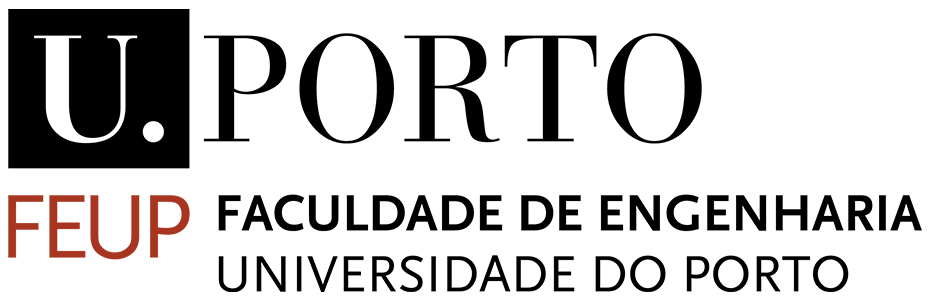
Configuração de uma rede e desenvolvimento de uma aplicação de download

**Relatório Final**

****

Mestrado Integrado Em Engenharia Informática e Computação

Redes de Computadores

**Professor:**

Sérgio Reis Cunha

**Turma 3, Bancada 4:**

Miguel Rodrigues Pires – up201406989

Diogo Henrique Pereira – up201505318

André Fernando Fernandes – up201505821

Faculdade de Engenharia da Universidade do Porto

Rua Roberto Frias, sn, 4200-465 Porto, Portugal

**Sumário**

Este relatório incide sobre o segundo projeto da Unidade Curricular de RCOM (Redes de Computadores), do curso MIEIC. O projeto consiste na configuração de uma rede bem como no desenvolvimento de uma aplicação de *download*. O projeto divide-se, então, em duas fases de desenvolvimento, as quais são:

* Desenvolvimento da aplicação *download*, onde vai ser descrita a sua arquitetura e apresentados os resultados da sua execução;
* A configuração de uma rede, onde vão ser apresentados as seis experiências elaboradas durantes as aulas da UC.

As experiências acima referidas basearam-se na configuração de um IP de rede, da configuração de duas LAN’s (*Local Area Network*), as VLAN’s 0 e 1, virtuais no *switch*, onde os computadores se encontravam em série, e através do NAT (*Network Address Translation*), e ainda de um *router* em Linux, de um *router* comercial e do DNS (*Domain Name System*).

**Índice**

**1. Introdução**

O segundo projeto de RCOM desenvolveu-se ao longo de diversas aulas laboratoriais, tendo a primeira aula servido de exemplo dos protocolos de IETF (*Internet Engineering Task Force*), tais como o *Telnet* e FTP. Estes protocolos promovem a solução de problemas relacionados com ligações á *Internet* usando documentos RFC que descrevem os protocolos. O protocolo que implementamos neste projeto foi o FTP usando o servidor da faculdade como, por exemplo, o tom.fe.up.pt. Aqui tivemos que criar uma aplicação de *download*, seguindo as instruções do guião, que procede à transferência de um ficheiro e que implemente o protocolo *FTP*.

Quanto à configuração de uma rede de computadores, o seu principal objetivo é executar uma aplicação, através de duas *VLAN’s*. Na *VLAN* 2 está implementado o *NAT* e não na *VLAN* 1, sendo esta possível de ter ligação à *Internet*, através da *VLAN* 2.

O relatório tem a seguinte estrutura:

* **Introdução**, onde é explicado o objetivo do projeto;
* **Aplicação de *download*,** onde é explicado o cliente FTP e a sua implementação;
* **Configuração de rede,** onde são explicados todos os passos feitos e todos os objetivos, experiência a experiência;
* **Conclusão,** onde é abordado o trabalho na sua totalidade, analisando o que podia ser feito, e a opinião geral do grupo;
* **Referências,** que consiste em todos os documentos e web*sites* utilizados pelo grupo;
* **Anexos,** que contêm o código, imagens, comandos corridos e *logs* gravados.

**2. Parte 1 – Aplicação de download**

# Como referido anteriormente, a primeira parte deste trabalho consiste numa aplicação que transfere um ficheiro utilizando o protocolo FTP descrito no ficheiro RFC959.

A aplicação desenvolvida permite que seja feito um *download* em modo anónimo. Para tal basta não colocar os caracteres ‘@’ e ‘:’ e não colocar nome de utilizador e password. Neste caso a aplicação irá assumir o utilizador *anonymous* e a palavra-passe vazia.

# **2.1 Arquitetura**

A aplicação divide-se em três partes. Primeiramente processa a string fornecida pelo utilizador que contém o url. Depois, processa-se o ip e a porta. A última parte diz respeito à conexão com o servidor FTP.

Estruturas de dados:

Foram usadas duas structs para guardar informação: struct FTP e struct URL. (colocar imagens das duas)

A struct URL guarda informação extraída da string input do utilizador., através das funções parseURL e getip(imagens das funções).

A função parseURL valida o input, e extrai toda a informação possível da string. Depois, a função getip obtém o ip e a porta.

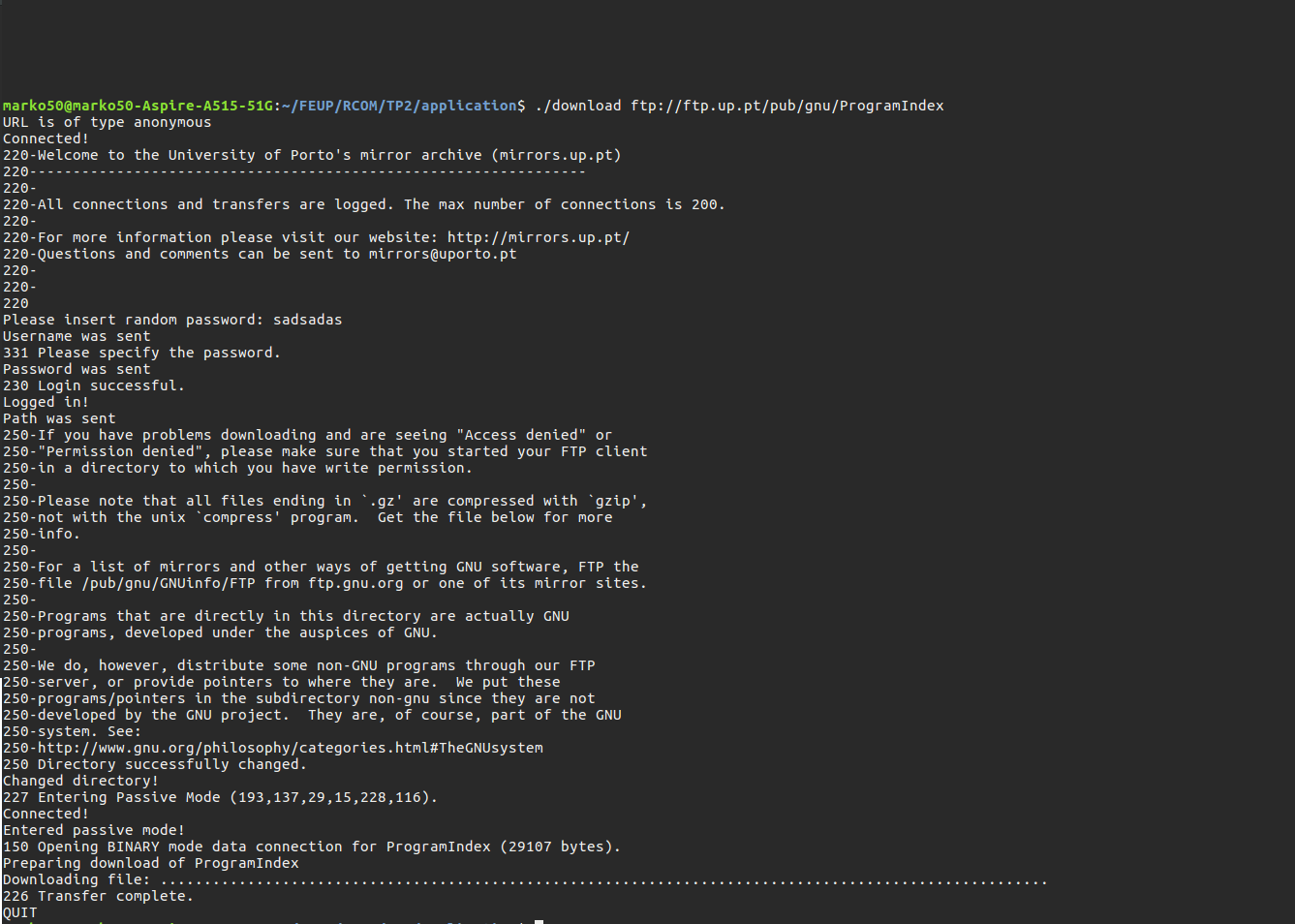
De seguida passa para a última parte da aplicação, que se destina ao contacto com o servidor FTP.

Primeiro conecta-se ao servidor FTP, com a função connectftp(imagem da função) que retorna a socket de controlo de conexão. Depois dá-se o login, com a função(loginftp) , usando o comando “USER \_username\_” e “PASS \_password\_”. Seguidamente muda-se de diretório para o diretório desejado, usando o comando “CWD \_directory\_” na função changedirectoryftp(imagem), seguindo-se a entrada em modo passivo com a função passiveMode(imagem). Esta função envia o comando “PASV” e recebendo como resposta, um ip e uma porta, que irão ser usados para criar a socket de transferência de dados. Esta e a outra socket são guardadas na struct FTP. Prepara-se de seguida o download do ficheiro usando o comando “RETR \_filename\_” na função retrftp(imagem) e faz-se o download com a função downloadftp(imagem).

No fim faz-se a desconexão usando a função disconnectftp(imagem).

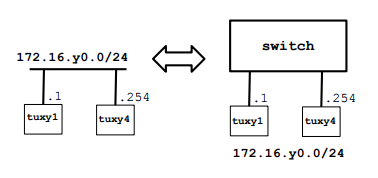
# **2.2 Resultados de download**

Este é um exemplo da execução da aplicação para <ftp://ftp.up.pt/pub/gnu/ProgramIndex>, transferindo um ficheiro de 24Kb

****

**3. Parte 2 – Configuração da rede e análise**

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



Esta primeira experiência teve como finalidade compreendermos a configuração de **IP’s** em máquinas diferentes, de modo a que estas consigam comunicar entre si.

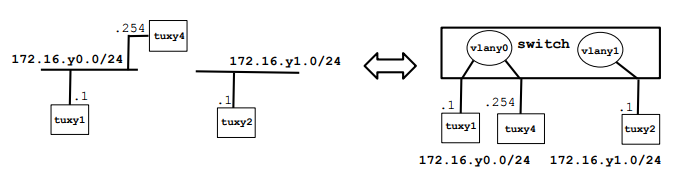
Para a configuração dos dois computadores **tux41** e **tux44** foi utilizado o comando ***<ifconfig ‘ipaddress’ eth0>***, que atribui ao IP da interface o IP passado como argumento e ativa as portas eth0 às quais foram ligados os cabos de rede. Neste caso o tux41 assume o IP de 172.16.40.1 e o tux44 assume 172.16.40.254. Como os computadores se encontram na mesma rede, não foi necessário usar o comando ***route***. Também foi necessário configurarmos a Vlan 40 para as portas 1 e 2, usando os comandos ***<interface fastethernet 0/nº porta>***, ***<switchport mode access>*** e ***<switchport access vlan 40>***.

Após esta configuração, executamos um comando ***ping*** de uma máquina para a outra com os IP’s definidos, no qual verificamos que existia ligação entre as duas. De seguida inspecionamos as tabelas **ARP** com o comando ***<arp –a>***, e depois apagamos todas as entradas na tabela ARP do tux41 através do comando ***<arp –d 172.16.40.254>***.

Por fim repetiu-se o comando *ping* registando o processo através do programa *wireshark*. Analisando o *log* do *wireshark* (ver figura nos Anexos), podemos verificar que, tendo apagado as entradas da tabela ARP é perguntado à rede qual o endereço **MAC** com um endereço **IP** igual a 172.16.40.254. Este computador responde com o seu endereço MAC e, a partir de aí, sempre que o primeiro faz um *request* **ICMP**, este é seguido de uma resposta do segundo.

A interface *loopback* é uma interface de rede virtual que o computador utiliza para comunicar com ele próprio, com o objetivo de realizar testes de diagnóstico, ou aceder a servidores na própria máquina, como se fosse um cliente. Assim, uma interface *loopback*, permite a existência de um endereço IP no *router*, que está sempre ativo, em vez de ser dependente de uma interface física.

**3.2 Experiência 2 – Implementar duas redes *LAN* virtuais num *switch***



Esta experiência consiste na criação de duas *VLAN’s* diferentes:

* *VLAN 40 – 172.16.40.0/24* – é constituída pelos computadores **tux41** e **tux44**;
* *VLAN 41 – 172.16.41.0/24* – é constituída pelo **tux42**.

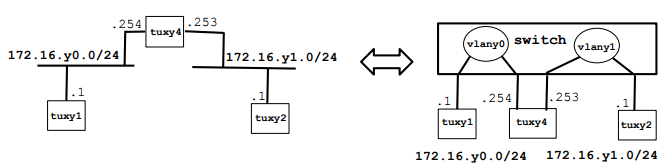
No final desta configuração, o computador **tux42** deixará de ter acesso aos computadores tux41e tux44, uma vez que se encontram em sub-redes diferentes. Para configurar os computadores da forma referida basta proceder do mesmo modo que se procedeu para a primeira experiência, mas desta vez fazê-lo no computador tux42 e atribuído **172.16.41.1** como endereço de **IP** (tendo em conta que os outros dois computadores continuam com a configuração da primeira experiência).

Para a configuração do ***switch*** foi necessário entrar na sua consola de configuração, através da aplicação ***GkTerm***, e executar o comando ***<vlan [n]>***, onde ***n*** é o número identificador da VLAN. Após a configuração das VLAN’s foi necessário adicionar as portas do *switch* às respetivas VLAN’s, para criar duas sub-redes individuais. Para isso usaram-se os comandos ***<interface fastethernet 0/[i]>***, onde ***i*** é o identificador da porta do *switch*, seguido de ***<switchport mode access>*** e ***<switchport access VLAN [n]>***, onde ***n*** é o identificador da VLAN criada. Os cabos ligados às portas do *switch* estão ligados à interface 0 dos computadores.

Estando os dois computadores na mesma rede, ao fazer ***ping*** do tux41 para o tux44, é enviado o pacote **ARP** para saber o endereço **MAC**. Executando um *ping* para o computador tux42, esse *ping* falhou como era esperado, uma vez que se encontrava numa sub-rede diferente, inacessível tanto pelo computador tux41 como pelo tux44.

Pode também verificar-se que cada VLAN tem um *broadcast domain* diferente, tendo em conta que o tux42 não detectou o pacote enviado, como tal, conclui-se que a configuração destas redes foi feita da forma correta pois verifica-se a falta de comunicação entre as redes, como referimos inicialmente. Logo, existem dois *broadcast domains*, em qualquer um dos computadores consegue-se *pingar* o respectivo *broadcast domain,* mas não se obtém um IP de resposta.

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



Esta experiência consiste na configuração do computador tux 44 como *router* de forma a ligar as duas *VLAN’s* existentes criadas na experiência anterior.

Para isso foi necessário ativar a interface *ethernet* 1 do computador tux44 e configurá-la com um IP dentro da mesma gama que o computador tux42, e adicionar esta interface à sub-rede do tux42. Esta porta é a que será ligada á *vlan 41*. Configuramos a porta com o endereço IP **172.16.41.253/24**. Sendo este computador aquele que pretendemos que sirva como *router*, é necessário ativar o reencaminhamento de IP’s através do comando ***<echo 1 > /proc/sys/net/ipv4/ip\_forward>***.

Por fim, adicionou-se as rotas necessárias no tux41 e no tux42 de forma a que estes, através do tux44, pudessem aceder à rede a que não pertencem. Para o tux41 usamos o comando ***<route add -net 172.16.41.0/24 gw 172.16.40.254>***. O primeiro endereço indica a gama de endereços para a qual se quer adicionar a rota e o segundo endereço identifica o IP para o qual se deve reencaminhar o pacote, neste caso o IP do tux44. Repetiu-se o mesmo procedimento para o tux42, mas usando os endereços ***<route add -net 172.16.40.0/24 gw 172.16.41.253>***.

Finalmente, foi possível *pingar* o tux42 a partir do tux41. O pedido para o IP do tux42 (172.16.41.1) é reencaminhado para o tux44 (172.16.40.254). Como o tux44 está ligado à sub-rede de ambos os computadores, consegue aceder ao tux42 através da sua interface eth1, que está nessa sub-rede, e assim reencaminha o pacote para o tux42. Na resposta o processo é idêntico, sendo o pacote reencaminhado do tux42 para o tux41.

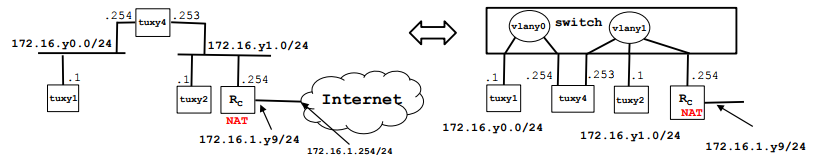
No tux y1 sao visiveis a defaiult route da rede em que esta e tambem uma rota para a sub-rede 172.16.41.0/24 pelo tuxy4 (172.16.40.254) no tuxy2 verifica-se tambem uma defualt route para a sub-rede onde esta inserido e uma rota para 172.16.40.0/24 pelo tuxy4 (172.16.41.253). Finalmente no tuxy4 verifica-se que existem duas rotas default para as duas sub-redes.

Uma entrada na routing table contém informação sobre rotas para uma determinada rede destinatária, tal como a gateway a utilizar, a interface e a máscara.

Quando o tux41 dá ping ao tux42, observam-se pacotes ARP, nos 3 pc’s que indicam endereços MAC de vários computadores para se poder enviar informação na ida e na vinda. Para o tux41 comunicar com o tux42, é preciso saber o seu endereço MAC, e antes disso o endereço MAC da interface eth0 do tux44. O tux44 responde e o pacote é reencaminhado para lá, e depois tem de enviar pela interface eth1, que já sabe o seu endereço NAT. A interface eth1 depois tem de reencaminhar para o tux42, mas precisa do seu endereço NAT, que aparece num pacote ARP, construido, tal como todos após um ping broadcast, e o pacote chega ao tux42. Depois é preciso o tux42 comunicar com o tux41 seguindo uma lógica semelhante.

Os pacotes IMCP aparecem porque na tabela de rotas existem rotas por gateways específicas para enviar pacotes. O destinatário é o ip do tux42 e o emissor é o ip do tux41.

**3.4 Experiência 4 – Configurar um *router* comercial implementando o *NAT***



Esta experiência tem como objetivo a configuração de um router comercial dentro da rede 41 de forma a que computadores tanto na *VLAN 40* como na *VLAN 41* tenham acesso à *Internet*. A implementação do NAT (*Network Address Translation*) tem como objetivo possibilitar a comunicação entre os computadores da rede criada com redes externas. Por se tratar de uma rede privada, os IP’s nunca seriam reconhecidos fora da rede. Por isso criou-se um método que permite reescrever os IP’s de origem de uma rede interna, para que possam aceder a uma rede externa. Este procedimento gera um número de 16 bits, utilizando esse valor numa *hash table*, e escrevendo-o no campo da porta de origem. Na resposta, o processo é revertido e o *router* sabe para qual computador da rede interna deve enviar a resposta.

Para configurar o *router*, depois de fazer *login*, foi necessário configurar a interface interna no processo de NAT. Para isso entramos na consola de configuração da interface *fastethernet 0/0* do *router*, com o comando ***<interface fastethernet 0/0>***. De seguida teve de ser especificado qual o IP para essa interface, introduzindo o comando ***<ip address 172.16.41.254 255.255.255.0>***. Posteriormente, foi configurada a interface externa, atribuindo um IP à interface 1, que estava ligada ao *router* da sala, introduzindo os seguintes comandos ***<interface fastethernet 0/1>***, ***<ip address 172.16.2.49 255.255.255.0>***. Para ambos os casos foi necessário introduzir o comando ***<no shutdown>***, para que estas configurações se mantivessem caso o *router* fosse desligado.

De seguida para que fosse garantida a gama de endereços, introduziram-se os comandos ***<ip nat pool ovrld 172.16.2.49 172.16.2.49 prefix 24>*** e ***<ip nat inside source list 1 pool ovrld overload>***. Foi criada também uma lista de acessos e permissões de pacotes, para cada uma das sub-redes, com o comando ***<access-list 1 permit ip [máximo]>***. Neste caso, o IP utilizado foi 172.16.40.0 e 172.16.41.0, que poderia ir até 172.16.4X.255, colocando 0.0.0.255 no campo máximo.

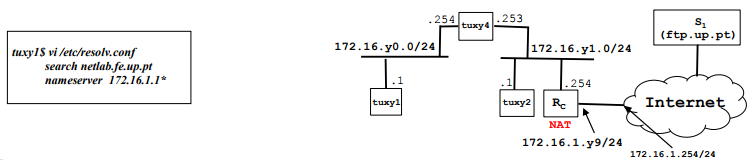
Depois de permitir que os computadores das redes criadas tenham acesso, é configurada uma rota pré-definida para o endereço *Internet*. Da mesma forma é adicionada aos computadores a rota pré-definida para o endereço 172.16.41.254, endereço do *router* comercial. Definimos rotas internas e externas, usando os comandos ***<ip route 0.0.0.0 0.0.0.0 172.16.2.254>*** e ***<ip route 172.16.40.0 255.255.255.0 172.16.41.253>***. Isto cria uma nova rota, quando o IP de destino for 172.16.40.0-255 deve redirecionar os pacotes para o IP 172.16.41.253.

Para testarmos fizemos um *ping* no tux41 para o *router* da sala e verificou-se que os pacotes enviados para este computador passavam pelo tux42, onde eram reencaminhados para o *router* no IP 172.16.41.254.

Quando se dá ping do tux41 para um endereço ip, por exemplo 8.8.8.8, que é o caso mais complexo na rede criada, este tem uma rota para enviar pacotes cujo destinatário é deste género para a gw da interface eth0 do tux44. Por sua vez o tux44, como reconhece o ip do router, envia os pacotes para lá. O router, para responder, tem uma rota, que quando quer enviar pacotes para a subrede do tux41, envia pela interface eth1 do tux44.

NAT reescreve os endereços ip privados de uma rede local, como públicos de modo a que os computadores desta rede tenham acesso à rede mundial de computadores.

**3.5 Experiência 5 – *DNS***



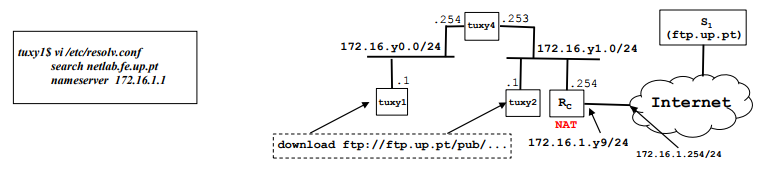
O objetivo desta experiência era conseguir aceder a redes externas, conseguindo desta forma aceder à Internet, através da rede interna criada. Para isto, foi apenas necessário configurar o **DNS**.

Esta configuração passa por, em todos os *hosts* da rede criada, aceder e editar o ficheiro **resolv.conf**, situado no diretório **/etc** nos computadores com o sistema operativo em *Linux*. Este ficheiro é lido cada vez que são invocadas rotinas que fornecem acesso à *Internet*. Neste caso, o ficheiro foi editado colocando as linhas:

* ***search services.netlab.fe.up.pt***
* ***nameserver 172.16.2.1***

Por fim foi testada a funcionalidade desta alteração fazendo o teste de *ping* usando *www.google.com*. Nos *logs*, consequentemente, verificou-se que o DNS pergunta a informação contida num dado *domain name*, e este responde com o tempo de vida e o tamanho do pacote de dados.

**3.6 Experiência 6 – Ligações *TCP***



Por fim, a experiência 6 tinha como objetivo juntar as duas partes deste trabalho, pondo em prática a **aplicação de *download****,* desenvolvida na Parte 1, na rede configurada ao longo destas experiências.

Para testar a aplicação, foi usado um servidor **FTP** e efetuado o *download* de um ficheiro, em modo anónimo e não anónimo. O *download* efetuou-se corretamente, o que demonstrou que a rede estava bem configurada, não trazendo qualquer problema no acesso por protocolo FTP, assim como à utilização de um servidor exterior à rede.

TCP utiliza *Selective Repeat ARQ*, que é semelhante ao *GO-BACK-N ARQ*, com a diferença que o receptor não deixa de processar os frames recebidos quando detecta um erro. Quando a falha de um frame é detetada, o receptor continua a receber e a processar as frames seguintes, enviando sempre, no *ack*, o número da frame que falhou primeiro. No final do envio, o emissor verifica os *ack* e reenvia os frames perdidos.

A aplicação usa duas conecções FTP, uma para comunicar com o servidor e outra para transferência de dados. A informação de controlo FTP é transportada na conexão de controlo com o servidor. Uma conexão FTP é iniciada com a conexão a um determinado ip numa determinada porta. Essa conexão é feita criando uma socket de controlo. Depois, dependendo do servidor, dá-se o login no servidor usando a socket de controlo. Entra-se em modo passivo, criando a socket de transferência de dados, transfere-se os dados a partir dessa socket e no fim, desconexão.

Em TCP, o emissor mantém uma janela de congestão que possui o número de bytes máximo(varia) que o emissor pode enviar ao recetor.

A abertura de uma nova conexão TCP para trasnferência de dados leva a uma queda na taxa de transmissão.

**4. Conclusão**

Após a conclusão do segundo projeto de Redes e Computadores, o grupo conseguiu captar quase todos os conceitos necessários para uma implementação estável e coerente daquilo que era pedido no guião.

A configuração de rede foi concluída com sucesso permitindo a todos os elementos do grupo ter uma noção de como tudo funciona e assim aplicar esta prática a nível profissional. A aplicação de download implicava um prévio estudo de alguns protocolos como os que são descritos nos RFC959 e RFC1738 sobre FTP e sintaxe do URL, respetivamente.

Todos os objetivos delineados pelo grupo foram atingidos com sucesso, com exceção da experiência 7, de cariz opcional, que não implementamos para a apresentação ao docente.

**5. Referências**

**Anexos**