TCP**





**Anexo 16 – Exp 6 – Encerramento Conexão TCP**



**Anexo 17 – Exp 6 – Campos TCP**

**Uma imagem com texto

Descrição gerada automaticamente**