

UNIVERSIDADE DO MINHO  
DEPARTAMENTO DE INFORMÁTICA  
COMUNICAÇÕES POR COMPUTADOR

Trabalho Prático Nº 1

Diogo Azevedo (A89565)  
António Silva (A89558)  
José Martins (A90122)

16 de março de 2021



Diogo Azevedo A89565



António Silva A89558



José Martins A90122

## Conteúdo

<b>1</b>	<b>Questões e Respostas</b>	<b>3</b>
1.1	Inclua no relatório uma tabela em que identifique, para cada comando executado, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte, como ilustrado no exemplo seguinte: . . . . .	3
1.2	Uma representação num diagrama temporal das transferências da file1 por FTP e TFTP respetivamente. Se for caso disso, identifique as fases de estabelecimento de conexão, transferência de dados e fim de conexão. Identifica também claramente os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações. (Nota: a transferência por FTP envolve mais que uma conexão FTP, nomeadamente uma de controlo [ftp] e outra de dados [ftp-data]. Faça o diagrama apenas para a conexão de transferência de dados do ficheiro mais pequeno)	8
1.3	Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança; . . . . .	10
1.4	As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. Discuta, relacionando a resposta com as experiências realizadas, as influências das situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis (se possível, relacionando com alguns dos mecanismos de transporte envolvidos). . . . .	11
<b>2</b>	<b>Conclusão</b>	<b>12</b>

## Questões e Respostas

- 1.1 Inclua no relatório uma tabela em que identifique, para cada comando executado, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte, como ilustrado no exemplo seguinte:

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)
Ping	não aplicável	não aplicável	não aplicável	não aplicável
tracert	não aplicável	UDP	*várias	8
telnet	TELNET	TCP	23	20
ftp	FTP	TCP	21	20
Tftp	TFTP	UDP	69	8
browser/http	HTTP	UDP	53	8
nslookup	DNS	UDP	53	8
ssh	SSHv2	TCP	22	20

Figura 1.1: Tabela de comandos

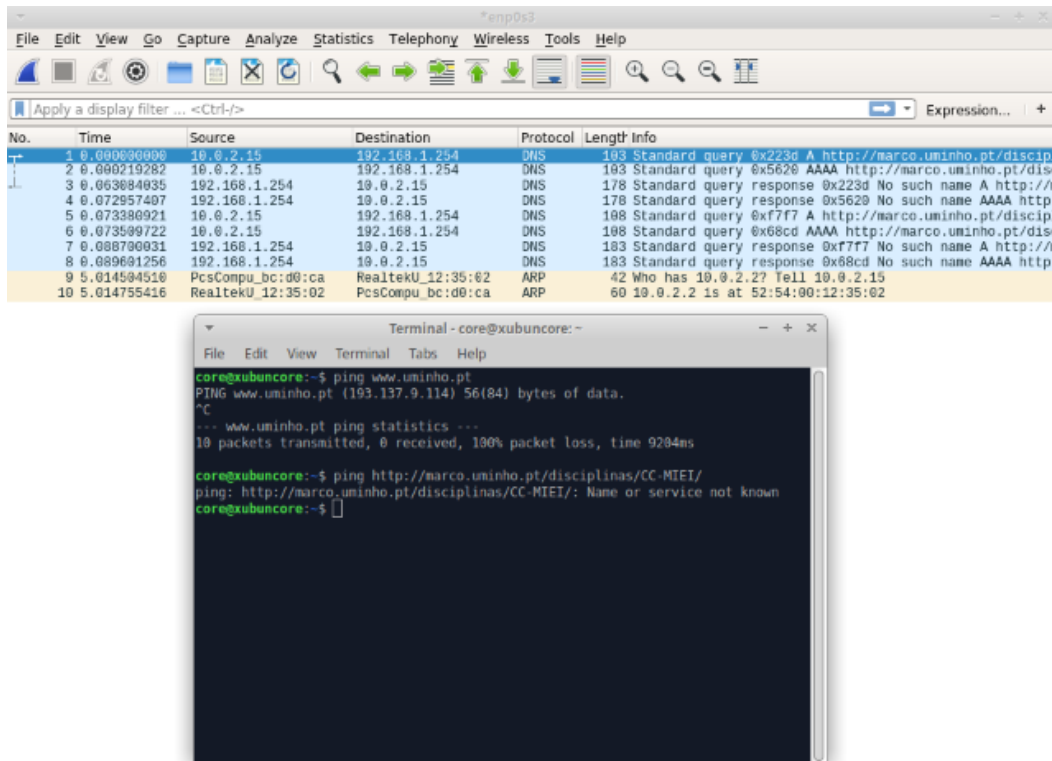


Figura 1.2: Ping

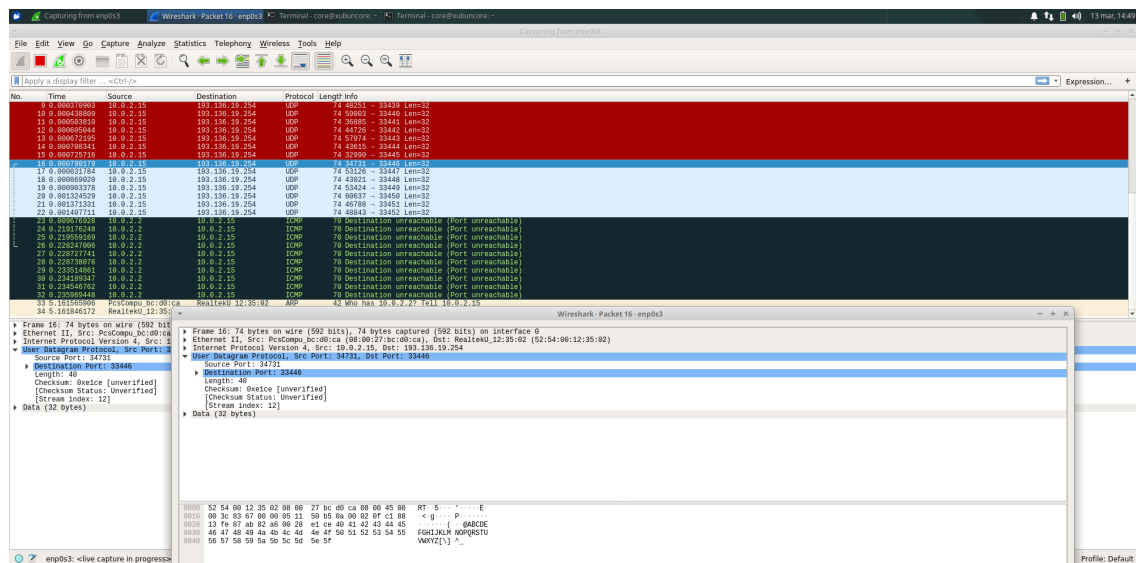


Figura 1.3: traceroute

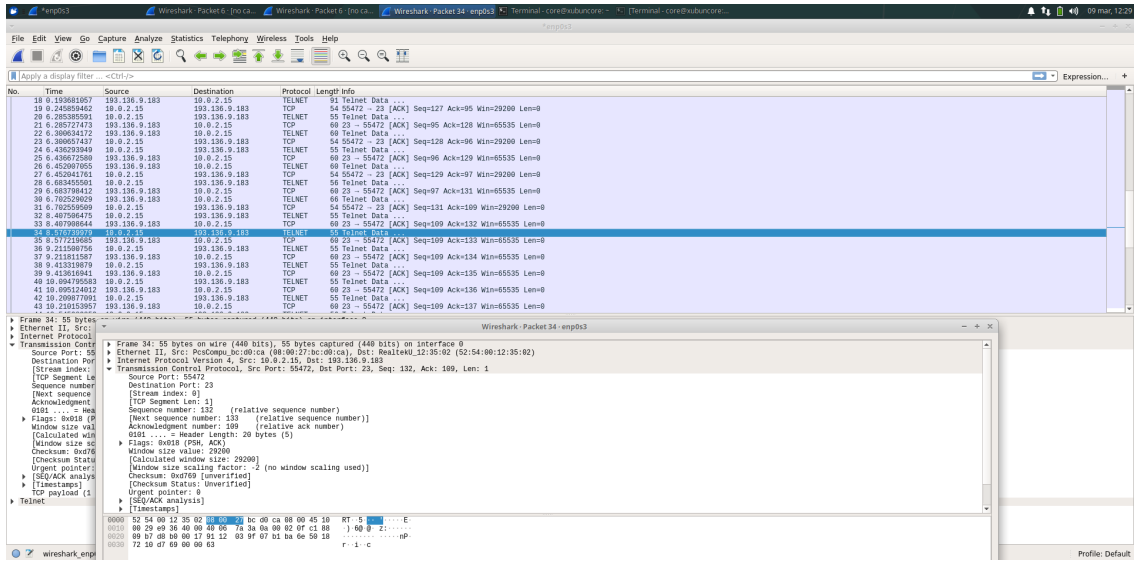


Figura 1.4: telnet

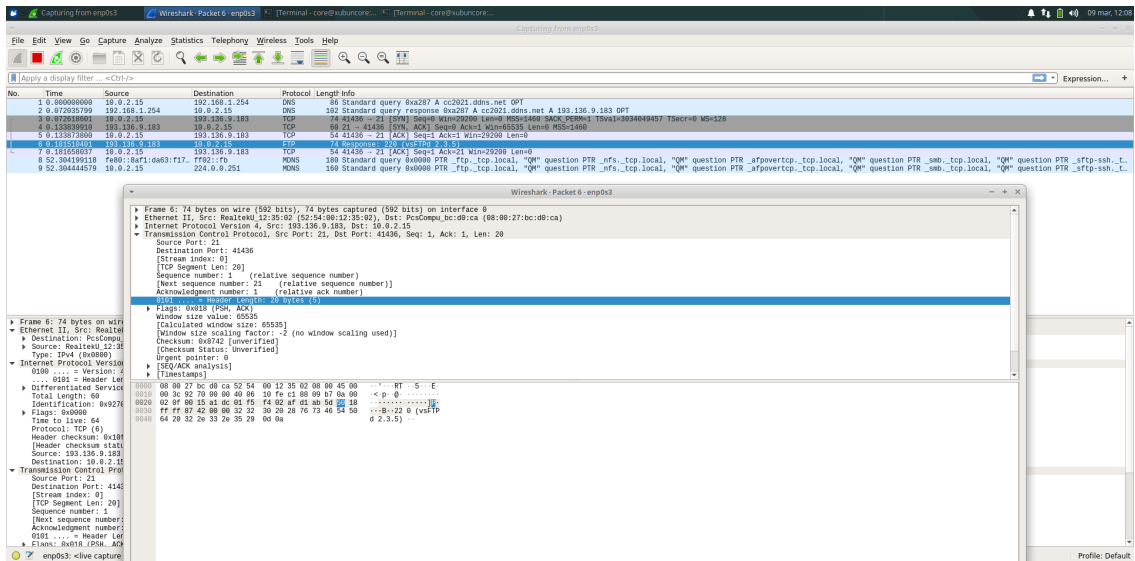


Figura 1.5: ftp

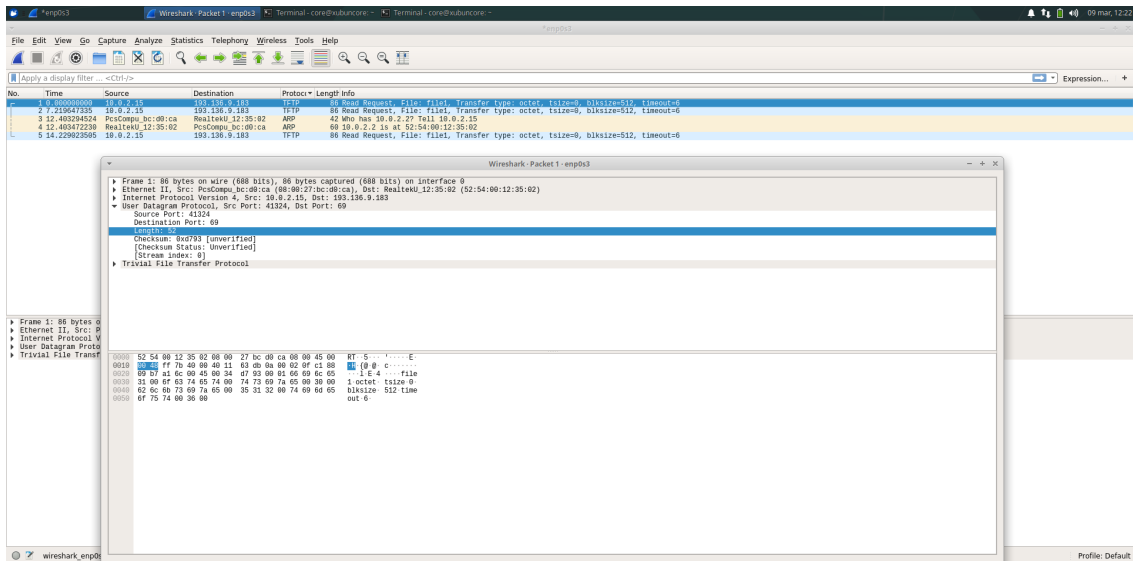


Figura 1.6: Tftp

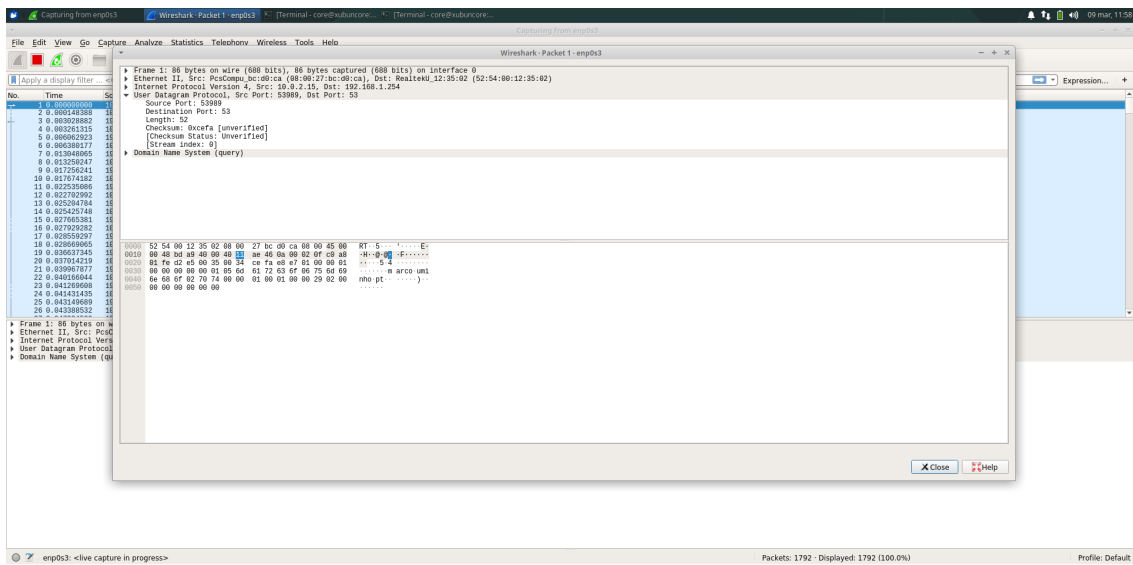


Figura 1.7: Browser/http

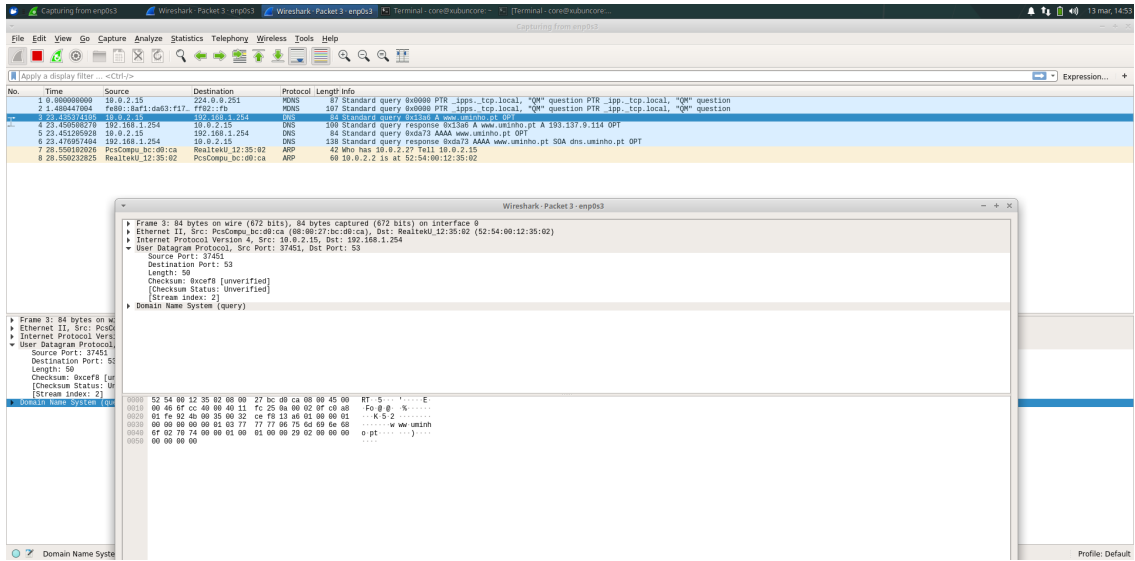


Figura 1.8: nslookup

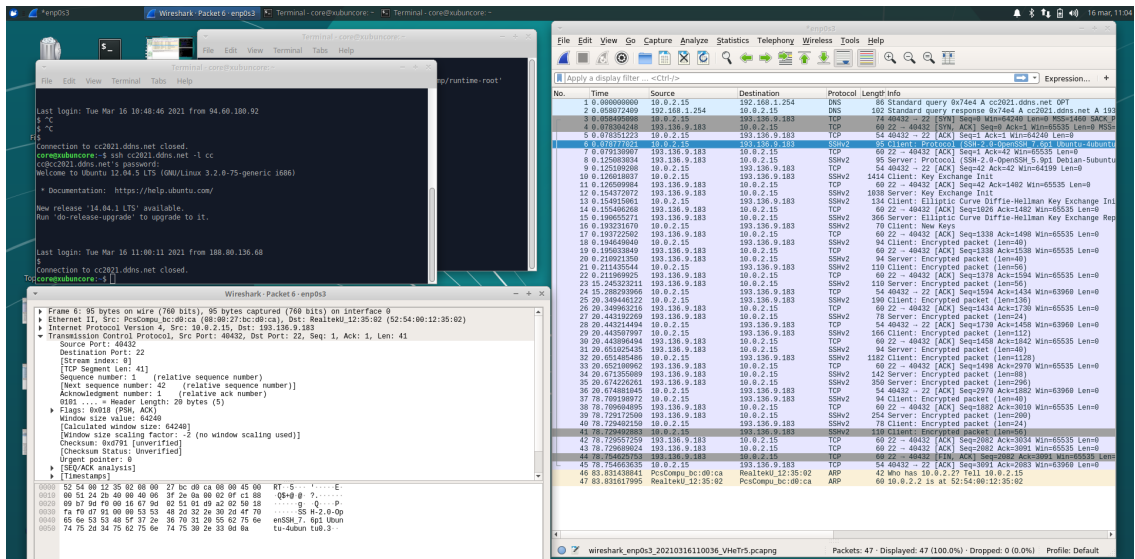


Figura 1.9: ssh

1.2 Uma representação num diagrama temporal das transferências da file1 por FTP e TFTP respetivamente. Se for caso disso, identifique as fases de estabelecimento de conexão, transferência de dados e fim de conexão. Identifica também claramente os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações. (Nota: a transferência por FTP envolve mais que uma conexão FTP, nomeadamente uma de controlo [ftp] e outra de dados [ftp-data]. Faça o diagrama apenas para a conexão de transferência de dados do ficheiro mais pequeno)

40	189,756593492	10.4.4.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
41	189,799754194	10.1.1.1	10.4.4.1	TFTP	558 Data Packet, Block: 1
42	189,800829621	10.4.4.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1

Figura 1.10: WireShark da transferência do file1 por TFTP

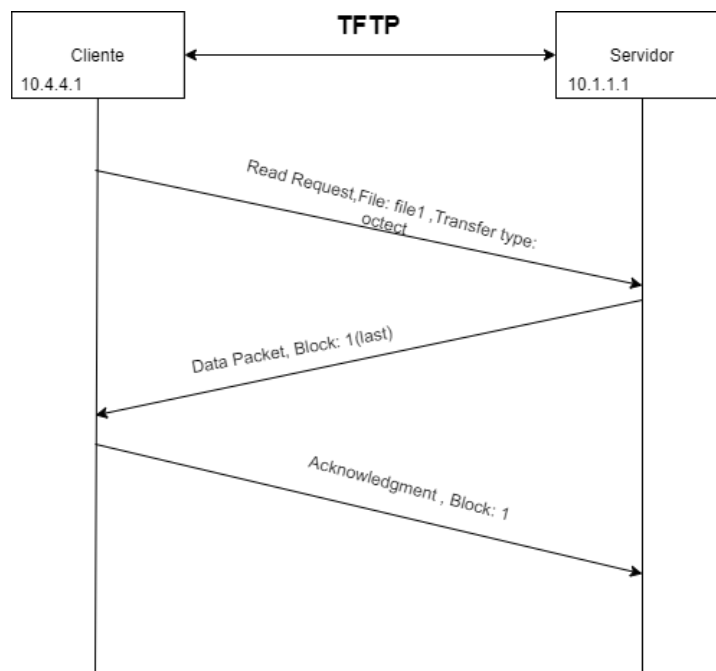


Figura 1.11: Diagrama da transferência do file1 por TFTP



51	29.158756267	10.4.4.1	10.1.1.1	FTP	78 Request: RETR file1
52	29.163990762	10.1.1.1	10.4.4.1	TCP	74 29 → 38659 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3594463826 TSecr=0 WS=128
53	29.168124941	10.4.4.1	10.1.1.1	TCP	74 38659 → 29 [SYN, ACK] Seq=9 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1866189674 TSecr=3594463826
54	29.168884130	10.1.1.1	10.4.4.1	TCP	66 29 → 38659 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3594463833 TSecr=1866189674
55	29.169249478	10.1.1.1	10.4.4.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (726 bytes).
56	29.170562785	10.1.1.1	10.4.4.1	FTP-DA	792 FTP Data: 726 bytes (PORT) (RETR file1)
57	29.179574957	10.1.1.1	10.4.4.1	TCP	66 29 → 38659 [FIN, ACK] Seq=727 Ack=1 Win=64256 Len=0 TSval=3594463835 TSecr=1866189674
58	29.171318318	10.4.4.1	10.1.1.1	TCP	66 38659 → 29 [ACK] Seq=1 Ack=727 Win=64512 Len=0 TSval=1866189689 TSecr=3594463834
59	29.171854039	10.4.4.1	10.1.1.1	TCP	66 38659 → 29 [FIN, ACK] Seq=1 Ack=728 Win=64512 Len=0 TSval=1866189689 TSecr=3594463835

Figura 1.12: WireShark da transferência do file1 por FTP

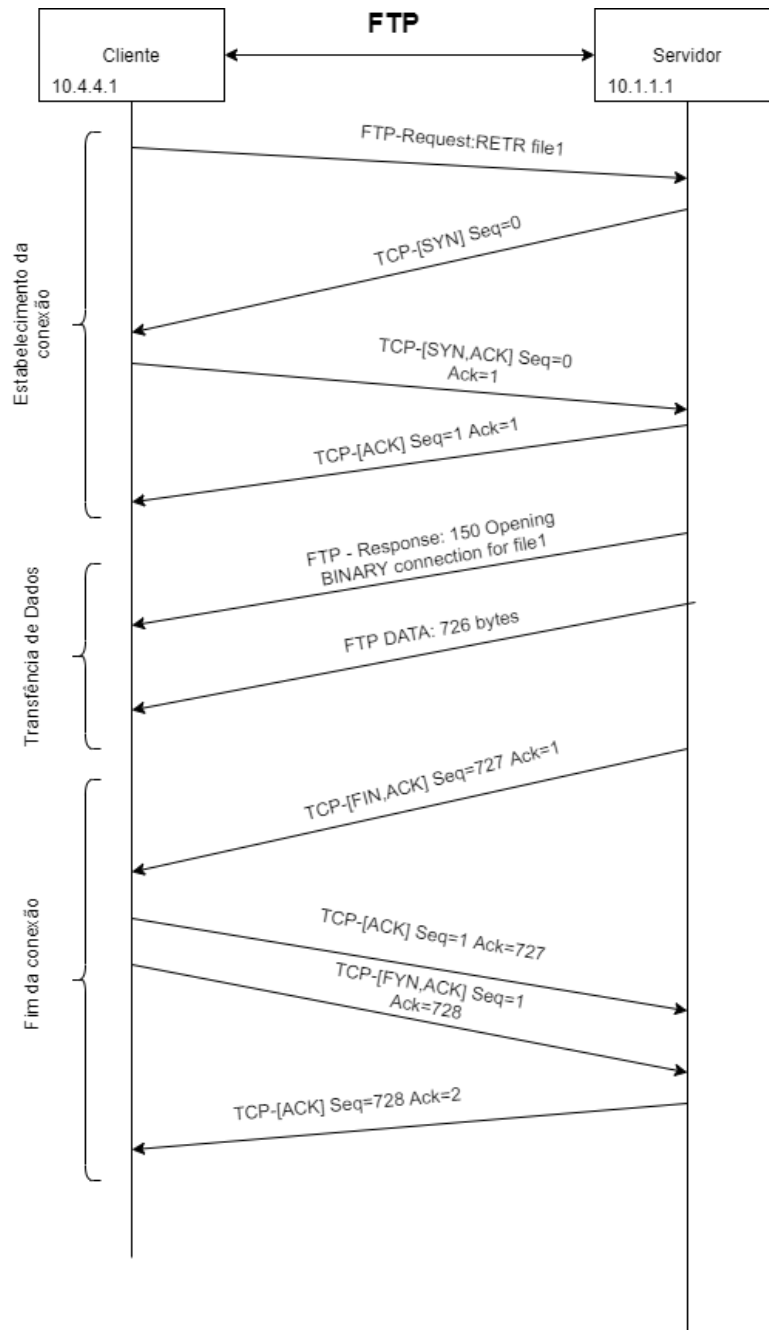


Figura 1.13: Diagrama da transferência do file1 por FTP

### 1.3 Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança;

**HTTP:** Pelo que podemos observar, através da análise dos pacotes no Wireshark, no router 1 da topologia do CORE, o protocolo **HTTP** utiliza o protocolo *TCP* como protocolo da camada de transporte e possui vários esquemas de autenticação. Este é fiável e usa pipelining o que o torna bastante rápido. Utiliza mecanismos de "handshake" e autenticação, tornando-se relativamente pouco complexo. No entanto, este é inseguro pois para qualquer pessoa na rede é possível ver o conteúdo dos ficheiros antes de chegar ao recetor, isto porque o **HTTP** não é encriptado e por isso é vulnerável a ataques.

**FTP:** O protocolo de transporte usado pelo **FTP** é o *TCP*. É um serviço básico de transferência fiável de ficheiros e não aparenta implementar nenhuma segurança adicional, porque todos os dados de conexão estão diretamente nos pacotes enviados, pelo que são fáceis de obter por parte de alguém que esteja à escuta na rede (até pudemos verificar que no processo de autenticação, as palavras-chave são transmitidas na rede em texto claro). O **FTP** é pouco eficiente, no entanto garante a transferência.

**TFTP:** O **TFTP** é um serviço básico de transferência de dados não fiável, dado que usa *UDP*. Não implementa mecanismos de autenticação nem de encriptação, pelo que é pouco complexo e seguro. Contudo é bastante rápido e eficiente, embora não assegure a receção dos dados.

**SFTP:** O **SFTP** usa o *TCP* como protocolo de transporte. Implementa encriptação (através de *SSH*) de conexão. Todas as mensagens são estabelecidas entre um cliente e um servidor, com processos de autenticação, e encriptadas para essa mesma conexão, o que assegura um elevado nível de segurança. O uso de *SSH* causa overhead que diminui a eficiência da transmissão.

#### 1.4 As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. Discuta, relacionando a resposta com as experiências realizadas, as influências das situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis (se possível, relacionando com alguns dos mecanismos de transporte envolvidos).

As características das ligações de rede têm uma enorme influência na fiabilidade e qualidade nos níveis de Transporte e de Aplicação.

Para aplicações que trabalham sobre **UDP**, não podemos ter a garantia que os pacotes enviados chegam ao destino nem que informação em falta seja reenviada. O **UDP** produz uma menor carga sobre o mecanismo de transporte e menor congestionamento da rede, sendo por isso usado quando necessitamos de uma entrega de dados rápida e simples, sem análise de estado.

Para aplicações que trabalham sobre **TCP**, temos a certeza que os pacotes chegam na ordem correta e o seu envio/recepção é confirmado por ambos os lados. No entanto, este processo gera um "peso" sobre a rede, tornando-a mais lenta e possivelmente congestionada, devido ao elevado número de transferência de pacotes que o mecanismo de troca de mensagens entre o destino e a origem forma. É, portanto, sacrificada performance em troca de fiabilidade.

Posto isto, ambas são opções válidas, apenas dependem do tipo de dados que se querem enviar e que tipo de conexão se quer estabelecer. Por exemplo, **UDP** é o mais indicado para *streaming*/jogos *online* sem atrasos e **TCP** para transferência de ficheiros sem erros.

## Conclusão

Terminado este trabalho prático, sentimos que conseguimos, num contexto geral, atingir os objetivos pretendidos e por isso o grupo sente-se satisfeito. Foi estabelecida a ponte entre a matéria lecionada nas aulas teóricas, bem como a aplicação prática dos conceitos propostos.

Pudemos observar os diferentes comportamentos dos vários protocolos, de aplicação e de transporte, de forma a compreendê-los melhor, utilizando para isso o Wireshark, no qual pudemos acompanhar todo o processo de transferência de dados.

Na realização deste TP, percebemos também as diferenças entre os protocolos *TCP* e *UDP* e as situações nas quais a sua utilização é a mais indicada.