

# Trabalho Prático Nº1 – Protocolos da Camada de Transporte

Duração: 4 aulas

Este trabalho deve ser realizado com recurso à máquina virtual XubunCORE\_7\_5 que está disponibilizada em <http://marco.uminho.pt/ferramentas/CORE/xubuncore.html> (user: core password: core)

## Relatório

O relatório final do TP1 deve incluir:

- Uma secção de "Questões e Respostas" que dê resposta adequada às questões enumeradas no enunciado, incluindo para cada questão: a questão, a resposta e a prova da realização da mesma (se aplicável);
- Uma secção de "Conclusões" que autoavale os resultados da aprendizagem decorrentes das várias vertentes estudadas no trabalho.

## Submissão

O relatório deve ser escrito em formato de artigo com um máximo de 10 páginas (recomenda-se o uso do formato LNCS - Lecture Notes in Computer Science, instruções para autores em <http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0>). A submissão se dá na plataforma de ensino <https://elearning.uminho.pt>, usando a funcionalidade de transferência de ficheiros do grupo, com o nome CC-TP1-PL<Turno>-G<Grupo>.pdf (por exemplo, CC-TP1-PL1-G1.pdf para o grupo 1 do PL1) no final do dia da aula prevista para a conclusão do trabalho.

## Objetivo

- familiarizar-se com as ferramentas que serão utilizadas ao longo do curso, CORE, Wireshark e seus recursos.
- testar a conectividade e analisar as características gerais dos links (ligações com diferentes larguras de banda, diferentes atrasos e perdas de pacotes) utilizando os comandos “ping”, “traceroute” e “iperf”;
- Realizar a transferência de ficheiros entre dispositivos com protocolos que se utilizam de diferentes protocolos de transporte e analisar o seu uso em condições de rede.

## Questões

### Parte I: Instalação, configuração e validação da rede de testes

**1.1** – Defina em modo de edição uma topologia com quatro roteadores. Faça uma ligação do nó n1 para o nó n2, deste para o nó n3, e deste para o nó n4, resultando numa topologia em anel. Em cada um desses roteadores, ligue um *host*. Renomeie os *hosts* como PCx, onde x é o mesmo dígito que identifica o roteador a que está ligado. Por exemplo, PC1 é o *host* ligado ao roteador n1.

Verifique que são atribuídos automaticamente endereços de rede IPv4 e IPv6 aos vários nós. Apague os endereços IPv6 e deixe apenas os IPv4.

Inspecione as ligações que interligam os nós. Configure o débito das ligações entre os roteadores e *hosts* a 10 Mbps. Configure as demais ligações, entre os roteadores, da seguinte maneira:

- Entre os nós 1 e 2: Utilize um débito de 10 Mbps, atraso de 0 ms e perdas de 0%.
- Entre os nós 2 e 3: Utilize um débito de 5 Mbps, atraso de 5 ms e perdas de 1%.
- Entre os nós 3 e 4: Utilize um débito de 2 Mbps, atraso de 10 ms e perdas de 5%.
- Entre os nós 4 e 1: Utilize um débito de 1 Mbps, atraso de 20 ms, perdas de 10% e 10% de duplicações.

Inicie a simulação e responda às questões seguintes:

**1.2** – Verifique que todas as rotas foram configuradas com sucesso e demonstre que os hosts possuem ligação entre si. Utilize-se das ferramentas traceroute, ping e iperf para verificar as rotas entre hosts, as estimativas de perdas de pacotes, atrasos e débito fim-a-fim.

**1.3** – As configurações das rotas foram realizadas dinamicamente pelo protocolo OSPF.

- a) Para obter melhores resultados de débito, atraso e perdas de pacotes, quais rotas alteraria? Justifique.
- b) Caso se desejasse manter o uso do OSPF, seria possível melhorar as rotas definidas dinamicamente? Como?

OBS: Salve a topologia para uso na Parte III deste TP.

## Parte II: Uso da camada de transporte por parte das aplicações

Neste exercício pretende-se utilizar diferentes protocolos de aplicação para transferência de ficheiros (TFTP, FTP e HTTP), para acesso remoto (TELNET e SSH), além de outras aplicações fundamentais para o funcionamento da Internet. Todos os pacotes trocados durante o exercício devem ser capturados com o Wireshark para a análise e fundamentação da resposta das questões. Para isso será necessário realizar os seguintes passos:

Verificar se o software (cliente e servidor) está instalado e instalar se necessário [Máquina XubunCORE Linux (host principal), usando linha de comando (bash), user core, password core]	
Comandos	Observações
\$ sudo apt install openssh-server \$ sudo apt install openssh-client	O software SSH já deve estar instalado de raiz no Linux e o serviço SSH/SFTP já está configurado e ativo por omissão em todas as topologias virtuais criadas pelo CORE; Não deverá ser necessário fazer nada de especial para usar SSH.
\$ sudo apt install vsftpd	Existem vários packages de software FTP para Linux. Neste exercício sugere-se a utilização do “vsftpd”. O cliente ftp já existe no Linux e não é preciso instalar.
\$ sudo apt install atftpd \$ sudo apt install atftp	Para software servidor TFTP propõe-se o uso do “atftpd”, que é um servidor tftp avançado e também do respetivo cliente “atftp”. Não existe nenhum servidor ou cliente pré-instalado.
\$ sudo apt install mini-httpd \$ sudo apt install wget \$ sudo apt install lynx	O software “mini-httpd” foi escolhido por ser um servidor web simples e que usa poucos recursos. Já o “lynx” e o “wget” são clientes Web, para consola, muito usados e poderosos!

Numa linha de comando da máquina virtual **XubunCORE**, mas fora do emulador, sem ativar o **core-gui**, execute:

```
$ sudo wireshark
```

Capture o tráfego em determinados instantes que considere adequados, observe atentamente como as várias aplicações utilizam os serviços da camada inferior:

- Acesso via browser ao URL: <http://marco.uminho.pt/disciplinas/CC-LEI/> (nota: pode usar comandos `wget` ou `lynx`)
- Acesso em `ftp` para <ftp.eq.uc.pt> (Username: `ftp` Password: `cc2023` Comando: `get README / quit`). OBS: Para utilizar a partir da VM XubunCORE utilize o passive mode (`ftp -p ftp.eq.uc.pt`).
- Acesso em `tftp` para [cc2024.ddns.net](http://cc2024.ddns.net) (usando `tftp` ou `curl`, por exemplo: `curl -v tftp://cc2024.ddns.net/file1`)
- Acesso via `telnet` para [193.136.9.33](http://193.136.9.33) (Username: `gcom` Password: `gcom` Comando: `exit`)
- Acesso `ssh` para [cc2024.ddns.net](http://cc2024.ddns.net) (Username: `cc` Password: `cc2024`, `ssh cc@cc2024.ddns.net`)
- Resolução de nomes usando `nslookup` [www.uminho.pt](http://www.uminho.pt)
- `ping` [www.google.pt](http://www.google.pt)
- `traceroute` [cc2024.ddns.net](http://cc2024.ddns.net) ou [www.fccn.pt](http://www.fccn.pt)

... e outras aplicações Internet bem conhecidas que considere importantes e que possa explorar!

**2.1-** Com base no trabalho realizado, identifique para cada aplicação executada, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e *overhead* de transporte.

Comando usado (aplicação)	Protocolo de Aplicação (se aplicável)	Protocolo de transporte (se aplicável)	Porta de atendimento (se aplicável)	Overhead de transporte em bytes (se aplicável)
wget, lynx ou via browser				
ssh, sftp				
ftp				
Tftp				
telnet				
nslookup ou dig				
Ping				
Traceroute				
Outras:				

## Parte III: Utilização de serviços de transferência de ficheiro no ambiente CORE.

Neste exercício pretende-se transferir ficheiros utilizando os protocolos TFTP, FTP e HTTP no ambiente do CORE. Para tal, deve-se criar os ficheiros (*file1* e *file2*) para as transferências e correr os servidores/clientes, conforme as instruções no Anexo I. Utilize a topologia criada na Parte I onde o PC1 deverá ser utilizado como servidor – o *host* que possui o ficheiro a ser partilhado com os demais.

**3.1** – Descarregue os ficheiros a partir do PC3 com os protocolos TFTP e FTP e responda:

- De que forma as perdas de pacotes afetaram o desempenho das aplicações? Que camada lidou com as perdas: transporte ou aplicação? Responda com base nas experiências feitas e nos resultados observados.
- Apresente um diagrama temporal para a transferência do *file1* por FTP. Foque-se apenas na transferência de dados [ftp-data] e não na conexão de controlo, pois o FTP usa mais que uma conexão em simultâneo. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão.
- Apresente um diagrama temporal para a transferência do *file1* por TFTP. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

**3.2** – Descarregue os ficheiros a partir do PC2 com os protocolos TFTP, FTP e HTTP e responda:

**a)** Na transferência HTTP:

- Identifique o início e o fim da sessão TCP e analise como os números de sequência e ACKs são usados na conexão.
- Identifique o número de sequência inicial e analise como ele é incrementado com cada pacote tanto pelo cliente quanto pelo servidor.

**b)** Qual dos protocolos seria o mais adequado para a obtenção dos ficheiros pelo PC2? Justifique.**3.3** – Descarregue os ficheiros a partir do PC4 com os protocolos TFTP, FTP e HTTP e responda:**a)** Na transferência HTTP:

- Identifique a perda e a duplicação de pacotes numa sessão TCP.
- Explique o impacto da perda e duplicação de pacotes numa sessão TCP, bem como os mecanismos usados pelo TCP para lidar com estas situações.

**b)** Qual dos protocolos seria o mais adequado para a obtenção dos ficheiros pelo PC4? Justifique.**3.4** – Simule uma congestão de rede fazendo o **iperf** gerar uma taxa de bits por segundo superior à largura de banda do canal (conexão) a partir de um *host*. Investigue como a janela de congestão muda durante o evento de congestão.

- Explique como os mecanismos de controle de congestão do TCP (ex.: slow start, congestion avoidance) ajustam o fluxo de dados.
- Apresente a taxa de transferência (throughput) da conexão TCP e compare-a com o débito calculado na Parte I.
- Forneça imagens das capturas de tráfego para suportar suas observações.

## Anexo I – Instalação, configuração e utilização de serviços de transferência de ficheiros

**ETAPA:** preparar uma pasta com os ficheiros a transferir; um ficheiro de texto e um ficheiro binário;  
*[Máquina XubunCORELinux (host principal), usando linha de comando (bash), user core, password core]*

Comandos	Observações
<pre>\$ sudo mkdir -p /srv/ftp \$ sudo usermod -d /srv/ftp ftp \$ sudo sh -c "yes 'A'   head -c 10K &gt; /srv/ftp/file1" \$ sudo dd if=/dev/urandom of=/srv/ftp/file2 bs=1K count=10</pre>	<p>O servidor FTP instala um novo utilizador no sistema com <i>username</i> "ftp" sem password para poder servir ficheiros da <i>home</i> desse utilizador de forma anónima a qualquer cliente FTP. A pasta a criar chama-se <i>"/srv/ftp"</i>. O comando <i>mkdir</i> criará a pasta se ela não existir (e todas as incluídas no path que forem necessárias – opção <i>"-p"</i>). O comando <i>usermod</i> faz dela a <i>"home"</i> do user <i>"ftp"</i>.</p> <p>Depois são criados para lá dois ficheiros: o <i>"file1"</i>, que é um ficheiro de texto com 10 KB, e o <i>"file2"</i>, ficheiro binário que também possui 10 KB. Pode optar por colocar ou editar outros ficheiros nessa pasta. Tudo o que estiver lá ficará acessível.</p>

Para transferir por **ftp** é necessário executar o servidor manualmente na *bash* do PC1.

#### Topologia Virtual, PC1 (servidor FTP)

```
root@PC1$ chmod a-w /srv/ftp
... directoria não pode ter acesso para escrita... por questões de segurança...

root@PC1$ vsftpd /etc/vsftpd.conf -osecure_chroot_dir=/srv/ftp -oanonymous_enable=YES
... este comando vai manter-se em execução no terminal sem passar para background...
... pode ser enviado para background com um "control-Z", que suspende o processo, seguido do comando "bg", que o envia para background
... para mais informações sobre os parâmetros escrever "man vsftpd" ou "man vsftpd.conf" num terminal
```

**NOTA:** Estes comandos servem para configurar e ativar manualmente o servidor FTP na linha de comando!

E depois transferir o ficheiro a partir do PCx.

#### Topologia Virtual, PCx (cliente FTP)

```
root@PCx$ ftp <endereço-ip-do-PC1>
... entrar com username anonymous e qualquer password (aconselha-se o e-mail)
ftp> status
ftp> pwd
ftp> dir
ftp> get file1
ftp> quit
```

### **TFTP**

Para transferir por **tftp** é necessário preparar a directoria e executar o servidor manualmente na *bash* do PC1.

#### Topologia Virtual, PC1 (servidor TFTP)

```
root@PC1$ chmod -R 777 /srv/ftp
... directoria tem de ter acesso para escrita para todos...
root@PC1$ touch atftpd.log
... se houver problemas podemos ver neste ficheiro de log o que se passou...
root@PC1$ atftpd --verbose=3 --user root.ftp --logfile atftpd.log \
--bind-address <endereço-ip-do-PC1> --daemon --no-fork /srv/ftp/
... atenção que a barra \ serve para continuar o comando noutra linha e não é necessária se escrever tudo na mesma linha ...
... este comando vai manter-se em execução no terminal sem passar para background...
... pode ser enviado para background com um "control-Z", que suspende o processo, seguido do comando "bg", que o envia para background
... para mais informações sobre os parâmetros escrever "man atftpd" num terminal
```

**NOTA:** Estes comandos servem para configurar e ativar manualmente o servidor TFTP na linha de comando!

E depois transferir o ficheiro a partir do PCx:

#### Topologia Virtual, Portatil1 (cliente TFTP)

```
root@PCx$ atftp <endereço-ip-do-PC1>
ftp> status
ftp> get file1
ftp> quit
```

## **HTTP**

Para transferir por **http** é necessário preparar a diretoria e executar o servidor manualmente na *bash* do PC1.

### Topologia Virtual, PC1 (servidor HTTP)

```
root@PC1$ mini_httpd -d /srv/ftp/  
...  
root@PC1$ ps -ef  
... para verificar se o daemon ficou em execução...
```

**NOTA:** Estes **comandos** servem configurar e ativar manualmente o servidor HTTP, confirmando que ficou em execução.

E depois transferir o ficheiro a partir do PCx.

### Topologia Virtual, Portatil1 (cliente HTTP)

```
root@PCx$ wget http://<endereço-ip-do-PC1>/file1  
root@PCx$ wget http://<endereço-ip-do-PC1>/file2  
...  
Ou com o comando lynx:  
  
root@PCx$ lynx http://<endereço-ip-do-PC1>/file1  
root@PC1$ lynx http://<endereço-ip-do-PC1>/file2
```