

Trabalho Prático Nº1 – Protocolos da Camada de Transporte

Duração: 4 aulas

Este trabalho deve ser realizado com recurso ao emulador **CORE** (*Common Open Research Emulator*).

Existem duas formas recomendadas de o utilizar (pode escolher a que for mais conveniente):

- **Docker:** usar imagem preparada segundo as instruções em <https://github.com/eivarin/Dockerized-Coreemu-Template>
- **Máquina virtual:** descarregar a VM já pronta 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 <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>). 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 (semana de 13 a 18 de outubro – ver slides de apresentação das aulas práticas).

Objetivos

- 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.

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

1.1 – Criação de uma topologia de testes

Usando a interface gráfica do CORE no modo de edição, construa uma topologia com quatro routers. Faça uma ligação do nó n1 para o nó n2, do nó 2 para o nó 3, do nó 3 para o nó 4 e finalmente do nó 4 para o nó 1, resultando numa topologia em anel. Em cada um desses *routers*, ligue um *host*. Renomeie os *hosts* como PC_x, onde x é o dígito que identifica o *router* a que está ligado. Por exemplo, PC1 é o *host* ligado ao *router* n1 (nó 1). Verifique que são atribuídos automaticamente endereços de rede IPv4 e IPv6 aos vários nós.

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

- a) Execute a simulação, e faça os testes necessários para poder responder a todas questões que se seguem. Inclua no relatório uma imagem da topologia criada e em execução no CORE.

1.2 – Testes elementares de conectividade

Verifique que todas as rotas foram configuradas com sucesso e demonstre que os *hosts* possuem ligação entre si. Utilize as ferramentas *traceroute*, *pinge* *iperf* (preferencialmente a [versão iperf3](#)) para verificar as rotas entre *hosts*, as estimativas de perdas de pacotes, atrasos e débito fim-a-fim.

- a) Preencha uma pequena tabela com os valores obtidos.

1.3 – Testes TCP e UDP em condições adversas

Utilizando o comando *iperf*, versão *iperf3*, realize testes de débito entre dois nós, escolhendo a ligação que apresenta mais perdas de pacotes. Sugestão: Inicie o servidor num dos nós (`iperf3 -s`) e depois o cliente no outro nó, primeiro gerando tráfego TCP (`iperf3 -c <IP_do_servidor> -t 20`) e depois tráfego UDP (`iperf3 -c <IP_do_servidor> -u -b 20M -t 20`). Consulte o manual do *iperf3* para ver o significado das opções de linha de comando disponíveis.

- a) Compare os resultados obtidos (débito, perdas, estabilidade).
b) Com base nas diferenças observadas, discuta as vantagens e desvantagens de cada protocolo de transporte, dando exemplos de tipos de aplicações que beneficiem mais de um ou do outro.

NOTA: Grave a topologia para uso na Parte II deste TP.

Parte II: Utilização de serviços de transferência de ficheiros

Neste exercício pretende-se transferir ficheiros utilizando os protocolos aplicacionais TFTP, FTP e HTTP no ambiente do CORE. Para tal, deve criar dois 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. Utilize o [Wireshark](#) para capturar e analisar o tráfego trocado em cada transferência, de modo a poder produzir os diagramas temporais solicitados. Explore convenientemente as funcionalidades de filtragem de tráfego e de elaboração de diagramas do Wireshark.

2.1 – Transfira os dois ficheiros *file1* e *file2* a partir do PC3 (descarga do PC1 para o PC3) via TFTP e FTP.

- a) Identifique qual a rota que foi usada na transferência dos dados
b) Apresente um diagrama temporal para a transferência do *file1* por FTP. Uma vez que o FTP cria mais que uma conexão de transporte em simultâneo, normalmente uma para comandos e outra para os dados, foque-se apenas na transferência de dados [*ftp-data*] e não na conexão de controlo. Identifique no diagrama as fases de início de conexão, transferência de dados e fim de conexão, caso existam.
c) 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.
d) De que forma as perdas de pacotes afetaram o desempenho das aplicações nos dois casos anteriores? Qual a camada que lidou com as perdas: transporte ou aplicação? Responda com base nas experiências feitas e nos resultados observados.

2.2 – Transfira os dois ficheiros *file1* e *file2* a partir do PC4 (descarga do PC1 para o PC4) via TFTP, FTP e HTTP.

- a) Relativamente a esta transferência HTTP, identifique:
- O início e o fim da sessão TCP e os números de sequência e de confirmação (ACK) usados na conexão;
 - O número de sequência inicial e a forma como é incrementado com cada pacote tanto pelo cliente quanto pelo servidor.

- b)** Com base nas transferências efetuadas, qual dos protocolos aplicacionais experimentados (TFTP, FTP e HTTP) seria o mais adequado para a obtenção dos ficheiros pelo PC4 (descarga do PC1 para o PC4)? Justifique.

2.3 – Transfira os dois ficheiros *file1* e *file2* a partir do PC2 (descarga do PC1 para o PC2) via TFTP, FTP e HTTP.

- a)** Relativamente a esta 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)** Com base nas transferências efetuadas, qual dos protocolos aplicacionais experimentados (TFTP, FTP e HTTP) seria o mais adequado para a obtenção dos ficheiros pelo PC2 (descarga do PC1 para o PC2)? Justifique.

2.4 – Prepare e execute uma captura de tráfego de uma transferência TCP realizada, por exemplo, com *iperf3* ou *curl*.

Identifique e explique, com base nos pacotes capturados, **dois** dos mecanismos internos do TCP:

- Retransmissão de segmentos;
- Controlo de fluxo (janela de receção);
- Controlo de congestionamento.

Mostre evidências na captura (ex.: pacotes repetidos, valores de janela, variação do débito) que suportem a sua explicação.

Parte III: 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 Já devem estar instalados na Máquina Virtual e no Docker recomendados. <i>[Máquina Virtual: abrir terminal, usar linha de comando (bash) como utilizador core e password core]</i> <i>[Docker: abrir terminal com "docker exec -it core bash", usar linha de comando (bash) já como root]</i>	
Comandos	Observações
<pre>\$ sudo apt install openssh-server \$ sudo apt install openssh-client</pre>	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.
<pre>\$ sudo apt install vsftpd</pre>	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.
<pre>\$ sudo apt install atftpd \$ sudo apt install atftp</pre>	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.
<pre>\$ sudo apt install mini-httpd \$ sudo apt install wget \$ sudo apt install lynx</pre>	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 (fora do emulador) ou numa bash do *Docker*, execute o Wireshark para capturar os pacotes:

```
$ sudo wireshark
```

Capture o tráfego em determinados instantes que considere adequados, observe atentamente como as várias aplicações utilizam os serviços da camada inferior. Os comandos abaixo são apenas exemplos. Alguns deles podem não estar acessíveis. Explore alternativas e mesmo outras aplicações Internet bem conhecidas que considere importantes!

1. Acesso via browser ao URL: <http://marco.uminho.pt/disciplinas/CC-LEI/> (nota: pode usar comandos *wget* ou *lynx*)
2. Acesso em *ftp* para <ftp.eq.uc.pt> (Username: *ftp* Password: *cc2025* Comando: *get README / quit*). Se estiver atrás de uma firewall com NAT, deverá utilizar o modo passivo (*ftp -p ftp.eq.uc.pt*). Pense um pouco no porquê.
3. Acesso em *tftp* para cc2025.ddns.net (usando *tftp* ou *curl*, por exemplo: *curl -v tftp://cc2025.ddns.net/file1*)
4. Acesso via *telnet* para cc2025.ddns.net (Username: *cc* Password: *cc-tp1* Comando: *exit*)
5. Acesso *ssh* para cc2025.ddns.net (Username: *cc* Password: *cc-tp1*, [ssh cc@cc2025.ddns.net](ssh://cc@cc2025.ddns.net))
6. Resolução de nomes usando *nslookup* www.uminho.pt
7. *ping* www.google.pt
8. *traceroute* cc2025.ddns.net ou www.fccn.pt

3.1– 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:				

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 Virtual: abrir terminal, usar linha de comando (bash) como utilizador core e password core]

[Docker: abrir terminal com “docker exec -it core bash”, usar linha de comando (bash) já como root]

Comandos	Observações
<pre>\$ sudo mkdir -p /srv/ftp</pre> <pre>\$ sudo usermod -d /srv/ftp ftp</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 “/srv/ftp”. O comando <i>mkdir</i> criará a pasta se ela não existir (e todas as incluídas no path que forem necessárias – opção “-p”). O comando <i>usermod</i> faz dela a “home” do user “ftp”.</p> <p>NOTA: se não ocorrerem alterações, significa que já foram feitas</p>
<pre>\$ sudo sh -c “yes 'A' head -c 10K > /srv/ftp/file1”</pre> <pre>\$ sudo dd if=/dev/urandom of=/srv/ftp/file2 bs=1K count=10</pre>	<p>Depois são criados para lá dois ficheiros: o “file1”, que é um ficheiro de texto com 10 KB, e o “file2”, 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>

FTP

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

Topologia Virtual, PC1, servidor FTP
<pre>root@PC1\$ chmod a-w /srv/ftp</pre> <p>... directoria não pode ter acesso para escrita... por questões de segurança...</p> <pre>root@PC1\$ vsftpd /etc/vsftpd.conf -osecure_chroot_dir=/srv/ftp -oanonymous_enable=YES</pre> <p>... 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</p>
<p>NOTA: Estes comandos servem para configurar e <u>ativar</u> manualmente o servidor FTP na linha de comando!</p>

E depois transferir o ficheiro a partir do PCx.

Topologia Virtual, PCx, cliente FTP
<pre>root@PCx\$ ftp <endereço-ip-do-PC1></pre> <p>... entrar com username anonymous e qualquer password (aconselha-se o e-mail)</p> <pre>ftp> status</pre> <pre>ftp> pwd</pre> <pre>ftp> dir</pre> <pre>ftp> get file1</pre> <pre>ftp> quit</pre>

TFTP

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

Topologia Virtual, PC1, servidor TFTP

```
root@PC1$ chmod -R 777 /srv/ftp
... diretoria 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, PCx, 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, PCx, 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@PCx$ lynx http://<endereço-ip-do-PC1>/file2
```