

Redes de Computadores

*2º Trabalho Laboratorial*

*Mestrado Integrado em Engenharia Informática e Computação*

*(22 de dezembro de 2017)*

Bárbara Silva  **up201505628**@fe.up.pt

Catarina Ferreira **up201506671**@fe.up.pt

Julieta Frade **up201506530**@fe.up.pt

Sumário

Este relatório foi elaborado no âmbito da unidade curricular de Redes de Computadores, e trata-se de uma complementação ao segundo trabalho laboratorial, cuja essência é Redes de Computadores. O trabalho consiste na configuração e estudo da mesma, em que foram utilizados comandos de configuração do Router Cisco e do Cisco Switch, e no desenvolvimento de uma aplicação de download de um ficheiro através de um cliente TCP (Transmission Control Protocol).

Isto posto, o trabalho foi concluído com sucesso, visto que todos os objetivos estabelecidos foram cumpridos e foi finalizada uma aplicação capaz de transferir um ficheiro, assim como a rede foi configurada corretamente.

Índice

[**Introdução 3**](#_Toc501571952)

[**Parte 1: Aplicação de Download 4**](#_Toc501571953)

[Arquitetura 4](#_Toc501571954)

[Resultados 5](#_Toc501571955)

[**Parte 2: Configuração de Rede e Análise 6**](#_Toc501571956)

[Experiência 1 – Configure na IP Network 6](#_Toc501571957)

[Experiência 2 – Implement two virtual LANs in a switch 7](#_Toc501571958)

[Experiência 3 – Configure a Router in Linux 8](#_Toc501571959)

[Experiência 4 – Configure a Commercial Router and Implement NAT 9](#_Toc501571960)

[Experiência 5 - DNS 10](#_Toc501571961)

[Experiência 6 – TCP Connections 11](#_Toc501571962)

[**Conclusões 12**](#_Toc501571963)

[**Referências 13**](#_Toc501571964)

[**Anexos 14**](#_Toc501571965)

# *Introdução*

Texto.

# *Parte 1: Aplicação de Download*

A primeira parte deste trabalho foi desenvolver uma aplicação download na linguagem de programação C que aceita um link como argumento **ftp://<user>:<password>@<host>/<url-path>**. Esta aplicação é um cliente TCP (*Transmission Control Protocol*), e consegue fazer o download de qualquer tipo de ficheiros de um servidor FTP (*File Transfer Protocol*). Para isso foi estudado o RFC959 que fala sobre o FTP e o RFC1738 que fala sobre o tratamento de informação provenientes de URL’s.

## **Arquitetura**

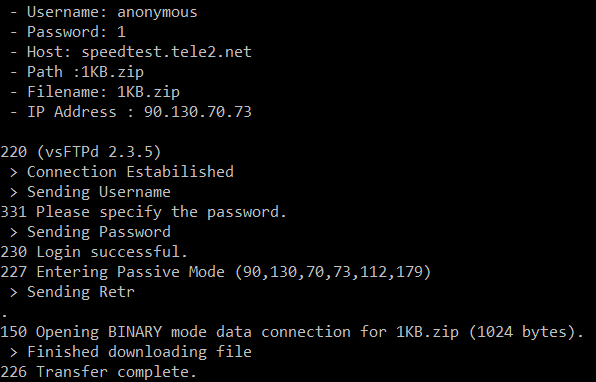
Primeiramente é feito o processamento do URL. Para tal, é reservado espaço para as variáveis *user*, *password*, *host* e *path* e depois é chamada a função ***parseArguments*** para obter estas mesmas variáveis a partir do URL. O nome do ficheiro é obtido a partir do *path* e o i*p address* a partir da função ***getip*** (código fornecido). A porta usada é sempre a 21.

Uma das funções principais é a ***sendCommandInterpretResponse*** que chama a função ***readResponse*** (obtém o código de resposta enviado pelo servidor) e analisa a resposta. O primeiro digito diz se a resposta é positiva (1, 2, 3) ou negativa (4 e 5). Caso o primeiro digito seja 1, é chamado a ***readResponse*** novamente para ser lida outra resposta, caso seja 2 e 3 a função retorna, caso seja 4 tenta ler-se novamente a resposta e caso seja 5 o programa termina.

Após ser aberto o *socket* pelo qual será feita a primeira conexão entre o cliente e o servidor, são enviados os comandos **USER user** e **PASS pass** para ser feito o login. Depois, é feita a entrada em modo passivo pela chamada do comando **PASV**, que vai retornar a porta necessária para a abertura de um outro *socket* que servirá para troca de dados. É enviado o comado **RETR** **filename** para pedir o ficheiro e seguidamente é feito o download do ficheiro com a ajuda da função ***createFile***. No final, são fechadas as duas conexões tanto a de transferência de comandos como a de dados.

**Resultados**

O nosso programa foi testado em diversas condições: modo anónimo, modo não anónimo, vários tipos de ficheiro, vários tamanhos do ficheiro entre outros. Este termina em caso de erro ou caso o ficheiro não exista. Todas as respostas são imprimidas na consola para maior controlo por parte do utilizador.



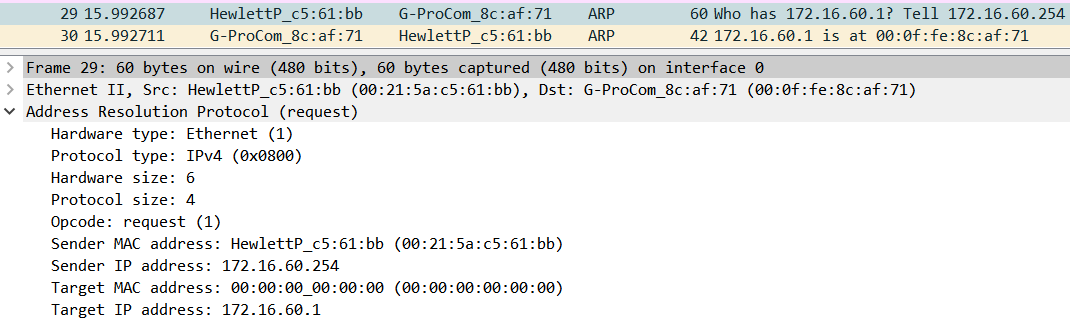
# *Parte 2: Configuração de Rede e Análise*

## **Experiência 1 – Configurar um IP de rede**

1. O que são os pacotes ARP e para o que são usados?

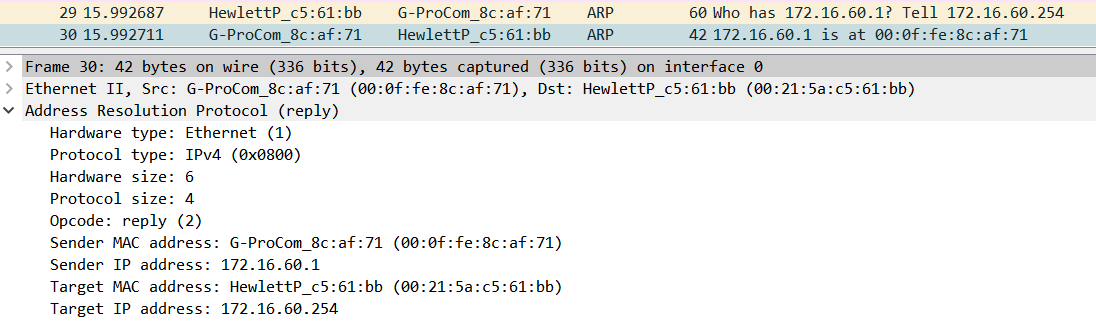
O ARP (*Address Resolution Protocol*) é um protocolo de comunicação que serve para descobrir o endereço da camada de ligação associado ao endereço IPv4. Serve para mapear o endereço de rede a um endereço físico como o endereço MAC.

1. Quais são os endereços MAC e IP dos pacotes ARP e porquê?



Quando fazemos *ping* do tux 1 para o tux 4, o tux 4 envia um pacote a perguntar quem é o tux com aquele IP, ou seja, a perguntar que endereço MAC tem o tux que lhe está a tentar mandar algo. Esta pergunta vem na forma de um pacote ARP com o endereço IP e endereço MAC do tux 4 (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.

De seguida, o tux 1 responde a dizer que é ele que tem aquele IP enviando o seu endereço MAC.



Neste pacote de resposta o endereço IP e MAC da origem são o do tux 1 (172.16.60.1 e 00:0f:fe:8c:af:71 respetivamente) e o endereço IP e MAC do destino são o do tux 4 (172.16.60.254 e 00:21:5a:c5:61:bb).

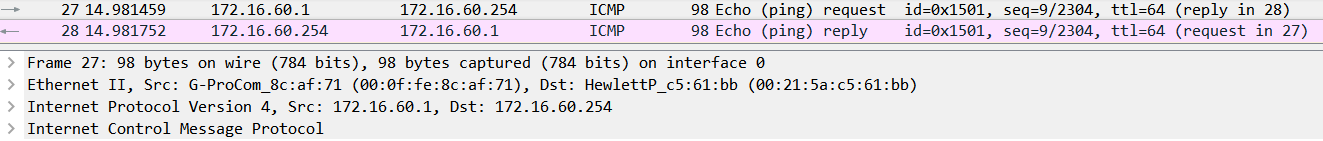
1. Quais os pacotes gerados pelo comando *ping*?

O comando *ping* gera primeiro pacotes ARP para obter os endereços MAC e de seguida gera pacotes ICMP (*Internet Control Message Protocol*).

1. Quais são os endereços MAC e IP dos pacotes *ping*?

Quando damos *ping* ao tux 4 a partir do tux 1 os endereços (origem e destino) IP e MAC dos pacotes vão ser os destes tux’s.

**Pacote de pedido:**



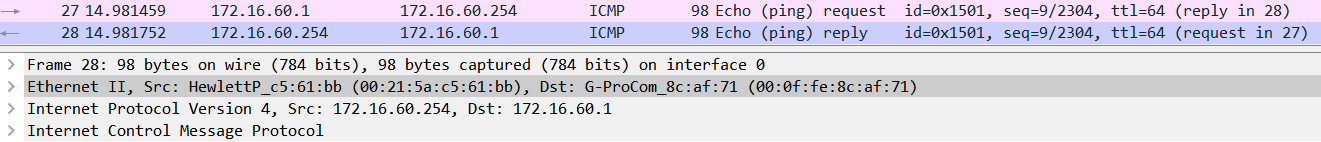
Endereço MAC da origem do pacote: 00:0f:fe:8c:af:71 (tux 1).

Endereço MAC do destino do pacote: 00:21:5a:c5:61:bb (tux 4).

Endereço IP da origem do pacote: 172.16.60.1 (tux 1).

Endereço IP do destino do pacote: 172.16.60.254 (tux 4).

**Pacote de resposta:**



Endereço MAC da origem do pacote: 00:21:5a:c5:61:bb (tux 4).

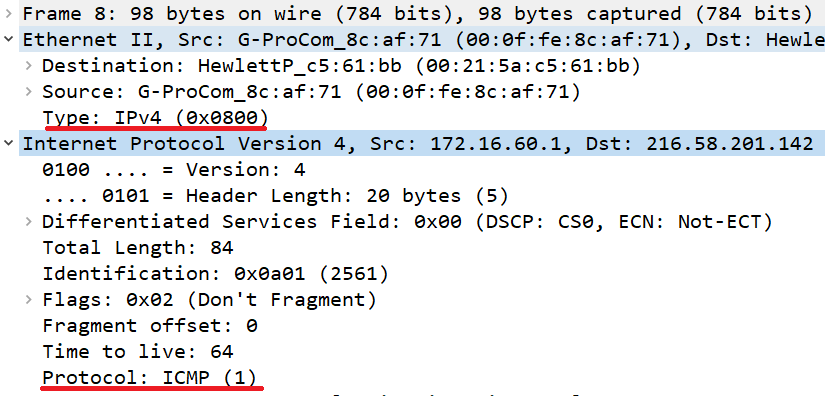
Endereço MAC do destino do pacote: 00:0f:fe:8c:af:71 (tux 1).

Endereço IP da origem do pacote: 172.16.60.254 (tux 4).

Endereço IP do destino do pacote: 172.16.60.1 (tux 1).

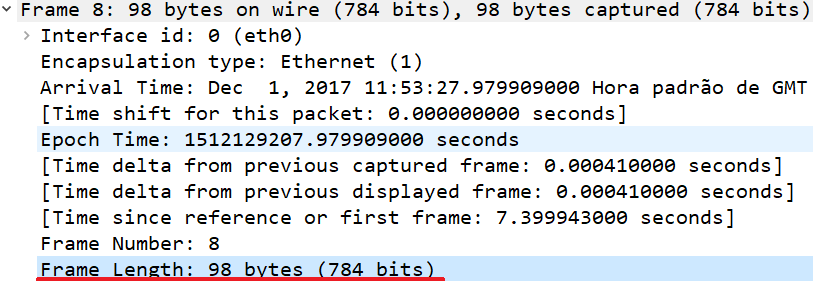
1. Como determinar se a trama recetora Ethernet é ARP, IP, ICMP?

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.



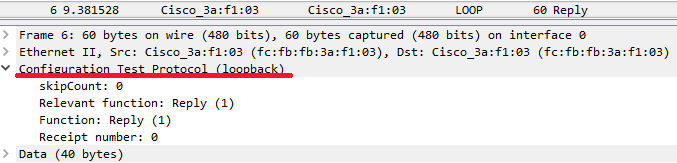
1. Como determinar o comprimento de uma trama recetora?

O comprimento de uma trama recetora é determinado inspecionando-a usando o **wireshark**.



1. O que é a interface *loopback* e porque é que é importante?

A interface *loopback* é uma interface virtual da rede que permite ao computador receber respostas de si mesmo. É usada para testar se a carta de rede está configurada corretamente.



## **Experiência 2 – Implementar duas LAN’s virtuais no switch**

1. Como configurar 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* invocam-se os seguintes comandos no **GTKTerm** do tux escolhido:

* configure terminal
* vlan y0
* end

Depois deverá adicionar-se as portas dos tux 1 e 4:

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

1. Quantos domínios de transmissão existem? O que se pode concluir a partir dos registros?

Existem dois domínios de transmissão, visto que o tux 1 recebe resposta do tux 4 quando faz *ping* *broadcast*, mas não do tux 2. O tux 2 não recebe resposta de ninguém quando faz *ping* *broadcast*. Assim, existem dois domínios de *broadcast*: o que contém o tux 1 e tux4 e o que contém o tux 2.

## **Experiência 3 – Configurar um router em Linux**

1. Que rotas há nos tuxes? Qual o seu significado?

As rotas para as vlans associadas:

1. Tux 1 tem uma rota para a vlan 0 (172.16.y0.0) pela gateway 172.16.y0.1.
2. Tux 4 tem uma rota para a vlan 0 (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.
3. Tux 2 tem uma rota para a vlan 1 (172.16.y1.0) pela gateway 172.16.y1.1.

As rotas que foram criadas durante a experiência:

1. Tux 1 tem uma rota para a vlan 1 (172.16.y1.0) pela gateway 172.16.y0.254.
2. Tux 2 tem uma rota pata a vlan 0 (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.

1. Que informação é que uma entrada da tabela de *forwarding* contém?

**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.

**Flags**: dá-nos informações sobre a rota.

**Metric**: o custo de cada rota.

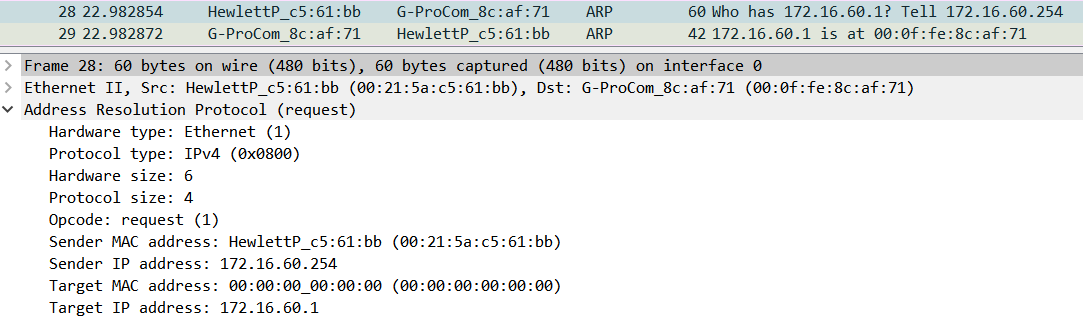
**Ref:** número de referências para esta rota (não usado no *kernel* do Linux).

**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).

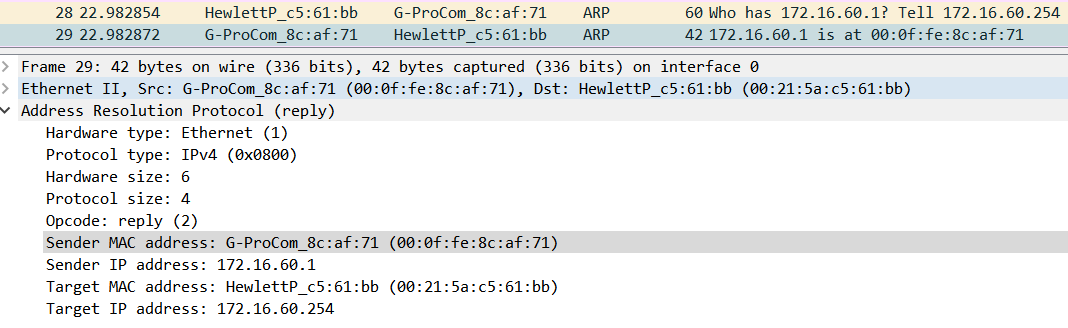
**Interface:** qual a placa de rede responsável pela *gateway* (eth0/eth1).

1. Que mensagens ARP e endereços MAC associados são observados e porquê?

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 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.

1. Que ICMP packets são observados e porquê?

São observados pacotes ICMP de pedido e resposta, pois depois de serem adicionadas as rotas todos os tux’s se conseguem ver uns aos outros. Se não se conseguissem ver, seriam enviados os pacotes ICMP de *Host Unreachable*.

1. Quais são os endereços IP e MAC associados a um ICMP packet e porquê?

Os endereços IP e MAC associados com os pacotes ICMP são os endereços IP e MAC dos tux’s de origem e destino. Por exemplo, quando se faz *ping* do tux 1 para o tux 4 (.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 – Configurar um router comercial e implementar o NAT**

1. Como se configura um router estático num router comercial?

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 gw]
* exit

1. Quais são as rotas seguidas pelos pacotes durante a experiência? Explique.

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.

1. Como se configura o NAT num router comercial?

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 x presente no anexo.

1. O que faz o NAT?

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 registrados 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 tux’s 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. Como configurar o serviço DNS num *host*?

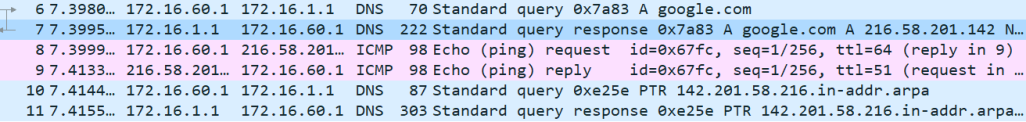
De forma a configurar o serviço DNS, é necessário mudar o ficheiro **resolv.conf** que se localiza em **vi/etc** no *host* tux. Esse ficheiro tem de conter a seguinte informação:

* 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 experiencia, é possível aceder à internet nos tux’s.

1. Que pacotes são trocados pelo DNS e que informações são transportadas?

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.



O servidor responde (linha 7) com um pacote que contem o endereço IP do *hostname*.

## **Experiência 6 – Conexões TCP**

1. Quantas conexões TCP foram abertas pela aplicação FTP?

A aplicação FTP abriu duas conexões TCP, uma para troca de comandos e outra para troca de dados.

1. Em que conexão é transportado o controlo de informação?

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

1. Quais as fases da conexão TCP?

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

1. Como é que o mecanismo ARQ TCP funciona? Quais os campos TCP relevantes? Qual a informação relevante observada nos logs?

O TCP (*Transmission Control Protocol*) utiliza o mecanismo ARQ (*Automatic Repeat Request*). Este consiste no controlo de erros na transmissão de dados. Para tal, utiliza ***acknowledgments***, que são mensagens enviadas pelo recetor que indicam que a trama foi recebida corretamente, e ***timeouts***, que é o tempo máximo de espera por um *acknowledgment*. No caso de um *acknowledgment* não ser recebido antes do *timeout*, a trama é retransmitida até este ser recebido. Assim, é garantida uma transmissão confiável.

1. Como é que o mecanismo de controlo de congestão TCP funciona? Como é que o fluxo de dados da conexão evoluiu ao longo do tempo? Está de acordo com o mecanismo de controlo de congestão TCP?

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.

1. De que forma é afetada a conexão de dados TCP pelo aparecimento de uma segunda conexão TCP? Como?

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*

Texto.

# *Referências*

* https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html

# *Anexos*

