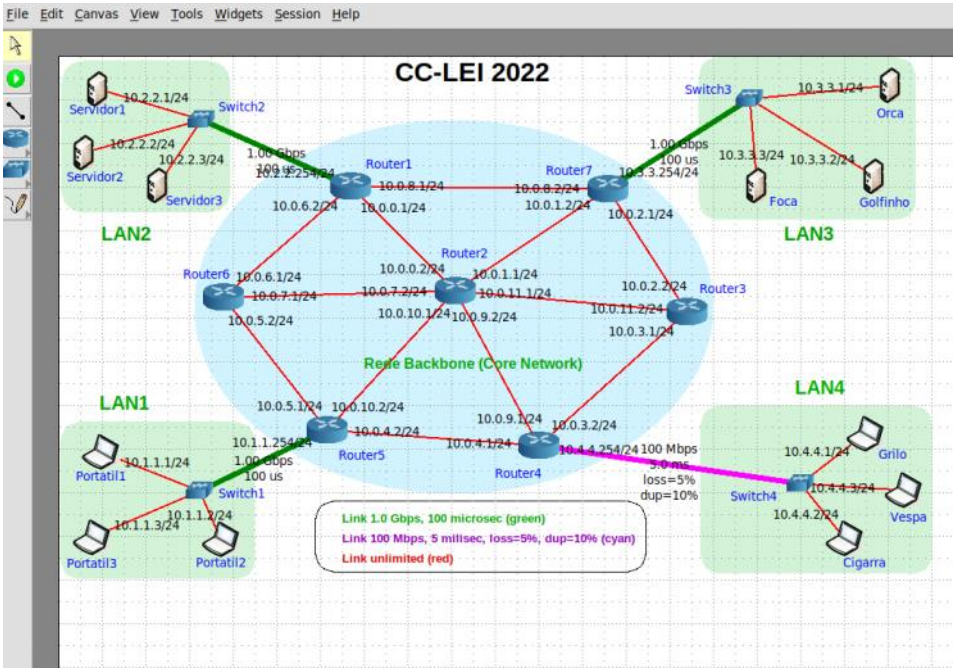
CC - TP1

18 de setembro de 2023 11:24

cc-Topo-2023.imn

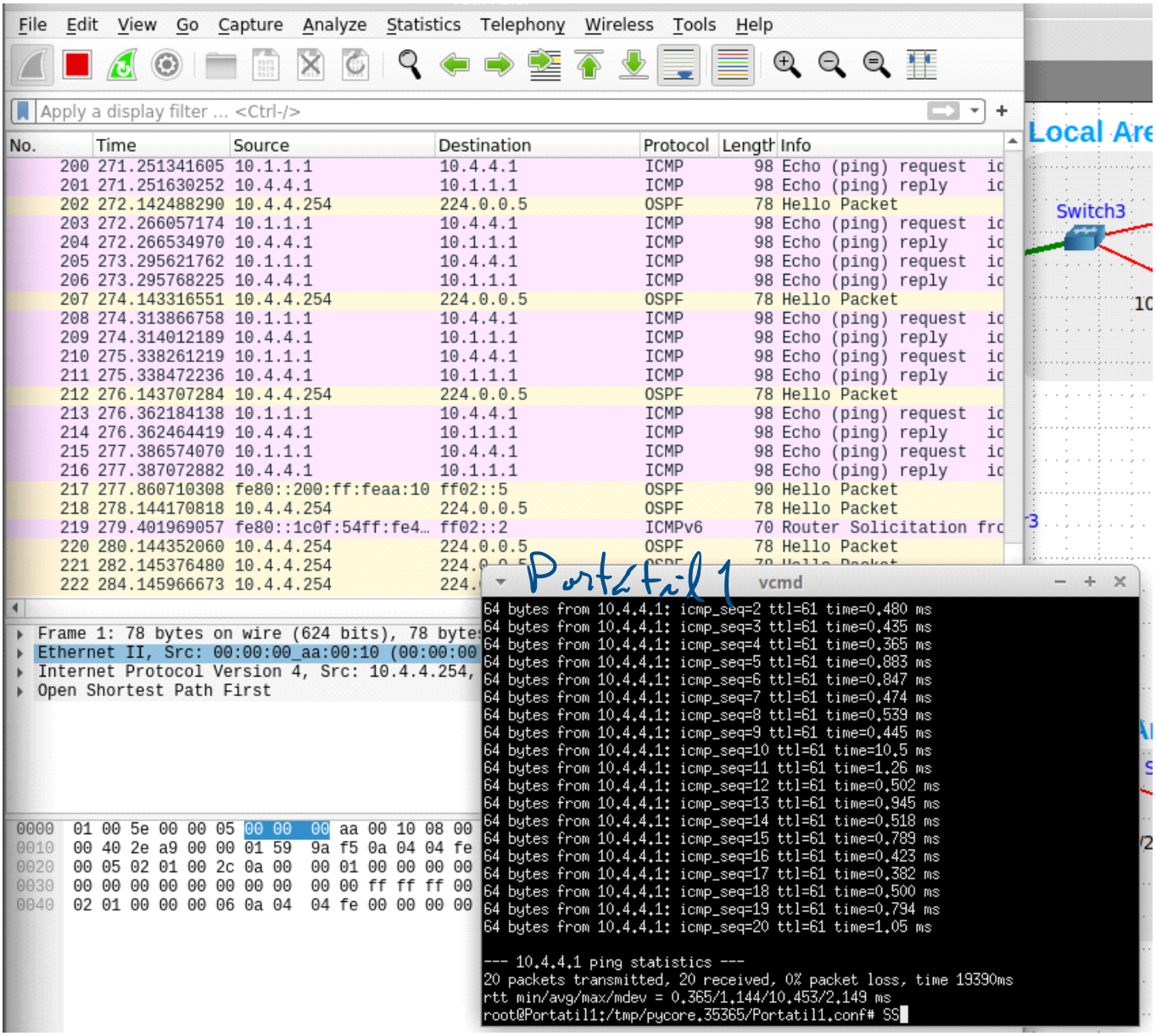


São três os objetivos fundamentais:

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 `Portatil1`, capturando a transferência com o *wireshark* no router `Router1`;

iii) comparar os tempos de transferência do ficheiro `file2` para o cliente `Portatil1` e para o cliente `PC1`.



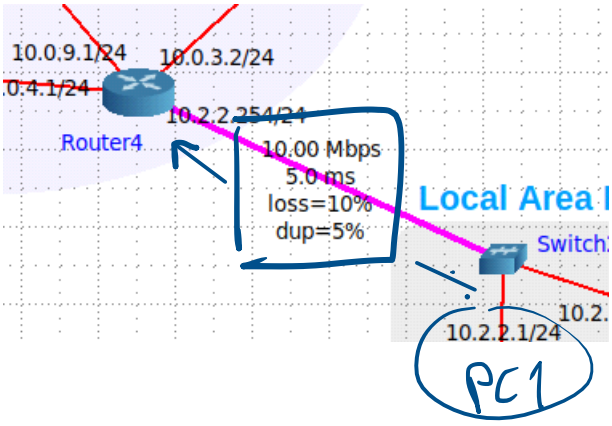
PC 1

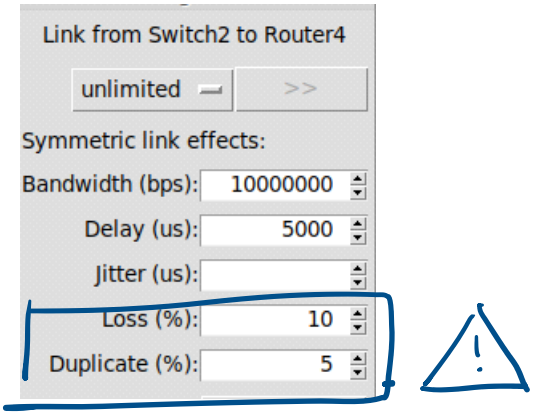
```
rtt min/avg/max/mdev = 5.373/12.056/22.048/6.517 ms
<.conf# ping -c 20 10.4.4.1 | tee file-ping-output-2
PING 10.4.4.1 (10.4.4.1) 56(84) bytes of data.
64 bytes from 10.4.4.1: icmp_seq=1 ttl=61 time=6.09 ms
64 bytes from 10.4.4.1: icmp_seq=2 ttl=61 time=7.79 ms
64 bytes from 10.4.4.1: icmp_seq=3 ttl=61 time=10.2 ms
64 bytes from 10.4.4.1: icmp_seq=4 ttl=61 time=16.1 ms
64 bytes from 10.4.4.1: icmp_seq=5 ttl=61 time=9.12 ms
64 bytes from 10.4.4.1: icmp_seq=6 ttl=61 time=9.62 ms
64 bytes from 10.4.4.1: icmp_seq=7 ttl=61 time=8.09 ms
64 bytes from 10.4.4.1: icmp_seq=8 ttl=61 time=13.8 ms
64 bytes from 10.4.4.1: icmp_seq=10 ttl=61 time=18.1 ms
64 bytes from 10.4.4.1: icmp_seq=12 ttl=61 time=23.8 ms
64 bytes from 10.4.4.1: icmp_seq=13 ttl=61 time=6.38 ms
64 bytes from 10.4.4.1: icmp_seq=16 ttl=61 time=26.3 ms
64 bytes from 10.4.4.1: icmp_seq=17 ttl=61 time=11.8 ms
64 bytes from 10.4.4.1: icmp_seq=18 ttl=61 time=13.9 ms
64 bytes from 10.4.4.1: icmp_seq=19 ttl=61 time=9.24 ms
64 bytes from 10.4.4.1: icmp_seq=20 ttl=61 time=5.89 ms
64 bytes from 10.4.4.1: icmp_seq=20 ttl=61 time=5.90 ms (DUP!)

--- 10.4.4.1 ping statistics ---
20 packets transmitted, 16 received, +1 duplicates, 20% packet loss, time 19115m
s
rtt min/avg/max/mdev = 5.893/11.892/26.321/5.963 ms
root@PC1:/tmp/pycore.35365/PC1.conf#
```

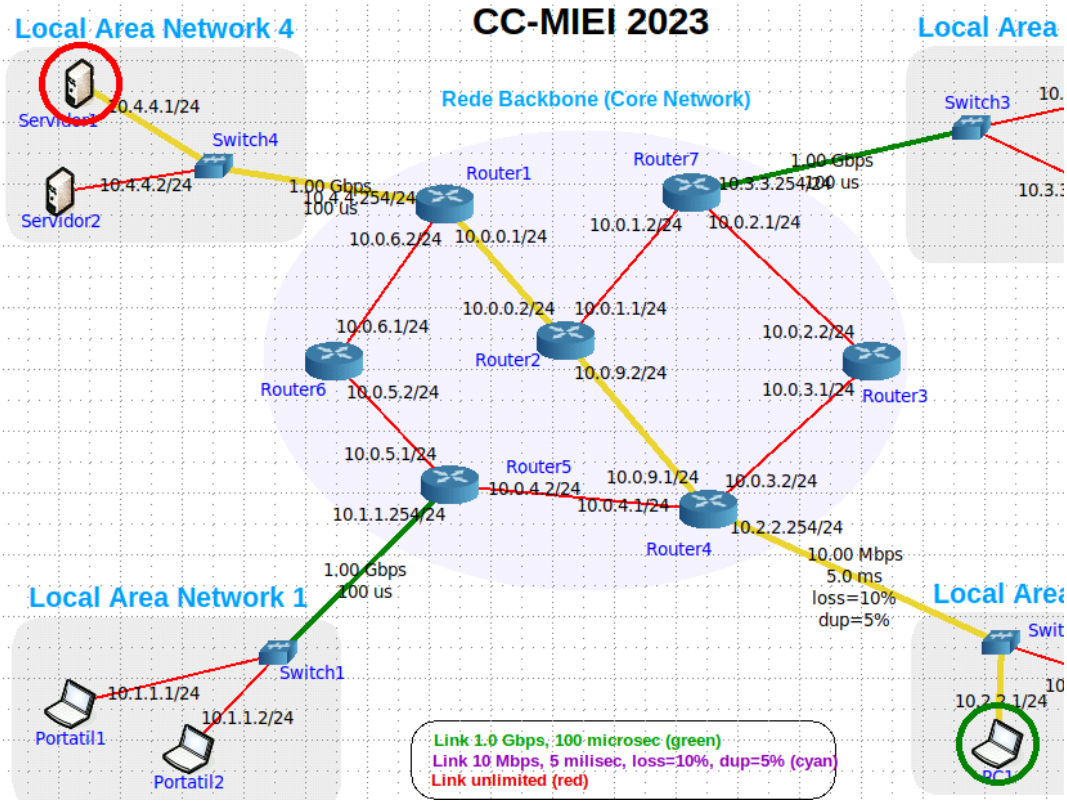
1 duplicado

20% packet loss

20% packet loss



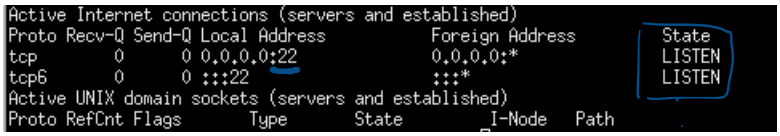





```
root@Servidor1:/tmp/pycore.35365/Servidor1.conf# ps -ef | grep ssh
root          32       1  0 11:57 ?        00:00:00 sshd: /usr/sbin/sshd -f /etc
/ssh/sshd_config [listener] 0 of 10-100 startups
root          41      33  0 12:26 pts/2    00:00:00 grep --color=auto ssh
```

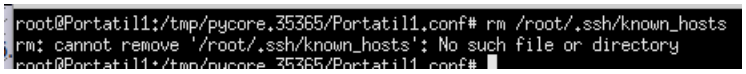
Processo SSH em execução

1. **SSH (Secure Shell)**:
   - O SSH é o protocolo de rede seguro usado para estabelecer conexões criptografadas entre computadores em uma rede.
   - O SSH é usado principalmente pelos clientes SSH para iniciar sessões seguras com servidores remotos.
   - Quando você executa um comando SSH (por exemplo, `` `ssh username@hostname` ``), você está iniciando uma conexão SSH do cliente SSH para o servidor SSHD.
2. **SSHD (Secure Shell Daemon)**:
   - O SSHD, por outro lado, é o servidor SSH que fica em um sistema remoto e recebe conexões SSH entrantes.
   - O SSHD é responsável por autenticar usuários, gerenciar sessões SSH, verificar permissões de acesso, registrar eventos de conexão e aplicar políticas de segurança no servidor remoto.
   - Ele executa em segundo plano no servidor e fica esperando por conexões SSH.

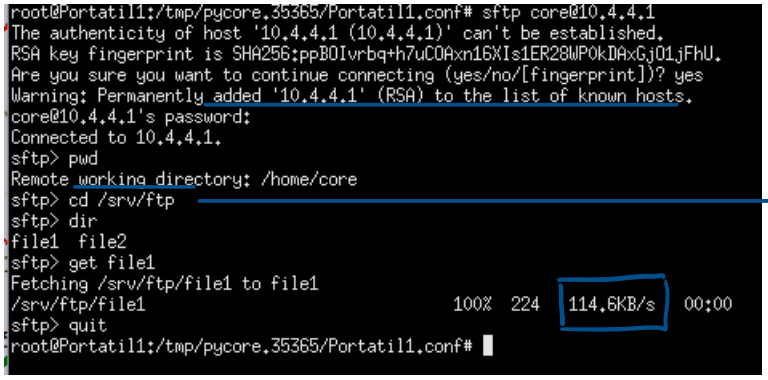


Porta 22

SFTP, que significa "SSH File Transfer Protocol" (Protocolo de Transferência de Arquivos SSH), é um protocolo seguro usado para transferir arquivos e gerenciar sistemas de arquivos remotamente por meio de uma conexão SSH (Secure Shell). Ele fornece uma maneira segura e criptografada de transferir dados entre sistemas, tornando-o uma alternativa segura ao FTP (File Transfer Protocol) e ao FTPS (FTP Secure).



• SFTP



Comandos	Observações

`$ sudo mkdir -p /srv/ftp`
 	
`$ sudo usermod -d /srv/ftp ftp`
 	
`$ sudo cp /etc/hosts /srv/ftp/file1`
 	
`$ sudo cp /bin/ls /srv/ftp/file2`
	O servidor FTP instala um novo utilizador no sistema com *username* "ftp" sem password para poder servir ficheiros da *home* desse utilizador de forma anónima a qualquer cliente FTP. A pasta a criar chama-se *"/srv/ftp"*. O comando *mkdir* criará a pasta se ela não existir (e todas as incluídas no path que forem necessárias – opção "-p"). O comando *usermod* faz dela a "*home*" do user "*ftp*".
	Depois são copiados para lá dois ficheiros: o *"/etc/hosts"* que é um ficheiro de texto pequeno e que vai ser o "*file1*" e o ficheiro executável *"/bin/ls"* que será o ficheiro binário (executável) "*file2*". Pode optar por colocar ou editar outros ficheiros nessa pasta. Tudo o que estiver lá ficará acessível.

```
root@PC1:/tmp/pycore.41989/PC1.conf# rm /root/.ssh/known_hosts
root@PC1:/tmp/pycore.41989/PC1.conf# sftp core@10.4.4.1
The authenticity of host '10.4.4.1 (10.4.4.1)' can't be established.
RSA key fingerprint is SHA256:DgwWmfmOrOVbq7ipET1xWyCtOMDp+OuhVkDucQzLpeO.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.4.4.1' (RSA) to the list of known hosts.
core@10.4.4.1's password:
Connected to 10.4.4.1.
sftp> pwd
Remote working directory: /home/core
sftp> cd /srv
sftp> cd /srv/ft
sftp> cd /srv/ftp/
sftp> dir
file1  file2
sftp> get file1
sftp> get file1
Fetching /srv/ftp/file1 to file1
/srv/ftp/file1                                100%  224     9.0KB/s   00:00
sftp> quit
root@PC1:/tmp/pycore.41989/PC1.conf#
```

114.6 KB/s (VS) 9.0 KB/s

- FTP (File Transfer Protocol)

```
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# chmod a-w /srv/ftp
```

a → all
w → write
Remove permissão de escrita a todos os utilizadores na diretoria /srv/ftp

⇒ Ninguém pode escrever na pasta

configuração anonymous FTP

anonymous FTP access is allowed

configuração

VSFPD Server

Users restricted to that root

Execução em background

-o → configuration option

— Transferir o ficheiro a partir do Portátil 1

```
NAME
     vsftpd - Very Secure FTP Daemon

SYNOPSIS
     vsftpd [configuration file and / or options]

DESCRIPTION
     vsftpd is the Very Secure File Transfer Protocol Daemon. The server can
     be launched via a "super-server" such as inetd(8) or xinetd(8).  Alterna-
     tively, vsftpd can be launched in standalone mode, in which case vsftpd
     itself will listen on the network. This latter mode is easier to use, and
     recommended. It is activated by setting listen=YES in /etc/vsftpd.conf.
     Direct execution of the vsftpd binary will then launch the FTP service
     ready for immediate client connections.

OPTIONS
     An optional configuration file or files may be given on the command line.
     These files must be owned as root if running as root. Any command line
     option not starting with a "-" character is treated as a config file that
     will be loaded. Note that config files are loaded in the strict order
     that they are encountered on the command line.  If no config files are
```

```
root@Portatil1:/tmp/pycore.41989/Portatil1.conf# ftp 10.4.4.1
Connected to 10.4.4.1.
220 (vsFTPd 3.0.3)
Name (10.4.4.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> status
Connected to 10.4.4.1.
No proxy connection.
Connecting using address family: any.
Mode: stream; Type: binary; Form: non-print; Structure: file
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Quote control characters: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
Tick counter printing: off
ftp> pwd
257 "/" is the current directory
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             224 Sep 18 11:45 file1
-rwxr-xr-x    1 0        0          142144 Sep 18 11:45 file2
226 Directory send OK.
ftp> get file1
local: file1 remote: file1
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for file1 (224 bytes).
226 Transfer complete.
224 bytes received in 0.00 secs (3.5604 MB/s)
ftp> quit
221 Goodbye.
root@Portatil1:/tmp/pycore.41989/Portatil1.conf#
```

— Transferência a partir do PC1

```
root@PC1:/tmp/pycore.41989/PC1.conf# ftp 10.4.4.1
Connected to 10.4.4.1.
220 (vsFTPd 3.0.3)
Name (10.4.4.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> status
Connected to 10.4.4.1.
No proxy connection.
Connecting using address family: any.
Mode: stream; Type: binary; Form: non-print; Structure: file
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Quote control characters: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
Tick counter printing: off
ftp> pwd
257 "/" is the current directory
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             224 Sep 18 11:45 file1
-rwxr-xr-x    1 0        0          142144 Sep 18 11:45 file2
226 Directory send OK.
ftp> get file1
local: file1 remote: file1
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for file1 (224 bytes).
226 Transfer complete.
224 bytes received in 0.00 secs (623.2194 kB/s)
ftp> quit
221 Goodbye.
root@PC1:/tmp/pycore.41989/PC1.conf#
```

- TFTP (Trivial File Transfer Protocol)
  - ↳ Versão simplificada do FTP
  - Menos seguro

```
<tftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.4.4.1 --daemon --no-fork /srv/ftp
^Z
[2]+  Stopped                 atftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.4.4.1 --daemon --no-fork /srv/ftp
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# bg
[2]+ atftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.4.4.1 --daemon --no-fork /srv/ftp &
root@Servidor1:/tmp/pycore.41989/Servidor1.conf#
```

↑ Ativação manual do servidor TFTP

Portátil 1

```
root@Portatil1:/tmp/pycore.41989/Portatil1.conf# atftp 10.4.4.1
tftp> status
Connected:  10.4.4.1 port 69
Mode:       octet
Verbose:    off
Trace:      off
Options
 tsize:     disabled
 blksize:   disabled
 timeout:   disabled
 multicast: disabled
mtftp variables
 client-port:   76
 mcast-ip:      0.0.0.0
 listen-delay:  2
 timeout-delay: 2
Last command: quit
tftp> get file1
Overwite local file [y/n]? y
tftp> quit
root@Portatil1:/tmp/pycore.41989/Portatil1.conf#
```

PC1

```
root@PC1:/tmp/pycore.41989/PC1.conf# atftp 10.4.4.1
tftp> status
Connected:  10.4.4.1 port 69
Mode:       octet
Verbose:    off
Trace:      off
Options
 tsize:     disabled
 blksize:   disabled
 timeout:   disabled
 multicast: disabled
mtftp variables
 client-port:   76
 mcast-ip:      0.0.0.0
 listen-delay:  2
 timeout-delay: 2
Last command: ---
tftp> get file1
Overwite local file [y/n]? y
tftp> quit
root@PC1:/tmp/pycore.41989/PC1.conf#
```

Não percebi as diferenças.

- HTTP

```
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# mini_httpd: started as root wit
hout requesting chroot(), warning only
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 17:01 ?        00:00:00 vnoded -v -c /tmp/pycore.41989/Servidor1 -l
root          32       1  0 17:01 ?        00:00:00 sshd: /usr/sbin/sshd -f /etc/ssh/sshd_config
root          33       1  0 17:01 pts/2    00:00:00 /bin/bash
root          44      33  0 17:03 pts/2    00:00:00 atftpd --verbose=3 --user root ftp --logfile
nobody        64       1  0 17:07 ?        00:00:00 mini_httpd -d /srv/ftp
root          65      33  0 17:08 pts/2    00:00:00 ps -ef
root@Servidor1:/tmp/pycore.41989/Servidor1.conf#
```

Em execução

— Portátil 1

```
root@Portatil1:/tmp/pycore.41989/Portatil1.conf# wget http://10.4.4.1/file1
--2023-09-18 17:09:42--  http://10.4.4.1/file1
Connecting to 10.4.4.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 224 [text/plain]
Saving to: 'file1.1'

file1.1             100%[===================>]     224  --.-KB/s    in 0s

2023-09-18 17:09:42 (70,1 MB/s) - 'file1.1' saved [224/224]
```

```
root@Portatil1:/tmp/pycore.41989/Portatil1.conf# wget http://10.4.4.1/file2
--2023-09-18 17:10:41--  http://10.4.4.1/file2
Connecting to 10.4.4.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 142144 (139K) [text/plain]
Saving to: 'file2'

file2               100%[===================>] 138,81K  --.-KB/s    in 0,02s

2023-09-18 17:10:41 (7,25 MB/s) - 'file2' saved [142144/142144]
```

200 OK = Successful Request
- Content included
- Succesful outcome