

Trabalho Prático Nº.1 – Protocolos da Camada de Transporte

Duração: 2 aulas

Este trabalho deve ser realizado com recurso à máquina virtual **xubuncore** que está disponibilizada em <http://marco.uminho.pt/disciplinas/CC-MIEI/xubuncore.html> (user: *core* password: *core*)

Não siga o guião cegamente, com *copy/paste* dos comandos, sem se interrogar do que está a fazer! Tente perceber os comandos, e o que é suposto fazerem, antes de os fazer! Não perca os objetivos de vista!

Parte I: Uso da camada de transporte por parte das aplicações

Execute numa linha de comando:

```
$ sudo wireshark
```

Capturando 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-MIEI/> (nota: pode usar comandos *wget* ou *lynx*)
- Acesso em *ftp* para gr2018.ddns.net (Username: *gr2018* Password: *gr2018*)
- Acesso em *tftp* para gr2018.ddns.net (Username: *gr2018* Password: *gr2018*) (usando *tftp* ou *curl*, por exemplo: *curl tftp://gr2018.ddns.net/file1*)
- Acesso via *telnet* para gr2018.ddns.net (Username: *gr2018* Password: *gr2018*) ou para *router-ext* (193.136.9.33)
- Acesso *ssh* para gr2018.ddns.net (Username: *gr2018* Password: *gr2018*)
- Resolução de nomes usando *nslookup* www.uminho.pt
- *traceroute* cisco.di.uminho.pt ou www.fccn.pt

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

Parte II: Instalação, configuração e utilização de serviços de transferência de ficheiros

Para este exercício recomenda-se que utilize a seguinte topologia *CC-Topo-2019.imn* que se apresenta na Figura 1 - Topologia de Rede (backbone, acesso e local) e que pode descarregar da plataforma de *elearning* (<http://elearning.uminho.pt>) ou diretamente de <http://marco.uminho.pt/disciplinas/CC-MIEI/CC-Topo-2019.imn>

```
$ wget http://marco.uminho.pt/disciplinas/CC-MIEI/CC-Topo-2019.imn
```

Neste exercício pretende-se transferir o mesmo ficheiro usando 4 serviços diferentes: SFTP, FTP, TFTP e HTTP, capturando todos os pacotes trocados durante a transferência com o Wireshark. Para isso será necessário realizar os seguintes passos:

- **[Xubuncore Linux]:** verificar se o software servidor está instalado; instalar se necessário;
- **[Xubuncore Linux]:** preparar uma pasta com os ficheiros a transferir; um ficheiro de texto e um ficheiro binário;
- **[Xubuncore Linux]:** executar o *core* com a topologia virtual;
- **[Topologia virtual, Servidor1]:** configurar servidores de modo a darem acesso a essa pasta;
- **[Topologia virtual, Cliente1]:** usar o software cliente respectivo para transferir os ficheiros disponibilizados.

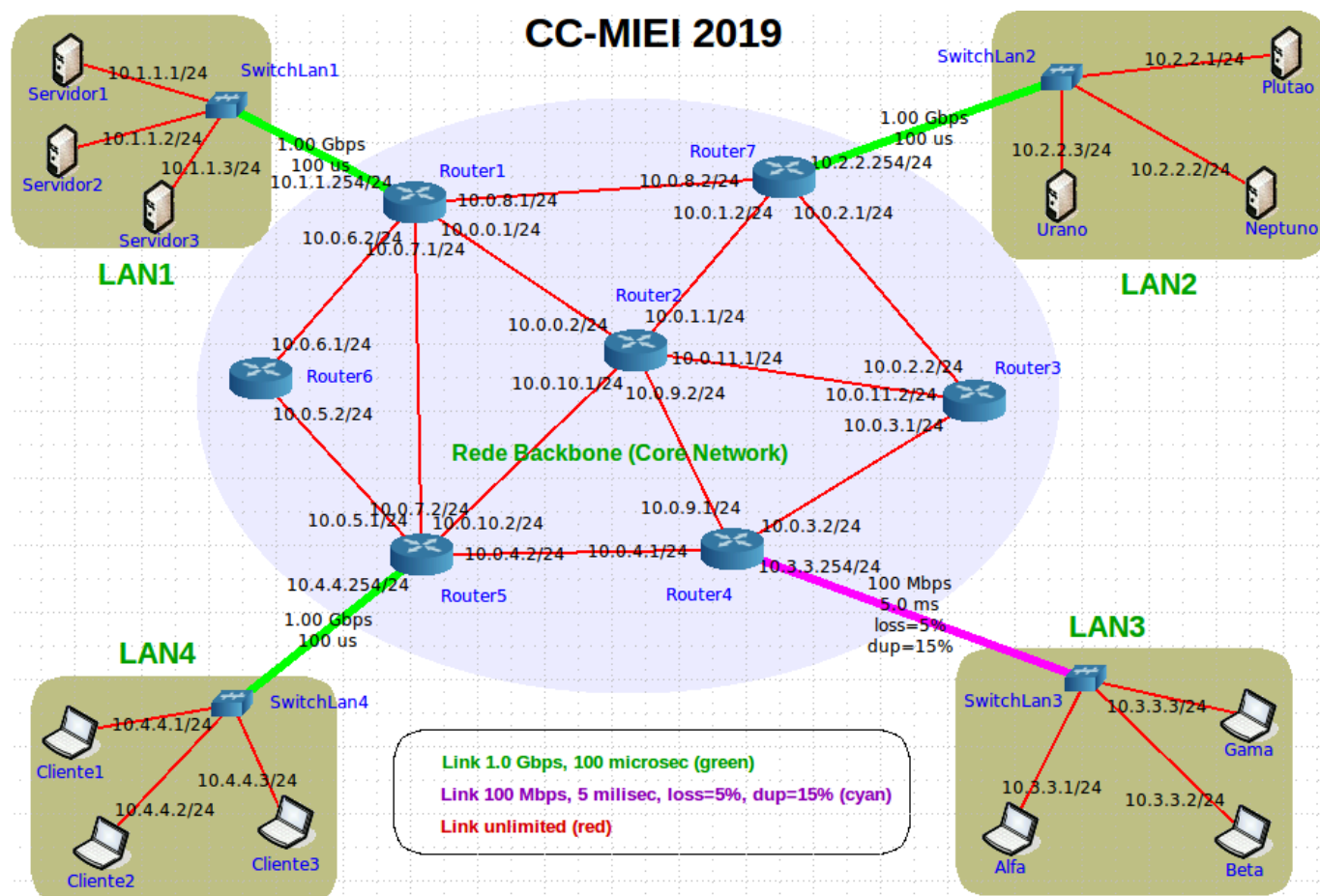


Figura 1 - Topologia de Rede (backbone, acesso e local)

| ETAPA 1: 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 |
| <pre>\$ sudo apt-get install openssh-server \$ sudo apt-get 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-get 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-get install atftpd \$ sudo apt-get 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-get install mini-httpd \$ sudo apt-get install wget \$ sudo apt-get 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! |
| <pre>\$ sudo dpkg --status openssh-server \$ sudo dpkg --status vsftpd \$ sudo dpkg --status atftpd \$ sudo dpkg --status atftp \$ sudo dpkg --status mini-httpd ...</pre> | Opcional. O comando “dpkg” é o gestor de pacotes Debian. Neste caso está a ser usado para verificar o estado dos pacotes que instalámos previamente. |

Uma vez instalado o software necessário, e ainda antes de iniciar o core com a topologia virtual, recomenda-se que prepare uma única pasta com dois ficheiros (um ficheiro de texto e um ficheiro binário) para serem disponibilizados pelos vários servidores (SFTP, FTP, TFTP e HTTP). A pasta estará acessível e visível em todos os nós da topologia virtual, pois todos partilham exatamente o mesmo sistema de ficheiros!

| ETAPA 2: preparar uma pasta com os ficheiros a transferir; um ficheiro de texto e um ficheiro binário; [Máquina Xubuncore Linux (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 cp /etc/motd /srv/ftp/file1 \$ sudo cp /bin/ls /srv/ftp/file2</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 "-p"). O comando <i>usermod</i> faz dela a "<i>home</i>" do user "<i>ftp</i>".</p> <p>Depois são copiados para lá dois ficheiros: o <i>"/etc/motd"</i> que é um ficheiro de texto e que vai ser o "<i>file1</i>" e o ficheiro executável <i>"/usr/bin/perl"</i> que será o ficheiro binário "<i>file2</i>". Pode optar por colocar ou editar outros ficheiros nessa pasta.</p> |

Podemos agora emular a topologia do ficheiro CC-Topo-2019.imn no core.

| ETAPA 3: executar o core com a topologia virtual CC-Topo-2019.imn; | |
|--|---|
| Comandos | Observações |
| <pre>\$ wget http://marco.uminho.pt/disciplinas/CC-MIEI/CC-Topo-2019.imn</pre> | Obtem o ficheiro "CC-Topo-2019.imn" da página da disciplina com o comando de linha " <i>wget</i> ". |
| <pre>\$ sudo /etc/init.d/core start \$ sudo core CC-Topo-2019.imn</pre> | Executa o <i>core</i> com a referida topologia... O primeiro comando inicia o <i>daemon</i> do <i>core</i> e o segundo comando lança a interface gráfica (cliente <i>core</i>). |
| Topologia virtual, Servidor1, botão do rato do lado direito, Shell Windows → bash. | Obter uma <i>bash shell</i> no nó Servidor1 |
| Topologia virtual, Cliente1, botão do rato do lado direito, Shell Windows → bash. | Obter uma <i>bash shell</i> no nó Cliente1 |
| Topologia virtual, Alfa, botão do rato do lado direito, Shell Windows → bash. | Obter uma <i>bash shell</i> no nó Alfa |
| Topologia virtual, Router1, botão do rato do lado direito, Wireshark → Eth3. | Lançar um processo <i>wireshark</i> no Router1 para capturar todos os pacotes que passam pelo seu interface eth3 (certifique-se de que esse é o local certo para a captura!). |

São três os objetivos fundamentais:

- i) testar a conectividade e analisar as características gerais dos *links* (ligações com diferentes larguras de banda e diferentes atrasos) utilizando o comando "*ping*" e/ou "*traceroute*";
- ii) depois transferir os ficheiros *file1* e/ou *file2* que colocámos na pasta */srv/ftp* (partilhada em todos os nós da topologia), inicialmente para o cliente *Cliente1*, capturando a transferência com o *wireshark* no router *Router1*;
- iii) comparar os tempos de transferência do ficheiro *file2* para o cliente *Cliente1* e para o cliente *Alfa*.

2.1 PING

O objetivo é testar a conectividade (mais do que 90 segundos após o *run* inicial da topologia) através do comando *ping* ao Servidor1. O comando `bash "ping -c 20 10.1.1.1 | tee file-ping-output"` realizado no Cliente1 e também no cliente Alfa. Os resultados da execução são armazenados em ficheiros com o nome "*file-ping-output*" nos nós Cliente1 e Alfa.

| Topologia Virtual, Cliente1 (ping) |
|---|
| <pre>root@Cliente1\$ ping -c 20 10.1.1.1 tee file-ping-output</pre> <p><i>Nota: ... o resultado irá ficar também armazenado em "file-ping-output"...</i></p> <pre>root@Cliente1\$ less file-ping-output</pre> |
| REPETIR (ping), na Topologia Virtual, a partir do nó Alfa na LAN3 |

2.2 SFTP

O servidor SSH já deve estar em execução no servidor Servidor1, e podemos verificar isso com os comandos "ps" e "netstat".

| Topologia Virtual, Servidor1 (servidor SFTP) |
|--|
| <pre>root@Servidor1\$ ps -ef grep ssh</pre> <p><i>... verificar se existe um processo sshd em execução...</i></p> <pre>root@Servidor1\$ netstat -n -a</pre> <p><i>... verificar se está alguém à escuta na porta 22...</i></p> |

Desta forma podemos transferir o ficheiro por **sftp**, a partir da *bash* do Cliente1

| Topologia Virtual, Cliente1 (cliente SFTP) |
|--|
| <pre>root@Cliente1\$ rm /root/.ssh/known_hosts</pre> <p><i>Nota: ... a identidade dos nós da topologia virtual está sempre a mudar... pelo que é preciso apagar a <u>lista de hosts bem conhecidos</u> do SSH</i></p> <pre>root@Cliente1\$ sftp core@10.1.1.1</pre> <p><i>Nota: ... não se esqueça que a password do utilizador "core" usado no exemplo é "core" ... se usar outro user terá que usar outra password...</i></p> <pre>sftp> pwd sftp> cd /srv/ftp sftp> dir sftp> get file1 sftp> quit</pre> |

2.3 FTP

Para transferir por **ftp** é necessário executar o servidor manualmente na *bash* do Servidor1.

Topologia Virtual, Servidor1 (servidor FTP)

```
root@Servidor1$ chmod a-w /srv/ftp
... directoria não pode ter acesso para escrita... por questões de segurança...
root@Servidor1$ vsftpd /etc/vsftpd.conf \
                    -osecure_chroot_dir=/srv/ftp -oanon_root=/srv/ftp
... atenção que a barra | serve para continuar o comando noutra linha e não é necessária se escrever tudo na mesma linha ...
```

E depois transferir o ficheiro a partir do Cliente1.

Topologia Virtual, Cliente1 (cliente FTP)

```
root@Cliente1$ ftp 10.1.1.1
... entrar com username anonymous e qualquer password (aconselha-se o e-mail)
ftp> status
ftp> pwd
ftp> dir
ftp> get file1
ftp> quit
```

2.4 TFTP

Para transferir por **tftp** é necessário preparar a directoria e executar o servidor manualmente na *bash* do Servidor1.

Topologia Virtual, Servidor1 (servidor TFTP)

```
root@Servidor1$ chmod -R 777 /srv/ftp
... directoria tem de ter acesso para escrita para todos...
root@Servidor1$ touch atftpd.log
... se houver problemas podemos ver neste ficheiro de log o que se passou...
root@Servidor1$ atftpd --verbose=3 --user root.ftp --logfile atftpd.log \
                    --bind-address 10.1.1.1 --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 ...
```

E depois transferir o ficheiro a partir do Cliente1:

Topologia Virtual, Cliente1 (cliente TFTP)

```
root@Cliente1$ atftp 10.1.1.1
ftp> status
ftp> get file1
ftp> quit
```

2.5 HTTP

Para transferir por **http** é necessário preparar a directoria e executar o servidor manualmente na *bash* do Servidor1.

Topologia Virtual, Servidor1 (servidor HTTP)

```
root@Servidor1$ mini-httpd -d /srv/ftp/  
...  
root@Servidor1$ ps -ef  
... para verificar se o daemon ficou em execução...
```

E depois transferir o ficheiro a partir do Cliente1.

Topologia Virtual, Cliente1 (cliente HTTP)

```
root@Cliente1$ wget http://10.1.1.1/file1  
root@Cliente1$ wget http://10.1.1.1/file2  
...  
Ou com o comando lynx:  
  
root@Cliente1$ lynx http://10.1.1.1/file1  
root@Cliente1$ lynx http://10.1.1.1/file2
```

NOTA: A submissão do trabalho deve seguir as regras definidas pelo docente do turno ou em documento complementar.