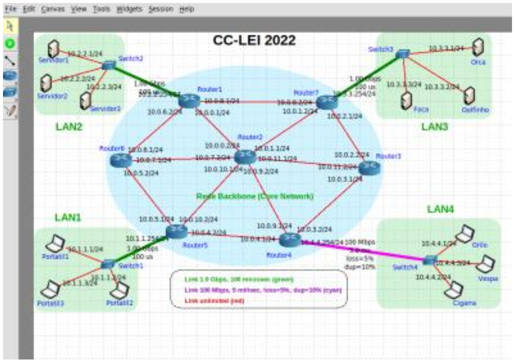
CC - TP1

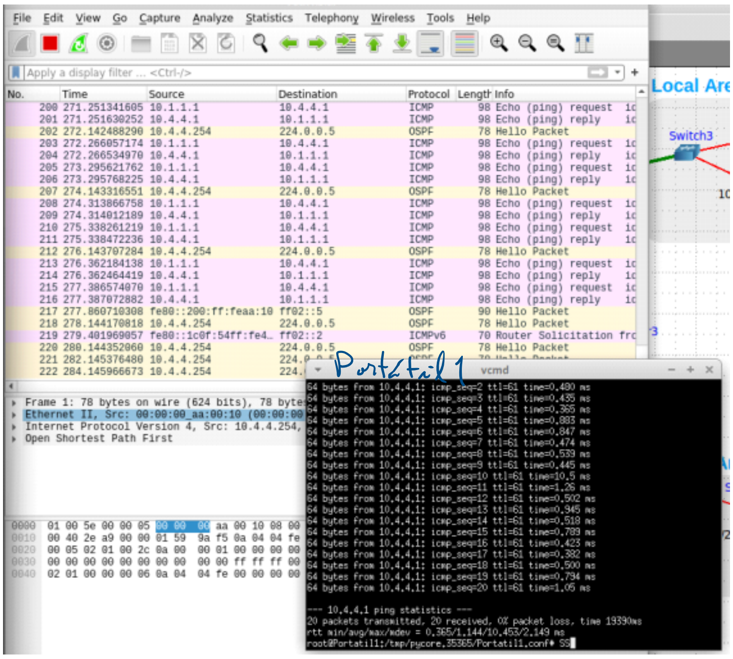
18 de setembro de 2023 11:24

cc - Topo - 2023.imn



São três os objetivos fundamentais:

- **i)** testar a conectividade e analisar as caracteristicas 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`.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Local Are
Switch3

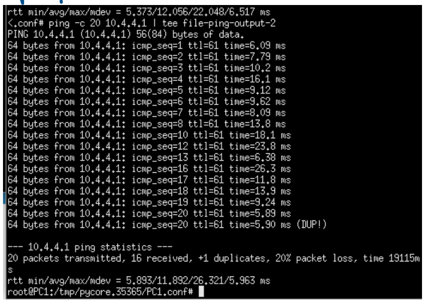
```
No.  Time           Source                   Destination  Protocol Length Info
 200 271.251341605 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 201 271.251630252 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 202 272.142488290 10.4.4.254                224.0.0.5    OSPF       78 Hello Packet
 203 272.266057174 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 204 272.266534970 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 205 273.295621762 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 206 273.295768225 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 207 274.143316551 10.4.4.254                224.0.0.5    OSPF       78 Hello Packet
 208 274.313866758 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 209 274.314012189 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 210 275.338261219 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 211 275.338472236 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 212 276.143707284 10.4.4.254                224.0.0.5    OSPF       78 Hello Packet
 213 276.362184138 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 214 276.362464419 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 215 277.386574070 10.1.1.1                  10.4.4.1     ICMP       98 Echo (ping) request  id
 216 277.387072882 10.4.4.1                  10.1.1.1     ICMP       98 Echo (ping) reply    id
 217 277.860710308 fe80::200:ff:feaa:10      ff02::5      OSPF       90 Hello Packet
 218 278.144170818 10.4.4.254                224.0.0.5    OSPF       78 Hello Packet
 219 279.401969057 fe80::1c0f:54ff:fe4…      ff02::2      ICMPv6     70 Router Solicitation fro
 220 280.144352060 10.4.4.254                224.0.0.5    OSPF       78 Hello Packet
 221 282.145376480 10.4.4.254                224.
 222 284.145966673 10.4.4.254                224.

▸ Frame 1: 78 bytes on wire (624 bits), 78 byte
▸ Ethernet II, Src: 00:00:00_aa:00:10 (00:00:00
▸ Internet Protocol Version 4, Src: 10.4.4.254,
▸ Open Shortest Path First

0000  01 00 5e 00 00 05 00 00  00 aa 00 10 08 00
0010  00 40 2e a9 00 00 01 59  9a f5 0a 04 04 fe
0020  00 05 02 01 00 2c 0a 00  00 01 00 00 00 00
0030  00 00 00 00 00 00 00 00  00 00 ff ff ff 00
0040  02 01 00 00 00 06 0a 04  04 fe 00 00 00 00
```

Portátil 1

vcmd

```
64 bytes from 10.4.4.1: icmp_seq=2 ttl=61 time=0.480 ms
64 bytes from 10.4.4.1: icmp_seq=3 ttl=61 time=0.435 ms
64 bytes from 10.4.4.1: icmp_seq=4 ttl=61 time=0.365 ms
64 bytes from 10.4.4.1: icmp_seq=5 ttl=61 time=0.883 ms
64 bytes from 10.4.4.1: icmp_seq=6 ttl=61 time=0.847 ms
64 bytes from 10.4.4.1: icmp_seq=7 ttl=61 time=0.474 ms
64 bytes from 10.4.4.1: icmp_seq=8 ttl=61 time=0.539 ms
64 bytes from 10.4.4.1: icmp_seq=9 ttl=61 time=0.445 ms
64 bytes from 10.4.4.1: icmp_seq=10 ttl=61 time=10.5 ms
64 bytes from 10.4.4.1: icmp_seq=11 ttl=61 time=1.26 ms
64 bytes from 10.4.4.1: icmp_seq=12 ttl=61 time=0.502 ms
64 bytes from 10.4.4.1: icmp_seq=13 ttl=61 time=0.945 ms
64 bytes from 10.4.4.1: icmp_seq=14 ttl=61 time=0.518 ms
64 bytes from 10.4.4.1: icmp_seq=15 ttl=61 time=0.789 ms
64 bytes from 10.4.4.1: icmp_seq=16 ttl=61 time=0.423 ms
64 bytes from 10.4.4.1: icmp_seq=17 ttl=61 time=0.382 ms
64 bytes from 10.4.4.1: icmp_seq=18 ttl=61 time=0.500 ms
64 bytes from 10.4.4.1: icmp_seq=19 ttl=61 time=0.794 ms
64 bytes from 10.4.4.1: icmp_seq=20 ttl=61 time=1.05 ms

--- 10.4.4.1 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19390ms
rtt min/avg/max/mdev = 0.365/1.144/10.453/2.149 ms
root@Portatil1:/tmp/pycore.35365/Portatil1.conf# SS
```

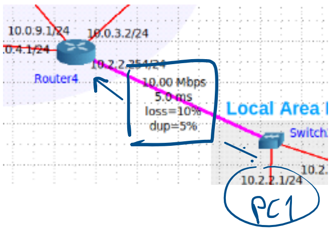
PC 1

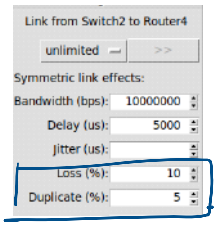


```
rtt min/avg/max/mdev = 5.373/12.056/22.048/6.517 ms
<.conf# ping -c 20 10.4.4.1 | tee file-ping-output-2
PING 10.4.4.1 (10.4.4.1) 56(84) bytes of data.
64 bytes from 10.4.4.1: icmp_seq=1 ttl=61 time=6.09 ms
64 bytes from 10.4.4.1: icmp_seq=2 ttl=61 time=7.79 ms
64 bytes from 10.4.4.1: icmp_seq=3 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





Link from Switch2 to Router4

unlimited ▭ | >>

Symmetric link effects:

Bandwidth (bps): 10000000

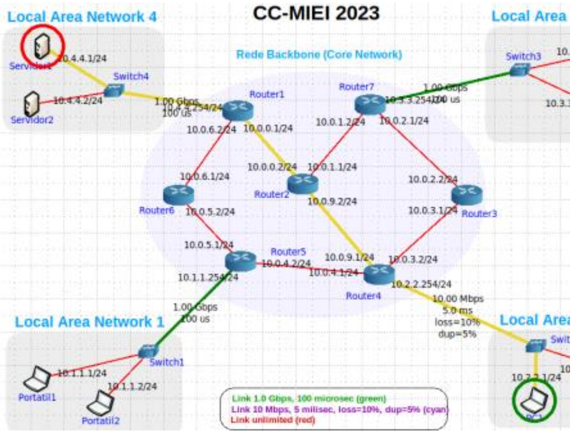
Delay (us): 5000

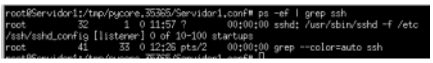
Jitter (us):

Loss (%): 10

Duplicate (%): 5

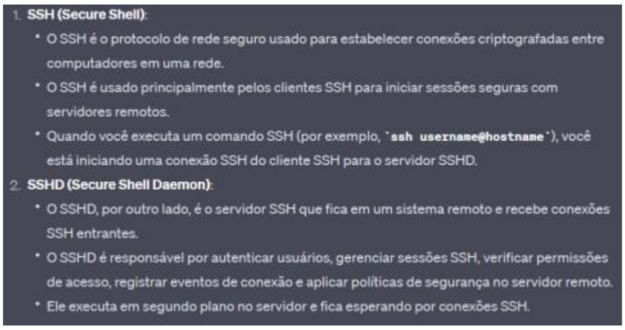
⚠



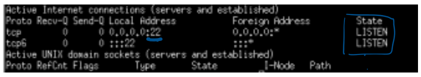


```
root@Servidor1:/tmp/pycore.35365/Servidor1.conf# ps -ef | grep ssh
root          32       1  0 11:57 ?        00:00:00 sshd: /usr/sbin/sshd -f /etc
/ssh/sshd_config [listener] 0 of 10-100 startups
root          41      33  0 12:26 pts/2    00:00:00 grep --color=auto ssh
```

Processo SSH em execução

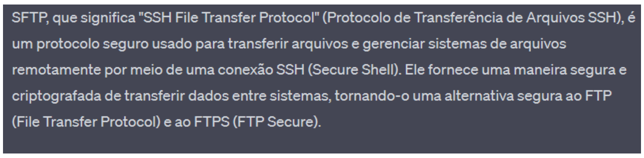


1. **SSH (Secure Shell)**:
   - O SSH é o protocolo de rede seguro usado para estabelecer conexões criptografadas entre computadores em uma rede.
   - O SSH é usado principalmente pelos clientes SSH para iniciar sessões seguras com servidores remotos.
   - Quando você executa um comando SSH (por exemplo, `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.

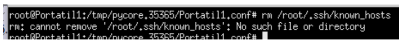


```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
```

Porta 22

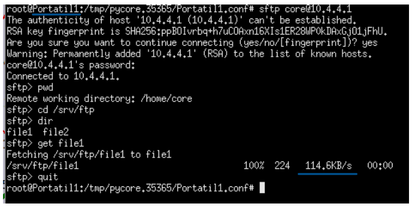


SFTP, que significa "SSH File Transfer Protocol" (Protocolo de Transferência de Arquivos SSH), é um protocolo seguro usado para transferir arquivos e gerenciar sistemas de arquivos remotamente por meio de uma conexão SSH (Secure Shell). Ele fornece uma maneira segura e criptografada de transferir dados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

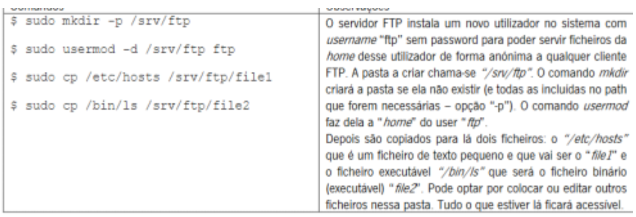


```
root@Portatil1:/tmp/pycore.35365/Portatil1.conf# rm /root/.ssh/known_hosts
rm: cannot remove '/root/.ssh/known_hosts': No such file or directory
root@Portatil1:/tmp/pycore.35365/Portatil1.conf# 
```

• SFTP

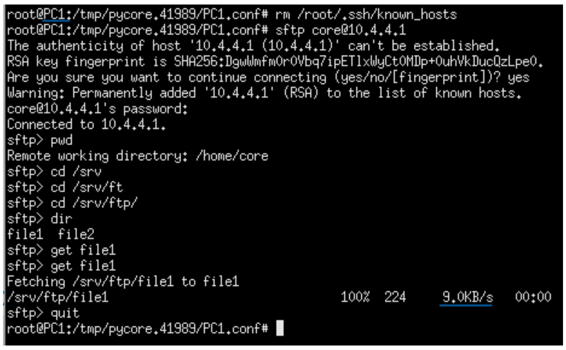


```
root@Portatil1:/tmp/pycore.35365/Portatil1.conf# sftp core@10.4.4.1
The authenticity of host '10.4.4.1 (10.4.4.1)' can't be established.
RSA key fingerprint is SHA256:ppBOIvrbq+h7uCOAxn16XIs1ER28WPOkDAxGjO1jFhU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently 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/ftp
sftp> dir
file1  file2
sftp> get file1
Fetching /srv/ftp/file1 to file1
/srv/ftp/file1                                100%  224   114.6KB/s   00:00
sftp> quit
root@Portatil1:/tmp/pycore.35365/Portatil1.conf# 
```



Comandos	Observações

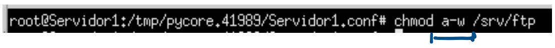
`$ sudo mkdir -p /srv/ftp`
`$ sudo usermod -d /srv/ftp ftp`
`$ sudo cp /etc/hosts /srv/ftp/file1`
`$ sudo cp /bin/ls /srv/ftp/file2`
	O servidor FTP instala um novo utilizador no sistema com *username* "ftp" sem password para poder servir ficheiros da *home* desse utilizador de forma anónima a qualquer cliente FTP. A pasta a criar chama-se *"/srv/ftp"*. O comando *mkdir* criará a pasta se ela não existir (e todas as incluidas 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:DgwWmfm0r0Vbq7ipETlxWyCt0MDp+0uhVkDucQzLpe0.
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)



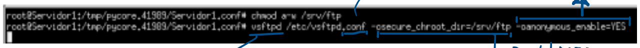
```
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# chmod a-w /srv/ftp
```

a → all

w → write

Remove permissões de escrita a todos os utilizadores na diretoria /srv/ftp

⇒ Ninguém pode escrever na pasta



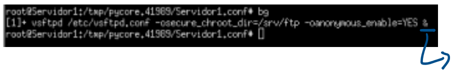
```
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# chmod a-w /srv/ftp
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# vsftpd /etc/vsftpd.conf -osecure_chroot_dir=/srv/ftp -oanonymous_enable=YES
█
```

configuração

anonymous FTP access is allowed

VSFPD Server

Users restricted to that root

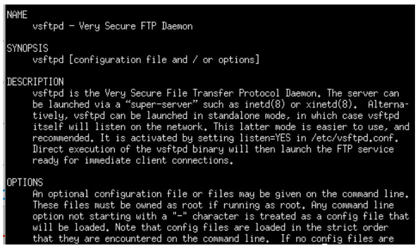


```
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# bg
[1]+ vsftpd /etc/vsftpd.conf -osecure_chroot_dir=/srv/ftp -oanonymous_enable=YES &
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# []
```

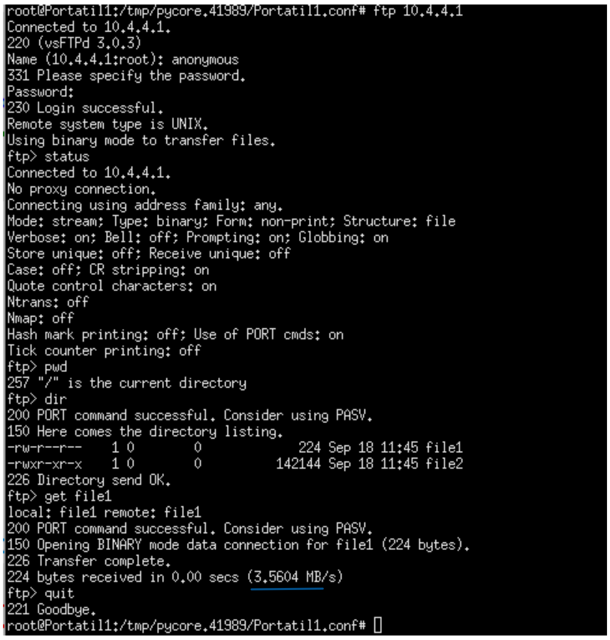
Execução em background

-O → configuration option

— Transferir o ficheiro a partir do Portátil 1

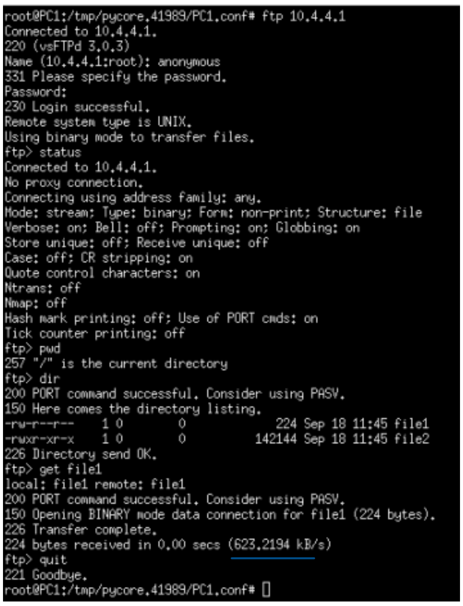


```
NAME
     vsftpd - Very Secure FTP Daemon

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
```



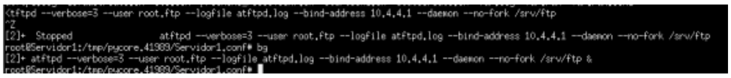
```
root@Portatil1:/tmp/pycore.41989/Portatil1.conf# ftp 10.4.4.1
Connected to 10.4.4.1.
220 (vsFTPd 3.0.3)
Name (10.4.4.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> status
Connected to 10.4.4.1.
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



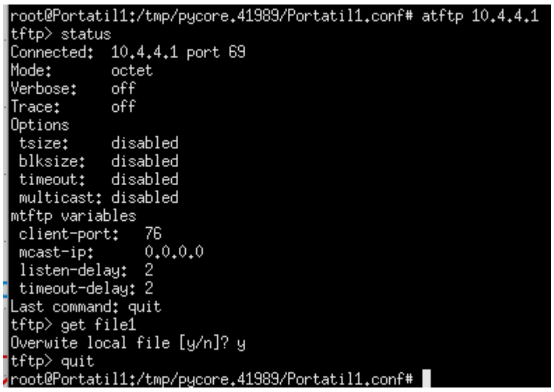
```
root@PC1:/tmp/pycore.41989/PC1.conf# ftp 10.4.4.1
Connected to 10.4.4.1.
220 (vsFTPd 3.0.3)
Name (10.4.4.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> status
Connected to 10.4.4.1.
No proxy 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



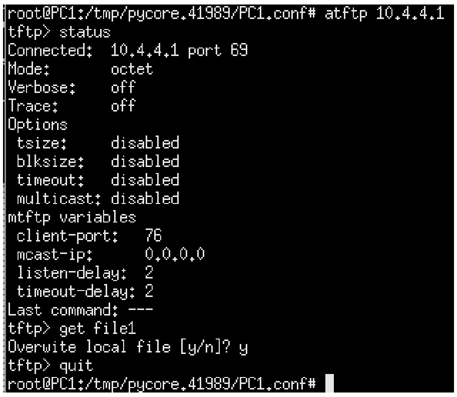
```
<tftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.4.4.1 --daemon --no-fork /srv/ftp
^Z
[2]+  Stopped                 atftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.4.4.1 --daemon --no-fork /srv/ftp
root@Servidor1:/tmp/pycore.41989/Servidor1.conf#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

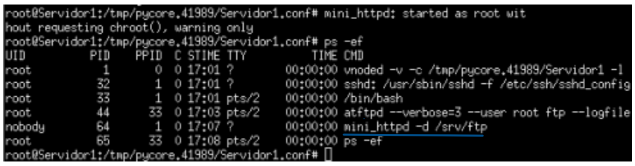
PC1



```
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.

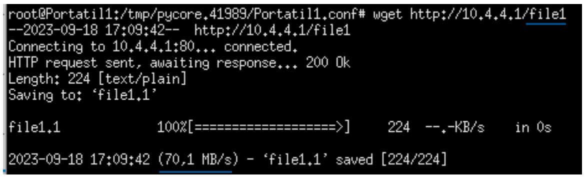
- HTTP



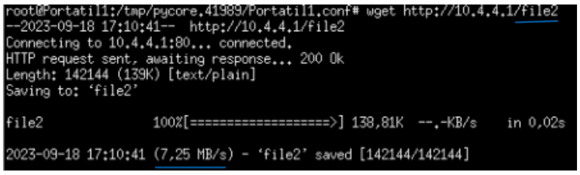
```
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# mini_httpd: started as root wit
hout requesting chroot(), warning only
root@Servidor1:/tmp/pycore.41989/Servidor1.conf# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 17:01 ?        00:00:00 vnoded -v -c /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

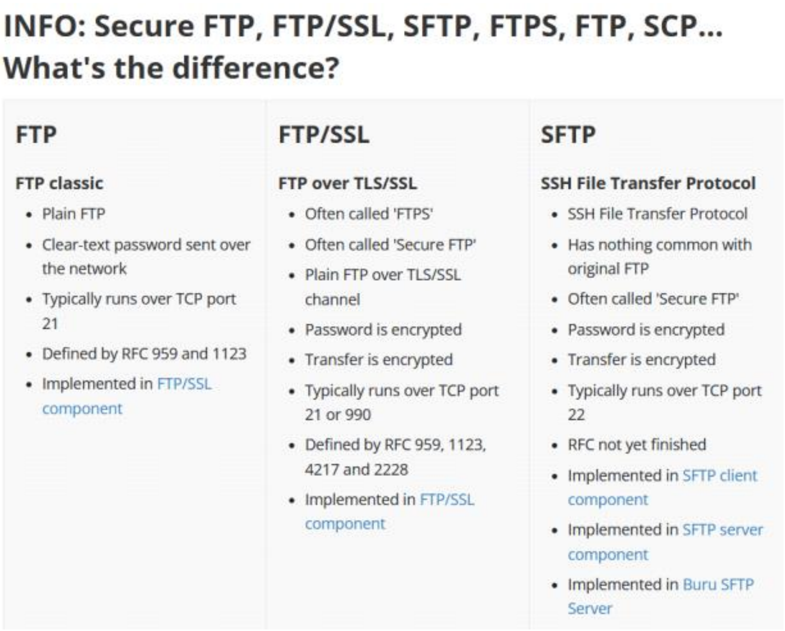


```
root@Portatil1:/tmp/pycore.41989/Portatil1.conf# wget http://10.4.4.1/file1
--2023-09-18 17:09:42--  http://10.4.4.1/file1
Connecting to 10.4.4.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 224 [text/plain]
Saving to: ‘file1.1’

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



INFO: Secure FTP, FTP/SSL, SFTP, FTPS, FTP, SCP... What's the difference?

FTP

FTP classic

- Plain FTP
- Clear-text password sent over the network
- Typically runs over TCP port 21
- Defined by RFC 959 and 1123
- Implemented in FTP/SSL component

FTP/SSL

FTP over TLS/SSL

- Often called 'FTPS'
- Often called 'Secure FTP'
- Plain FTP over TLS/SSL channel
- Password is encrypted
- Transfer is encrypted
- Typically runs over TCP port 21 or 990
- Defined by RFC 959, 1123, 4217 and 2228
- Implemented in FTP/SSL component

SFTP

SSH File Transfer Protocol

- SSH File Transfer Protocol
- Has nothing common with original FTP
- Often called 'Secure FTP'
- Password is encrypted
- Transfer is encrypted
- Typically runs over TCP port 22
- RFC not yet finished
- Implemented in SFTP client component
- Implemented in SFTP server component
- Implemented in Buru SFTP Server

② Wireshark

Diagrama Temporal : file1 por FTP

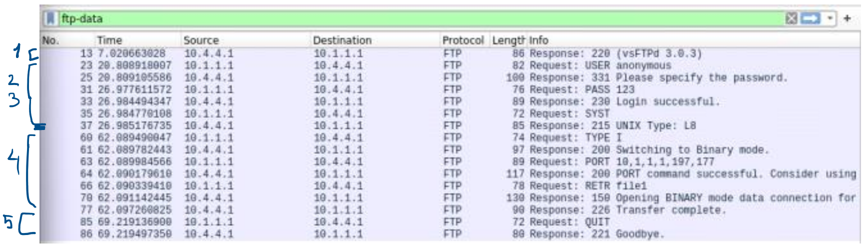
[ftp-data]

FTP usa mais do que uma conexão em simultâneo

- Fases de início de conexão
- Transferência de dados
- Fim de conexão

Tipos de segmentos trocados e os nºs de sequência usados

↳ Nos dados e nas confirmações



```
No.  Time          Source     Destination  Protocol Length Info
  13 7.020663028   10.4.4.1   10.1.1.1     FTP       86 Response: 220 (vsFTPd 3.0.3)
  23 20.808918007  10.1.1.1   10.4.4.1     FTP       82 Request: USER anonymous
  25 20.809105586  10.4.4.1   10.1.1.1     FTP      100 Response: 331 Please specify the password.
  31 26.977611572  10.1.1.1   10.4.4.1     FTP       76 Request: PASS 123
  33 26.984494347  10.4.4.1   10.1.1.1     FTP       89 Response: 230 Login successful.
  35 26.984770108  10.1.1.1   10.4.4.1     FTP       72 Request: SYST
  37 26.985176735  10.4.4.1   10.1.1.1     FTP       85 Response: 215 UNIX Type: L8
  60 62.089490047  10.1.1.1   10.4.4.1     FTP       74 Request: TYPE I
  61 62.089782443  10.4.4.1   10.1.1.1     FTP       97 Response: 200 Switching to Binary mode.
  63 62.089984566  10.1.1.1   10.4.4.1     FTP       89 Request: PORT 10,1,1,1,197,177
  64 62.090179610  10.4.4.1   10.1.1.1     FTP      117 Response: 200 PORT command successful. Consider using
  66 62.090339410  10.1.1.1   10.4.4.1     FTP       78 Request: RETR file1
  70 62.091142445  10.4.4.1   10.1.1.1     FTP      130 Response: 150 Opening BINARY mode data connection for
  77 62.097260825  10.4.4.1   10.1.1.1     FTP       90 Response: 226 Transfer complete.
  85 69.219136900  10.1.1.1   10.4.4.1     FTP       72 Request: QUIT
  86 69.219497350  10.4.4.1   10.1.1.1     FTP       80 Response: 221 Goodbye.
```

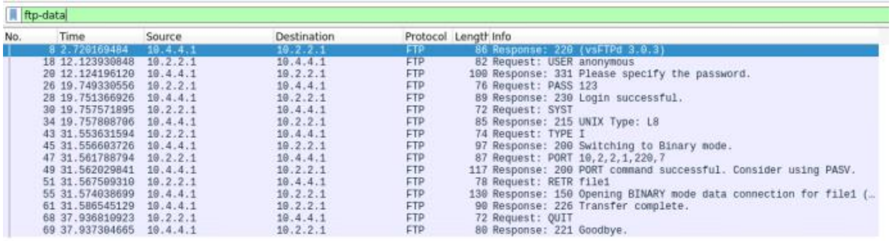
1 • Conexão FTP

2 • Login : Username

3 • Login : PassWord

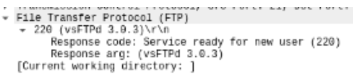
4 • Get file1

5 • Quit

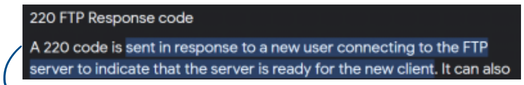


```
ftp-data

No.  Time          Source      Destination  Protocol  Length Info
   8 2.720169484   10.4.4.1    10.2.2.1     FTP        86 Response: 220 (vsFTPd 3.0.3)
  18 12.123930848  10.2.2.1    10.4.4.1     FTP        82 Request: USER anonymous
  20 12.124196120  10.4.4.1    10.2.2.1     FTP       100 Response: 331 Please specify the password.
  26 19.749330556  10.2.2.1    10.4.4.1     FTP        76 Request: PASS 123
  28 19.751366926  10.4.4.1    10.2.2.1     FTP        89 Response: 230 Login successful.
  30 19.757571895  10.2.2.1    10.4.4.1     FTP        72 Request: SYST
  34 19.757808706  10.4.4.1    10.2.2.1     FTP        85 Response: 215 UNIX Type: L8
  43 31.553631594  10.2.2.1    10.4.4.1     FTP        74 Request: TYPE I
  45 31.556603726  10.4.4.1    10.2.2.1     FTP        97 Response: 200 Switching to Binary mode.
  47 31.561788794  10.2.2.1    10.4.4.1     FTP        87 Request: PORT 10,2,2,1,220,7
  49 31.562029841  10.4.4.1    10.2.2.1     FTP       117 Response: 200 PORT command successful. Consider using PASV.
  51 31.567509310  10.2.2.1    10.4.4.1     FTP        78 Request: RETR file1
  55 31.574038699  10.4.4.1    10.2.2.1     FTP       130 Response: 150 Opening BINARY mode data connection for file1 (…
  61 31.586545129  10.4.4.1    10.2.2.1     FTP        90 Response: 226 Transfer complete.
  68 37.936810923  10.2.2.1    10.4.4.1     FTP        72 Request: QUIT
  69 37.937304665  10.4.4.1    10.2.2.1     FTP        80 Response: 221 Goodbye.
```

8: Conexão:

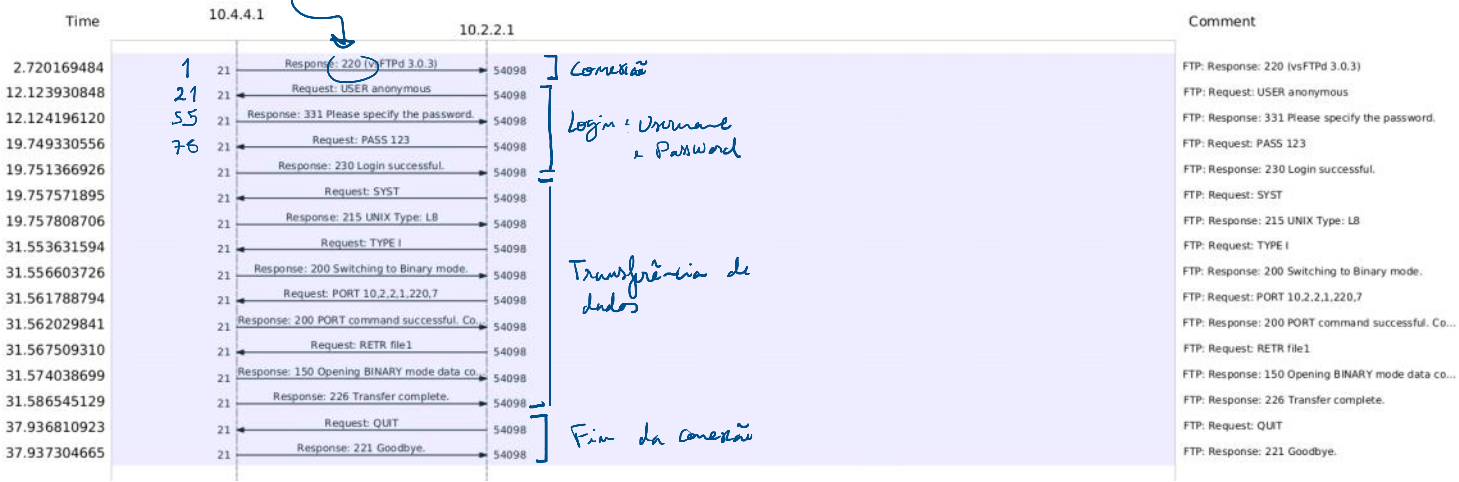


```
▾ File Transfer Protocol (FTP)
  ▾ 220 (vsFTPd 3.0.3)\r\n
        Response code: Service ready for new user (220)
        Response arg: (vsFTPd 3.0.3)
    [Current working directory: ]
```

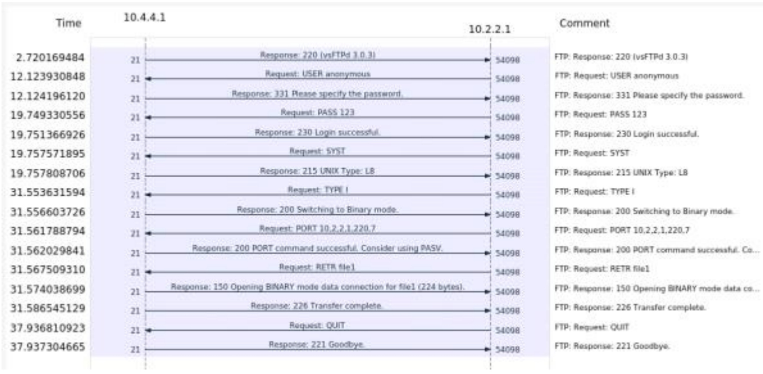


220 FTP Response code

A 220 code is sent in response to a new user connecting to the FTP server to indicate that the server is ready for the new client. It can also



```
Time          10.4.4.1                                               10.2.2.1   Comment

2.720169484   1   21  Response: 220 (vsFTPd 3.0.3)  ------------------>  54098  FTP: Response: 220 (vsFTPd 3.0.3)
12.123930848  21  21  <------------------  Request: USER anonymous        54098  FTP: Request: USER anonymous
12.124196120  55  21  Response: 331 Please specify the password. ---->  54098  FTP: Response: 331 Please specify the password.
19.749330556  76  21  <------------------  Request: PASS 123              54098  FTP: Request: PASS 123
19.751366926      21  Response: 230 Login successful. --------------->  54098  FTP: Response: 230 Login successful.
19.757571895      21  <------------------  Request: SYST                  54098  FTP: Request: SYST
19.757808706      21  Response: 215 UNIX Type: L8 ------------------->  54098  FTP: Response: 215 UNIX Type: L8
31.553631594      21  <------------------  Request: TYPE I                54098  FTP: Request: TYPE I
31.556603726      21  Response: 200 Switching to Binary mode. ------->  54098  FTP: Response: 200 Switching to Binary mode.
31.561788794      21  <------------------  Request: PORT 10,2,2,1,220,7   54098  FTP: Request: PORT 10,2,2,1,220,7
31.562029841      21  Response: 200 PORT command successful. Co… ---->  54098  FTP: Response: 200 PORT command successful. Co…
31.567509310      21  <------------------  Request: RETR file1           54098  FTP: Request: RETR file1
31.574038699      21  Response: 150 Opening BINARY mode data co… ---->  54098  FTP: Response: 150 Opening BINARY mode data co…
31.586545129      21  Response: 226 Transfer complete. -------------->  54098  FTP: Response: 226 Transfer complete.
37.936810923      21  <------------------  Request: QUIT                  54098  FTP: Request: QUIT
37.937304665      21  Response: 221 Goodbye. ------------------------>  54098  FTP: Response: 221 Goodbye.
```



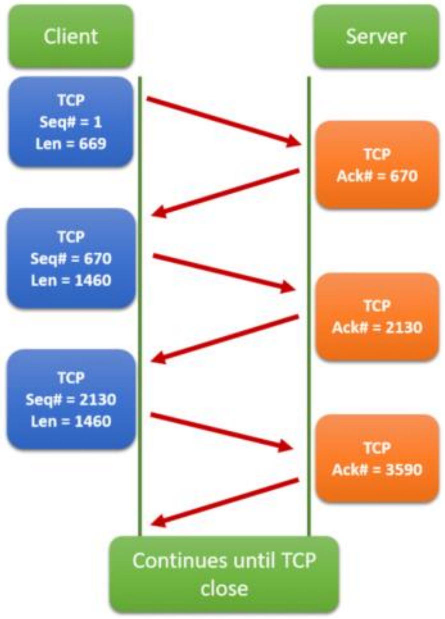
```
Time          10.4.4.1                                                                          10.2.2.1   Comment

2.720169484   21  Response: 220 (vsFTPd 3.0.3) -------------------------------------------->  54098  FTP: Response: 220 (vsFTPd 3.0.3)
12.123930848  21  <--------------------------------------------  Request: USER anonymous       54098  FTP: Request: USER anonymous
12.124196120  21  Response: 331 Please specify the password. ------------------------------->  54098  FTP: Response: 331 Please specify the password.
19.749330556  21  <--------------------------------------------  Request: PASS 123             54098  FTP: Request: PASS 123
19.751366926  21  Response: 230 Login successful. ------------------------------------------>  54098  FTP: Response: 230 Login successful.
19.757571895  21  <--------------------------------------------  Request: SYST                 54098  FTP: Request: SYST
19.757808706  21  Response: 215 UNIX Type: L8 ---------------------------------------------->  54098  FTP: Response: 215 UNIX Type: L8
31.553631594  21  <--------------------------------------------  Request: TYPE I               54098  FTP: Request: TYPE I
31.556603726  21  Response: 200 Switching to Binary mode. ---------------------------------->  54098  FTP: Response: 200 Switching to Binary mode.
31.561788794  21  <--------------------------------------------  Request: PORT 10,2,2,1,220,7  54098  FTP: Request: PORT 10,2,2,1,220,7
31.562029841  21  Response: 200 PORT command successful. Consider using PASV. -------------->  54098  FTP: Response: 200 PORT command successful. Co…
31.567509310  21  <--------------------------------------------  Request: RETR file1          54098  FTP: Request: RETR file1
31.574038699  21  Response: 150 Opening BINARY mode data connection for file1 (224 bytes). ->  54098  FTP: Response: 150 Opening BINARY mode data co…
31.586545129  21  Response: 226 Transfer complete. ----------------------------------------->  54098  FTP: Response: 226 Transfer complete.
37.936810923  21  <--------------------------------------------  Request: QUIT                 54098  FTP: Request: QUIT
37.937304665  21  Response: 221 Goodbye. --------------------------------------------------->  54098  FTP: Response: 221 Goodbye.
```

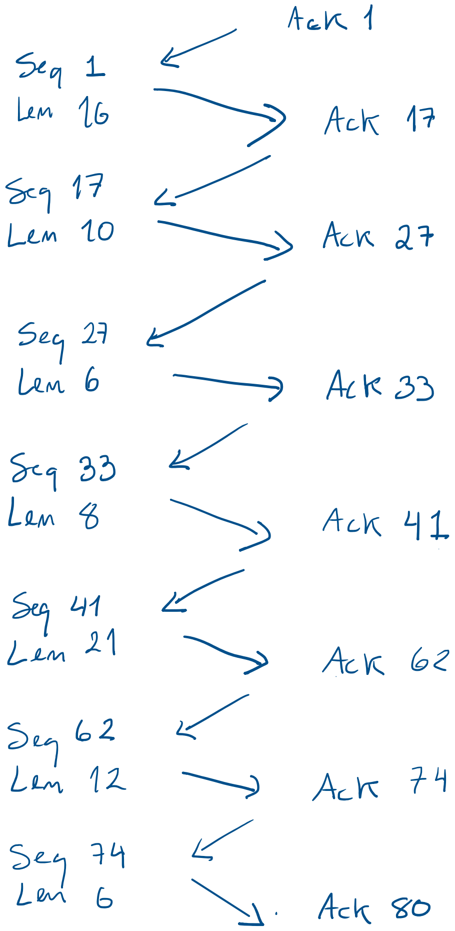
TCP Sequence (seq) and Acknowledgement (ack) numbers help enable ordered reliable data transfer for TCP streams.

The seq number is sent by the TCP client, indicating how much data has been sent for the session (also known as the byte-order number).
The ack number is sent by the TCP server, indicating that it has received cumulated data and is ready for the next segmen.

The TCP seq and ack numbers are coordinated with one another

[By default, Wireshark converts all sequence and acknowledgement numbers into relative numbers. This means that all SEQ and ACK numbers always start at 0 for the first packet seen in each conversation.]



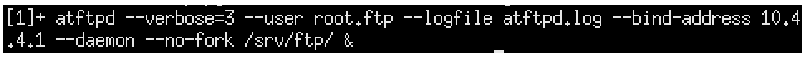


3. Obtenha a partir do *wireshark*, ou desenhe manualmente, um diagrama temporal para a transferência de *file1* por TFTP. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

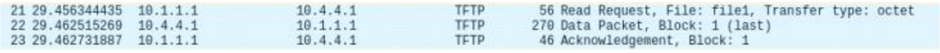
- Transferência do file1 por TFTP
- Indicar fases : início de conexão
  transferência de dados
  fim de conexão
- Segmentos trocados

- Segmentos trocados
- Nº seq usados

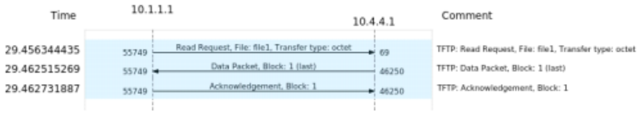
Servidor1



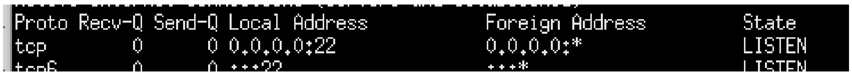
```
[1]+ atftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.4
.4.1 --daemon --no-fork /srv/ftp/ &
```



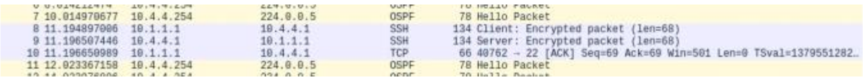
```
21 29.456344435  10.1.1.1             10.4.4.1            TFTP        56 Read Request, File: file1, Transfer type: octet
22 29.462515269  10.4.4.1             10.1.1.1            TFTP       270 Data Packet, Block: 1 (last)
23 29.462731887  10.1.1.1             10.4.4.1            TFTP        46 Acknowledgement, Block: 1
```


Time
10.1.1.1
10.4.4.1
Comment
29.456344435
29.462515269
29.462731887
55749
55749
55749
Read Request, File: file1, Transfer type: octet
Data Packet, Block: 1 (last)
Acknowledgement, Block: 1
69
46250
46250
TFTP: Read Request, File: file1, Transfer type: octet
TFTP: Data Packet, Block: 1 (last)
TFTP: Acknowledgement, Block: 1

SSH ativo na porta 22:

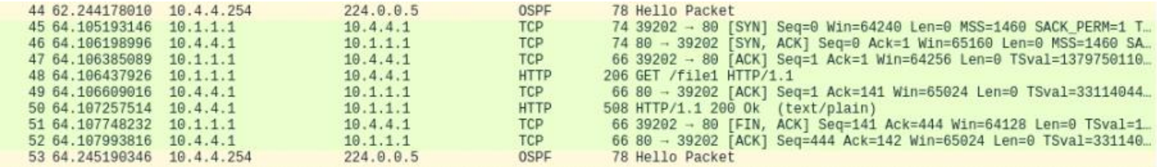


```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
```



```
 7 10.014970677  10.4.4.254           224.0.0.5           OSPF        78 Hello Packet
 8 11.194897006  10.1.1.1             10.4.4.1            SSH        134 Client: Encrypted packet (len=68)
 9 11.196507446  10.4.4.1             10.1.1.1            SSH        134 Server: Encrypted packet (len=68)
10 11.196650989  10.1.1.1             10.4.4.1            TCP         66 40762 → 22 [ACK] Seq=69 Ack=69 Win=501 Len=0 TSval=1379551282…
11 12.023367158  10.4.4.254           224.0.0.5           OSPF        78 Hello Packet
```

HTTP transport layer



```
44 62.244178010  10.4.4.254           224.0.0.5           OSPF        78 Hello Packet
45 64.105193146  10.1.1.1             10.4.4.1            TCP         74 39202 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
46 64.106198996  10.4.4.1             10.1.1.1            TCP         74 80 → 39202 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA…
47 64.106385089  10.1.1.1             10.4.4.1            TCP         66 39202 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1379750110…
48 64.106437926  10.1.1.1             10.4.4.1            HTTP       206 GET /file1 HTTP/1.1
49 64.106609016  10.4.4.1             10.1.1.1            TCP         66 80 → 39202 [ACK] Seq=1 Ack=141 Win=65024 Len=0 TSval=33114044…
50 64.107257514  10.4.4.1             10.1.1.1            HTTP       508 HTTP/1.1 200 Ok  (text/plain)
51 64.107748232  10.1.1.1             10.4.4.1            TCP         66 39202 → 80 [FIN, ACK] Seq=141 Ack=444 Win=64128 Len=0 TSval=1…
52 64.107993816  10.4.4.1             10.1.1.1            TCP         66 80 → 39202 [ACK] Seq=444 Ack=142 Win=65024 Len=0 TSval=331140…
53 64.245190346  10.4.4.254           224.0.0.5           OSPF        78 Hello Packet
```

https://stackoverflow.com/questions/8849240/why-when-i-transfer-a-file-through-sftp-it-takes-longer-than-ftp

https://madpackets.com/2018/04/25/tcp-sequence-and-acknowledgement-numbers-explained/

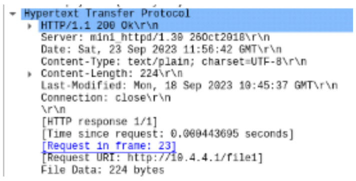
https://packetlife.net/blog/2010/jun/7/understanding-tcp-sequence-acknowledgment-numbers/

https://www.ibm.com/support/pages/ftp-performance-considerations

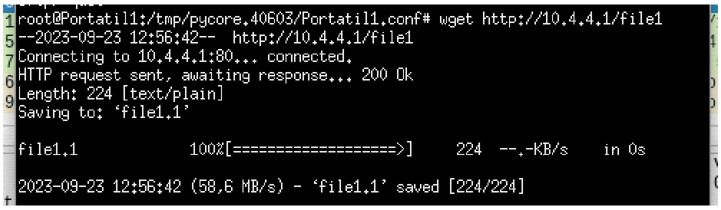
https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol

https://www.w3.org/Protocols/HTTP-NG/http-prob.html

https://www.cloudflare.com/learning/performance/http2-vs-http1.1/

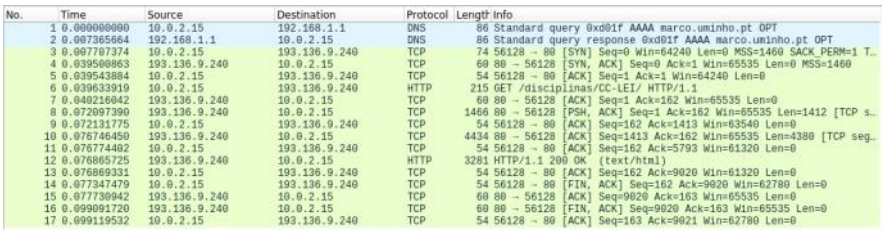


```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 Ok\r\n
    Server: mini_httpd/1.30 26Oct2018\r\n
    Date: Sat, 23 Sep 2023 11:56:42 GMT\r\n
    Content-Type: text/plain; charset=UTF-8\r\n
  ▸ Content-Length: 224\r\n
    Last-Modified: Mon, 18 Sep 2023 10:45:37 GMT\r\n
    Connection: close\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.000443695 seconds]
    [Request in frame: 23]
    [Request URI: http://10.4.4.1/file1]
    File Data: 224 bytes
```

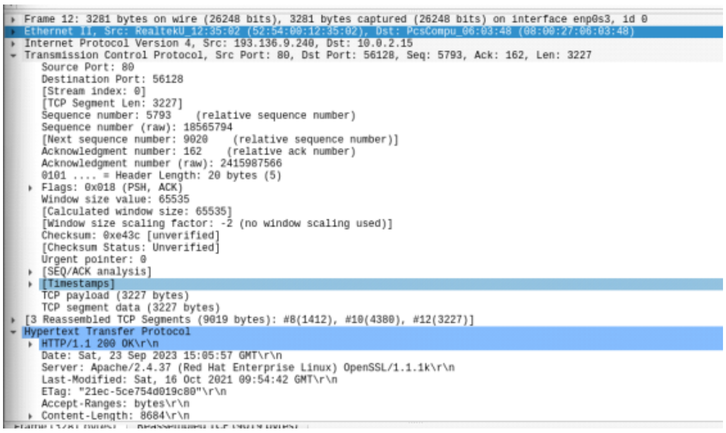


```
root@Portatil1:/tmp/pycore.40603/Portatil1.conf# wget http://10.4.4.1/file1
--2023-09-23 12:56:42--  http://10.4.4.1/file1
Connecting to 10.4.4.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 224 [text/plain]
Saving to: ‘file1.1’

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

2023-09-23 12:56:42 (58.6 MB/s) - ‘file1.1’ saved [224/224]
```

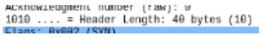
HTTP



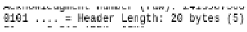
```
No.    Time          Source             Destination        Protocol Length Info
     1 0.000000000   10.0.2.15          192.168.1.1        DNS         86 Standard query 0xd01f AAAA marco.uminho.pt OPT
     2 0.007365664   192.168.1.1        10.0.2.15          DNS         86 Standard query response 0xd01f AAAA marco.uminho.pt OPT
     3 0.007707374   10.0.2.15          193.136.9.240      TCP         74 56128 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
     4 0.039500863   193.136.9.240      10.0.2.15          TCP         60 80 → 56128 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
     5 0.039543884   10.0.2.15          193.136.9.240      TCP         54 56128 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
     6 0.039633919   10.0.2.15          193.136.9.240      HTTP       215 GET /disciplinas/CC-LEI/ HTTP/1.1
     7 0.040216042   193.136.9.240      10.0.2.15          TCP         60 80 → 56128 [ACK] Seq=1 Ack=162 Win=65535 Len=0
     8 0.072097390   193.136.9.240      10.0.2.15          TCP       1466 80 → 56128 [PSH, ACK] Seq=1 Ack=162 Win=65535 Len=1412 [TCP s…
     9 0.072131775   10.0.2.15          193.136.9.240      TCP         54 56128 → 80 [ACK] Seq=162 Ack=1413 Win=63540 Len=0
    10 0.076746450   193.136.9.240      10.0.2.15          TCP       4434 80 → 56128 [ACK] Seq=1413 Ack=162 Win=65535 Len=4380 [TCP seg…
    11 0.076774402   10.0.2.15          193.136.9.240      TCP         54 56128 → 80 [ACK] Seq=162 Ack=5793 Win=61320 Len=0
    12 0.076865725   193.136.9.240      10.0.2.15          HTTP      3281 HTTP/1.1 200 OK  (text/html)
    13 0.076869331   10.0.2.15          193.136.9.240      TCP         54 56128 → 80 [ACK] Seq=162 Ack=9020 Win=61320 Len=0
    14 0.077347479   10.0.2.15          193.136.9.240      TCP         54 56128 → 80 [FIN, ACK] Seq=162 Ack=9020 Win=62780 Len=0
    15 0.077730942   193.136.9.240      10.0.2.15          TCP         60 80 → 56128 [ACK] Seq=9020 Ack=163 Win=65535 Len=0
    16 0.099091720   193.136.9.240      10.0.2.15          TCP         60 80 → 56128 [FIN, ACK] Seq=9020 Ack=163 Win=65535 Len=0
    17 0.099119532   10.0.2.15          193.136.9.240      TCP         54 56128 → 80 [ACK] Seq=163 Ack=9021 Win=62780 Len=0
```



```
▸ Frame 12: 3281 bytes on wire (26248 bits), 3281 bytes captured (26248 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_06:03:48 (08:00:27:06:03:48)
▸ Internet Protocol Version 4, Src: 193.136.9.240, Dst: 10.0.2.15
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 56128, Seq: 5793, Ack: 162, Len: 3227
     Source Port: 80
     Destination Port: 56128
     [Stream index: 0]
     [TCP Segment Len: 3227]
     Sequence number: 5793    (relative sequence number)
     Sequence number (raw): 18565794
     [Next sequence number: 9020    (relative sequence number)]
     Acknowledgment number: 162    (relative ack number)
     Acknowledgment number (raw): 2415987566
     0101 .... = Header Length: 20 bytes (5)
   ▸ Flags: 0x018 (PSH, ACK)
     Window size value: 65535
     [Calculated window size: 65535]
     [Window size scaling factor: -2 (no window scaling used)]
     Checksum: 0xe43c [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   ▸ [SEQ/ACK analysis]
   ▸ [Timestamps]
     TCP payload (3227 bytes)
     TCP segment data (3227 bytes)
▸ [3 Reassembled TCP Segments (9019 bytes): #8(1412), #10(4380), #12(3227)]
▾ Hypertext Transfer Protocol
   ▸ HTTP/1.1 200 OK\r\n
     Date: Sat, 23 Sep 2023 15:05:57 GMT\r\n
     Server: Apache/2.4.37 (Red Hat Enterprise Linux) OpenSSL/1.1.1k\r\n
     Last-Modified: Sat, 16 Oct 2021 09:54:42 GMT\r\n
     ETag: "21ec-5ce754d019c80"\r\n
     Accept-Ranges: bytes\r\n
   ▸ Content-Length: 8684\r\n
```

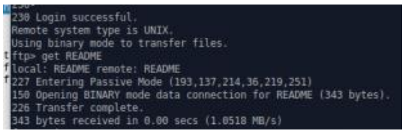


```
Acknowledgment number (raw): 0
1010 .... = Header Length: 40 bytes (10)
Flags: 0x002 (SYN)
```

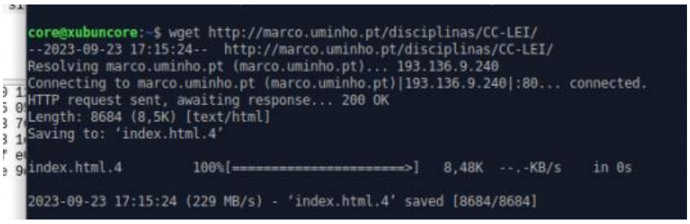


```
0101 .... = Header Length: 20 bytes (5)
```

FTP



```
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get README
local: README remote: README
227 Entering Passive Mode (193,137,214,36,219,251)
150 Opening BINARY mode data connection for README (343 bytes).
226 Transfer complete.
343 bytes received in 0.00 secs (1.0518 MB/s)
```



```
core@xubuncore:~$ wget http://marco.uminho.pt/disciplinas/CC-LEI/
--2023-09-23 17:15:24--  http://marco.uminho.pt/disciplinas/CC-LEI/
Resolving marco.uminho.pt (marco.uminho.pt)... 193.136.9.240
Connecting to marco.uminho.pt (marco.uminho.pt)|193.136.9.240|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8684 (8,5K) [text/html]
Saving to: 'index.html.4'

index.html.4        100%[=======================>]   8,48K  --.-KB/s    in 0s

2023-09-23 17:15:24 (229 MB/s) - 'index.html.4' saved [8684/8684]
```