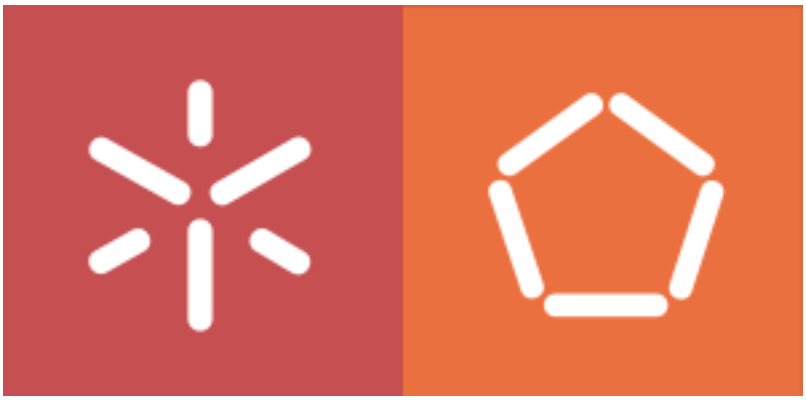

Universidade do Minho

Departamento de Informática

Comunicações por Computador

Grupo nº 8 PL1

Pedro Veloso (A89557)

Carlos Preto (A89587)

Joan Rodriguez (A89980)

Questões e Respostas

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	ICMP/ Não aplicável	Não aplicável	Não aplicável	Não aplicável
traceroute	ICMP/ Não aplicável	UDP	33434-33523	8
telnet	TELNET	TCP	23	20
ftp	FTP	TCP	21	20
Tftp	TFTP	UDP	69	8
browser/http	HTTP	TCP	80	20
nslookup	DNS	UDP	53	8
ssh	SSHv2	TCP	22	20

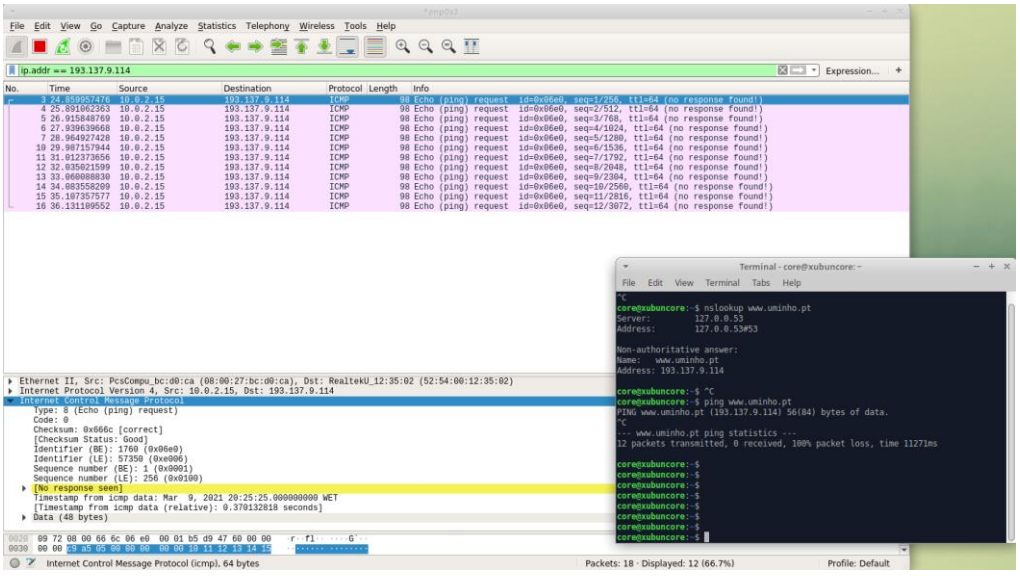

Figura 1: Ping

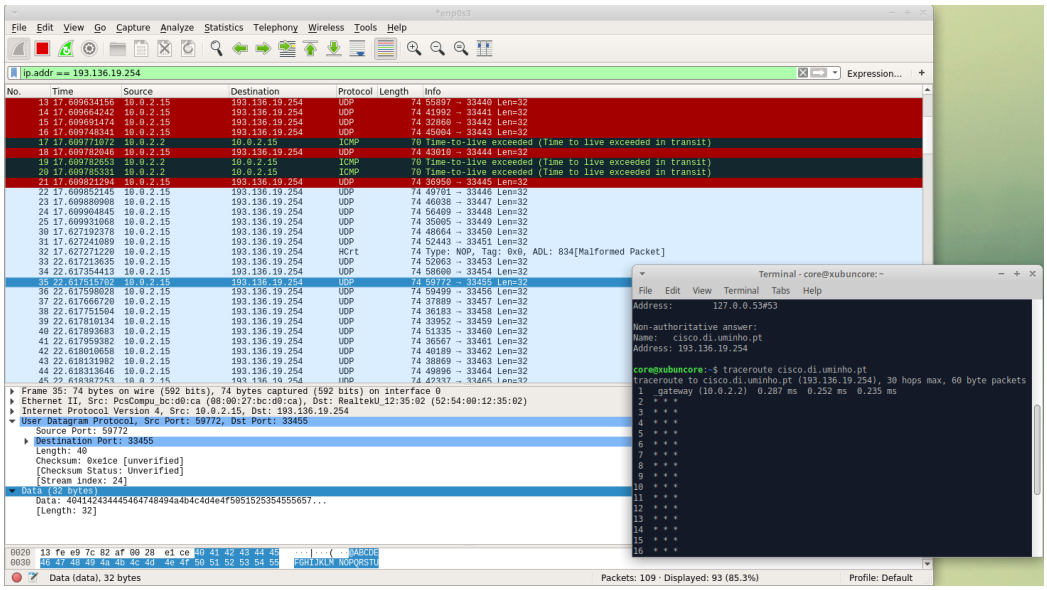

Figura 2: Traceroute

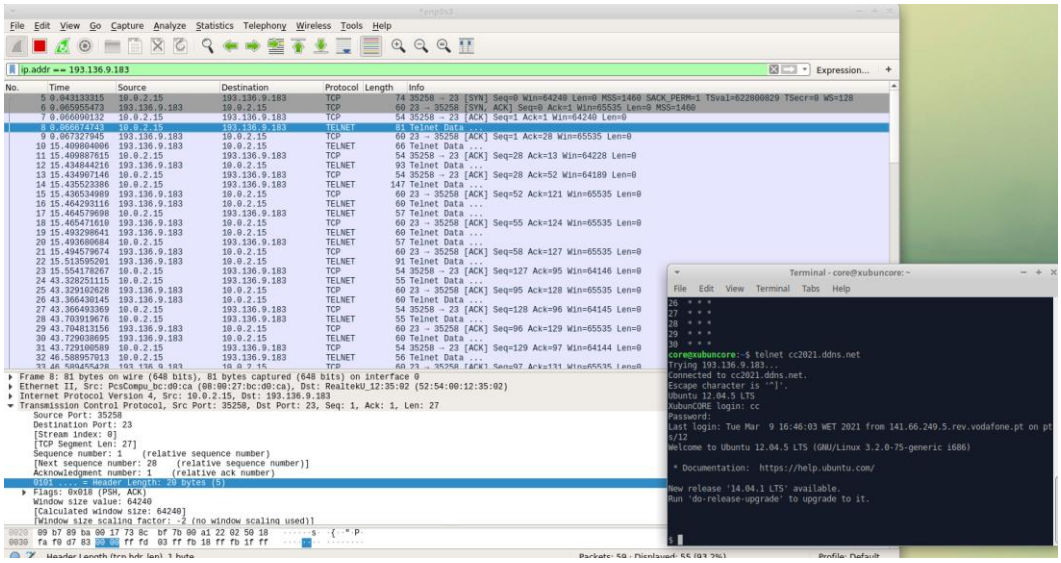

Figura 3: Telnet

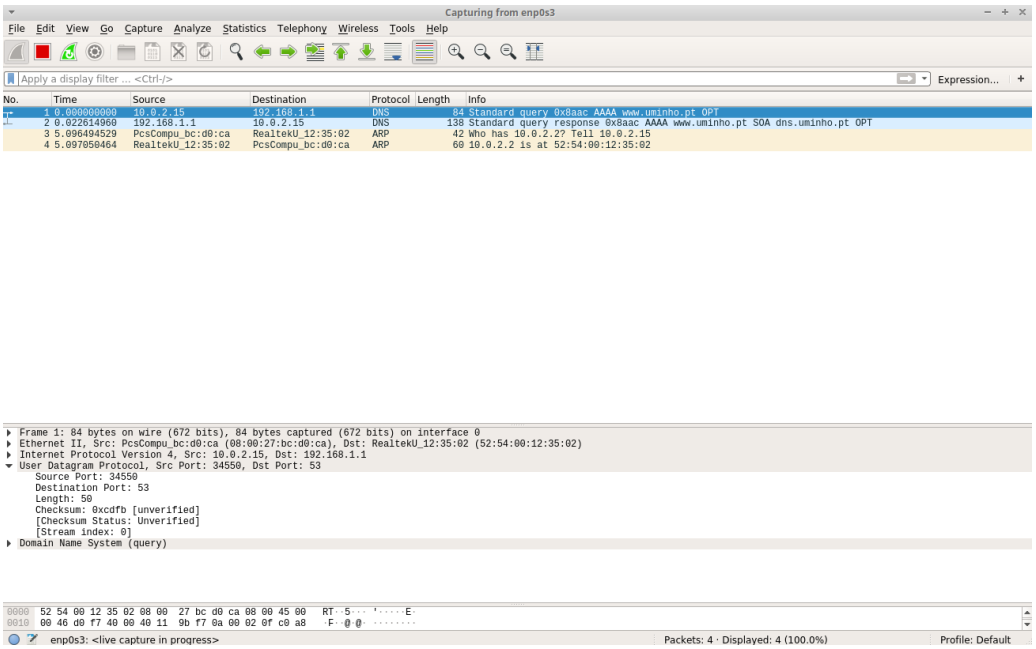

Figura 4: Nslookup

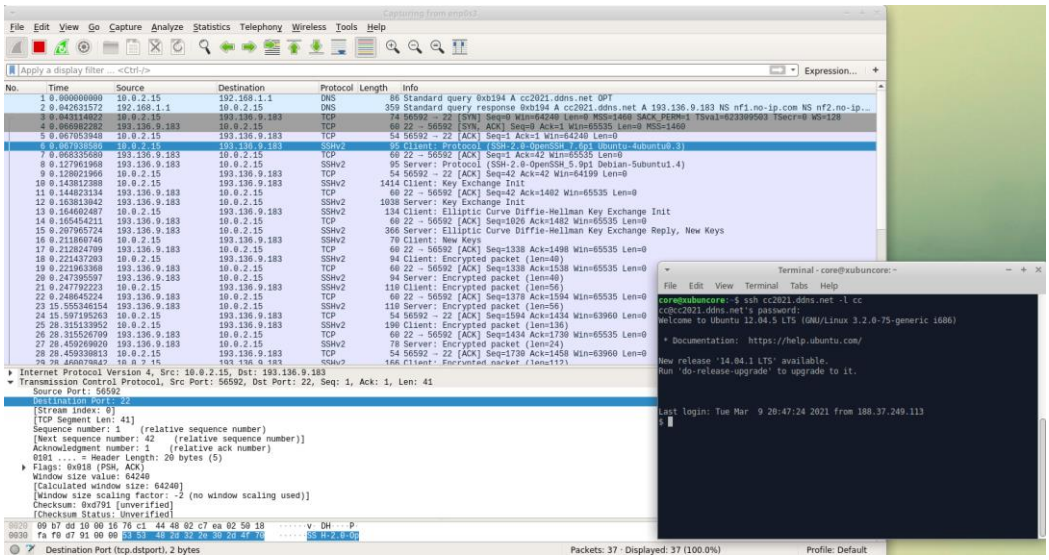

Figura 5: Ssh

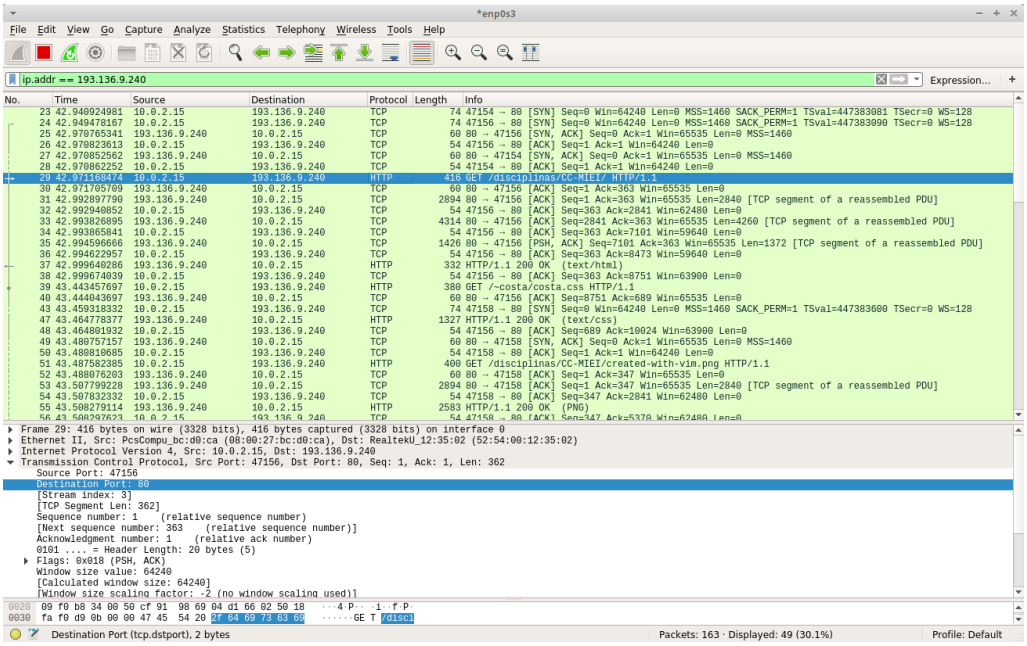

Figura 6: Http

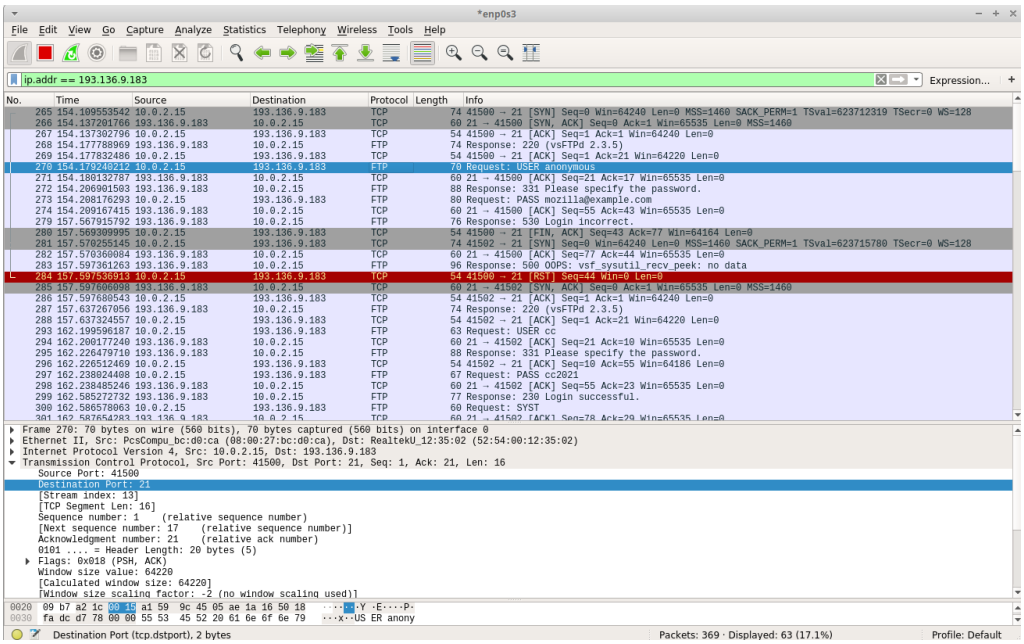

Figura 7: Ftp

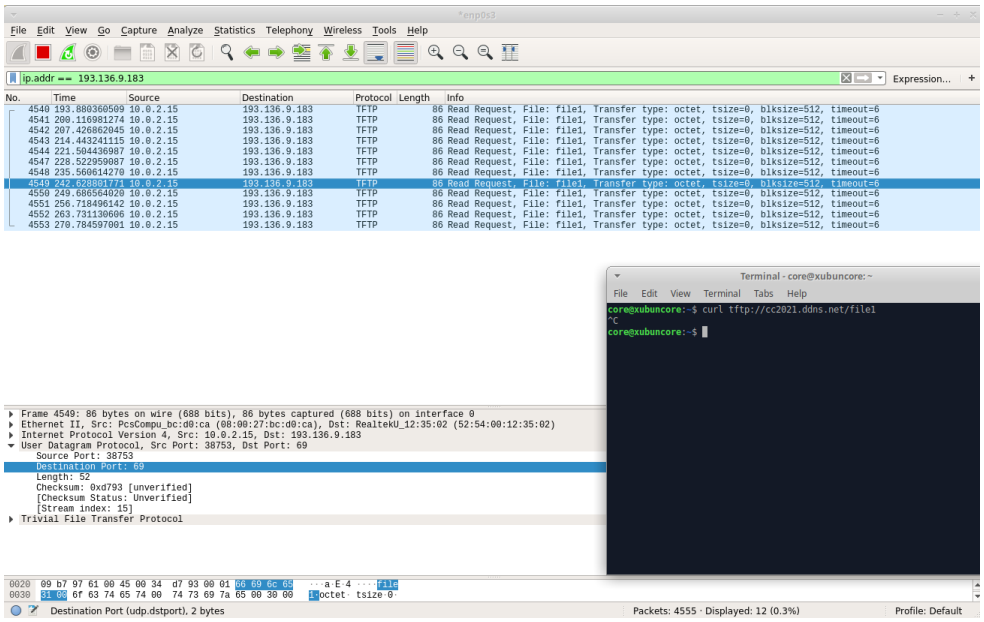

Figura 8: Tftp

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)

(i) FTP

Após se ter executado o core com a topologia virtual CC-Topo-2021.imn, criou-se uma *bash shell* no nó Server1 e no nó Laptop1. Por fim, lançou-se um processo wireshark no Router1, de maneira a capturar-se todos os pacotes que passam pela interface eth2.

Numa primeira fase, analisou-se a transferência da file1 por FTP. Para isso, dentro da *bash shell* do Server1 executou-se o comando "vsftpd /etc/vsftpd.conf -osecure_chroot_dir=/srv/ftp -oanon_root=/srv/ftp". Seguidamente, na *bash shell* do Laptop1, executou-se o comando "ftp 10.1.1.1", sendo necessário introduzir o nome e a palavra-passe.

Após estes procedimentos, transferiu-se o file1 através do comando "get file1". Uma vez concluída a transferência, terminou-se a conexão com o comando "quit".

O diagrama temporal obtido foi o seguinte:

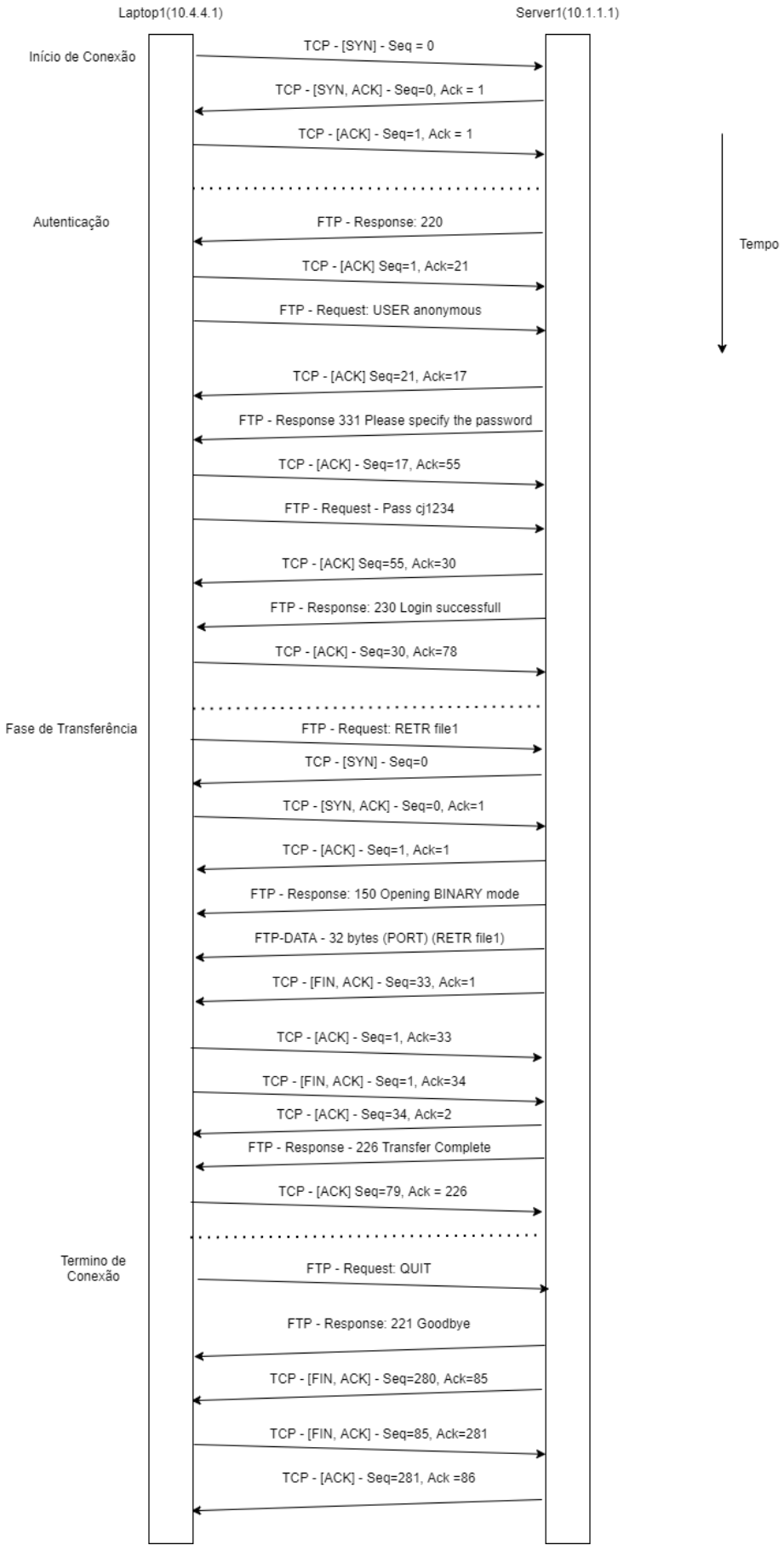

Figura 9: Diagrama Temporal da Transferência do file1 por FTP

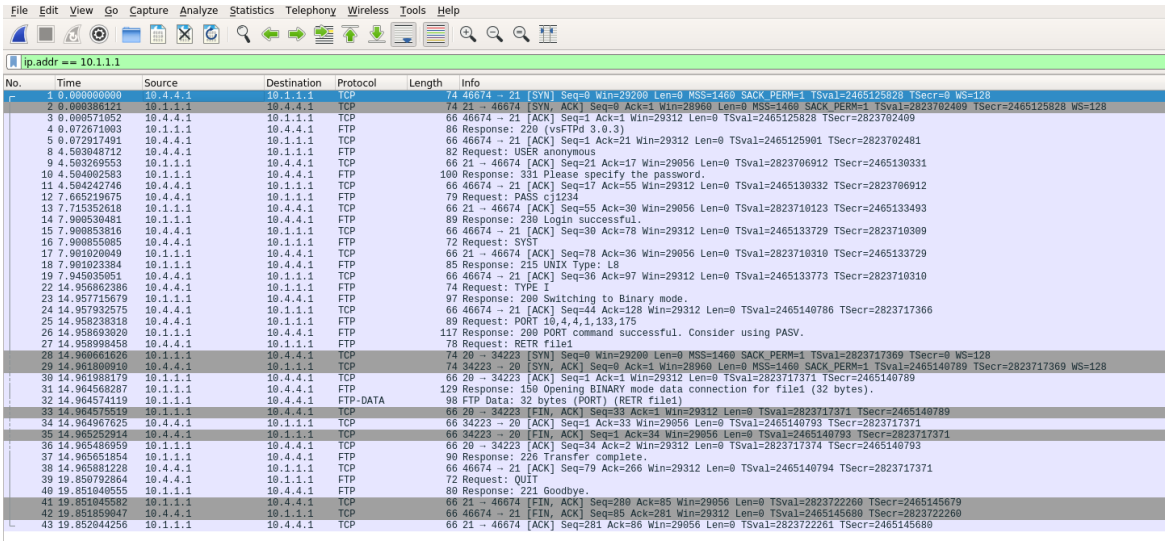

Figura 10: Resultado da Transferência do file1 por FTP

(ii) TFTP

Tal como no FTP, para a utilização do TFTP foi necessário realizar a mesma preparação. Contudo, dentro da bash shell do Server1 executou-se o comando "atftpd –verbose=3 –user root.ftp –logfile atftpd.log –bind-address 10.1.1.1 –daemon –no-fork /srv/ftp". Posteriormente, na bash shell do Laptop1, executou-se o comando "atftp 10.1.1.1".

Após tais procedimentos, realizou-se de novo a transferência do file1 usando o comando "get file1". Concluída a transferência, terminou-se a conexão com o comando "quit", o que permitiu obter o seguinte diagrama temporal:

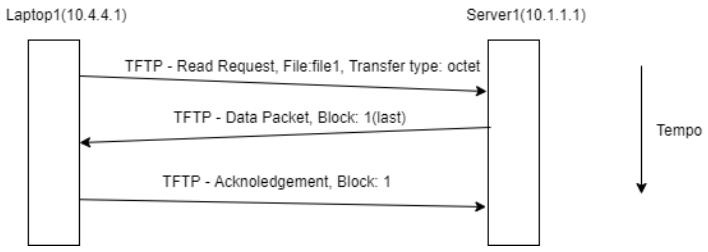

Figura 11: Diagrama Temporal da Transferência do file1 por TFTP

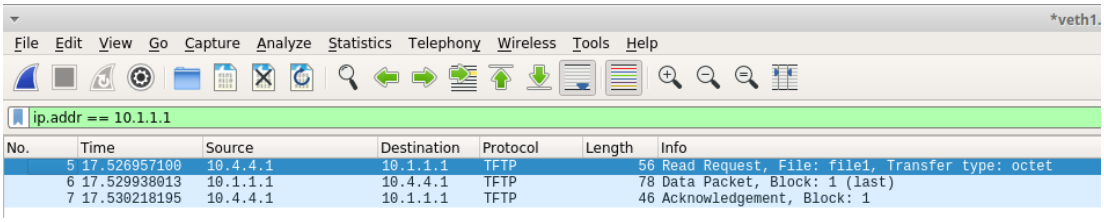

Figura 12: Resultado da Transferência do file1 por TFTP

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;

Com base nas experiências realizadas, identificou-se a utilização de duas camadas de transportes diferentes – TCP e UDP. Sendo que o TCP, ao contrário do UDP, é um protocolo de transporte orientado à conexão, oferecendo um serviço fiável.

Os protocolos que utilizam o TCP como camada de transporte são: SFTP, FTP e HTTP.

Apenas o protocolo TFTP, utiliza o UDP como camada de transporte.

Apesar de alguns protocolos utilizarem a mesma camada de transporte, cada um deles conecta-se a uma porta diferente. Assim temos:

Comando usado (aplicação)	Protocolo de transporte	Porta de atendimento
FTP	TCP	21
TFTP	UDP	69
HTTP	TCP	80
SFTP	TCP	22

(ii) eficiência na transferência;

Para analisar a eficiência das diferentes aplicações de transferência de ficheiros realizou-se a transferência do ficheiro 1 (com cerca de 147 bytes) e calculou-se para cada tipo de transferência o número de bytes transportado, do servidor para Laptop1, de modo a perceber o *overhead* introduzido, nos dados, por cada protocolo.

Conclui-se, que o TFTP não introduziu um peso extra nos dados transacionados, sendo por isso o mais eficiente. Ao contrário do TFTP, temos o SFTP que introduz um maior número de bytes em relação ao verdadeiro tamanho do ficheiro.

Por fim o FTP e o HTTP apesar de introduzirem um *overhead*, o seu valor não é tão elevado como no SFTP, sendo por isso mais eficientes.

Assim classificando do mais ineficiente para o mais eficiente temos: SFTP, HTTP, FTP e TFTP.

(iii) complexidade;

Ao analisar a complexidade de cada um dos protocolos percebe-se que em termos do programador/administrador o TFTP torna-se o protocolo mais complexo, não só devido à complexidade exigida pelos comandos necessários para transferir um ficheiro, mas também a nível de desenvolvimento de aplicações. Uma vez que o TFTP utiliza como camada de transporte

o UDP exige que o programador tenha um cuidado especial relativamente à perda de dados ou à introdução de erros nos mesmos.

Por outro lado, o SFTP, pode ser considerado de mais simples utilização, uma vez que a dificuldade dos comandos introduzidos é pequena. Utilizando o mesmo raciocínio percebe-se que o HTTP também não exige a utilização de comandos complexos, ao contrário do FTP.

Relativamente à complexidade do protocolo, percebe-se que o SFTP, como utiliza ligações SSH que permitem encriptar os dados transferidos, é considerado o mais complexo. O HTTP é um protocolo simples (uma vez que foi projetado com esse objetivo) permitindo que as mensagens possam ser lidas e entendidas por qualquer um, o que facilita o desenvolvimento de testes. Por fim o TFTP e o FTP a nível de protocolo também são simples.

(iv) segurança;

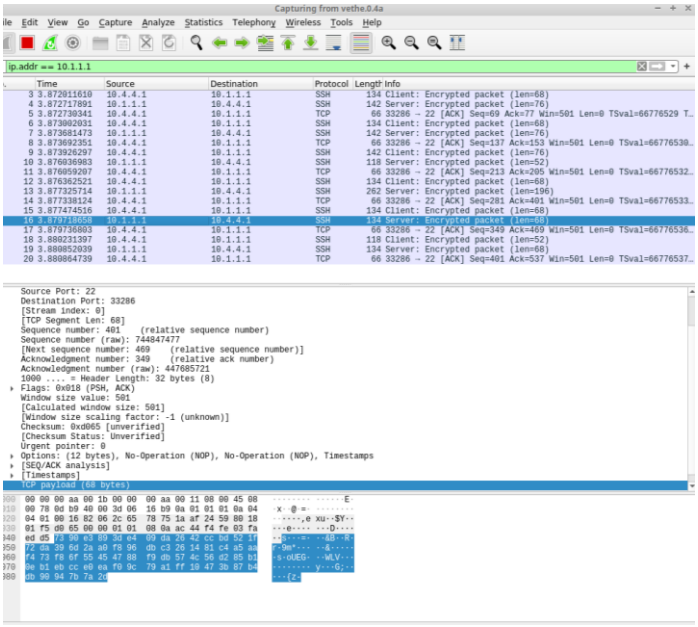

Figura 13: Dados encriptados, usando o SFTP

Ao realizar uma conexão com o servidor, utilizando os diferentes protocolos consegue-se perceber diferenças entre eles, relativamente à segurança. Desta forma tal como observado na figura 13, nota-se que o SFTP utiliza ligações SSH e necessita de autenticação. Também é de notar que os dados enviados são encriptados, garantindo uma maior segurança.

Ao contrário do SFTP os outros protocolos utilizados na experiência não encriptam os dados, sendo desta forma menos seguros.

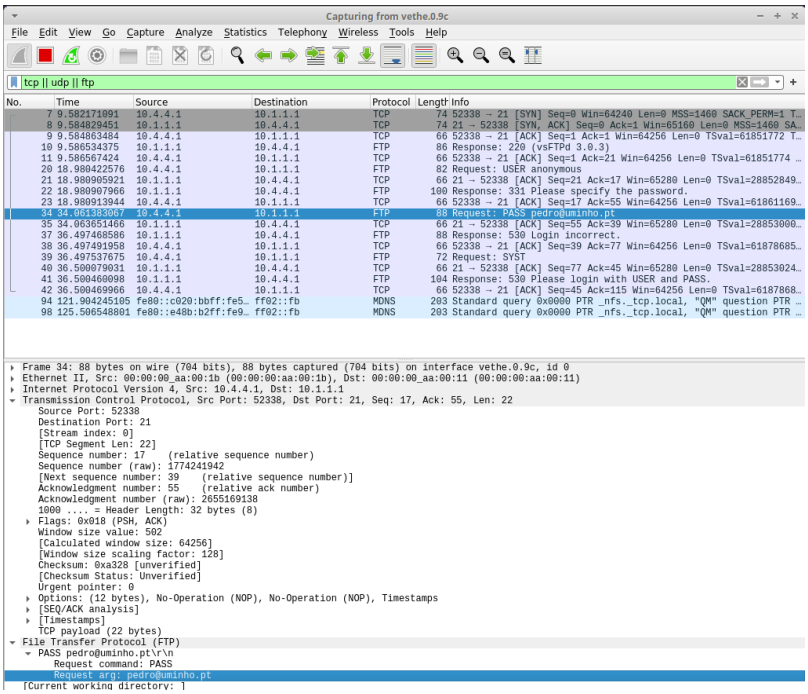

Figura 14: Insegurança do FTP

O FTP apesar de implementar um serviço de autenticação, permite que os dados circulem em texto claro na rede (observando a figura 14 percebe-se que a password utilizada para realizar o login é transportada na rede de forma legível), sendo por isso inseguro.

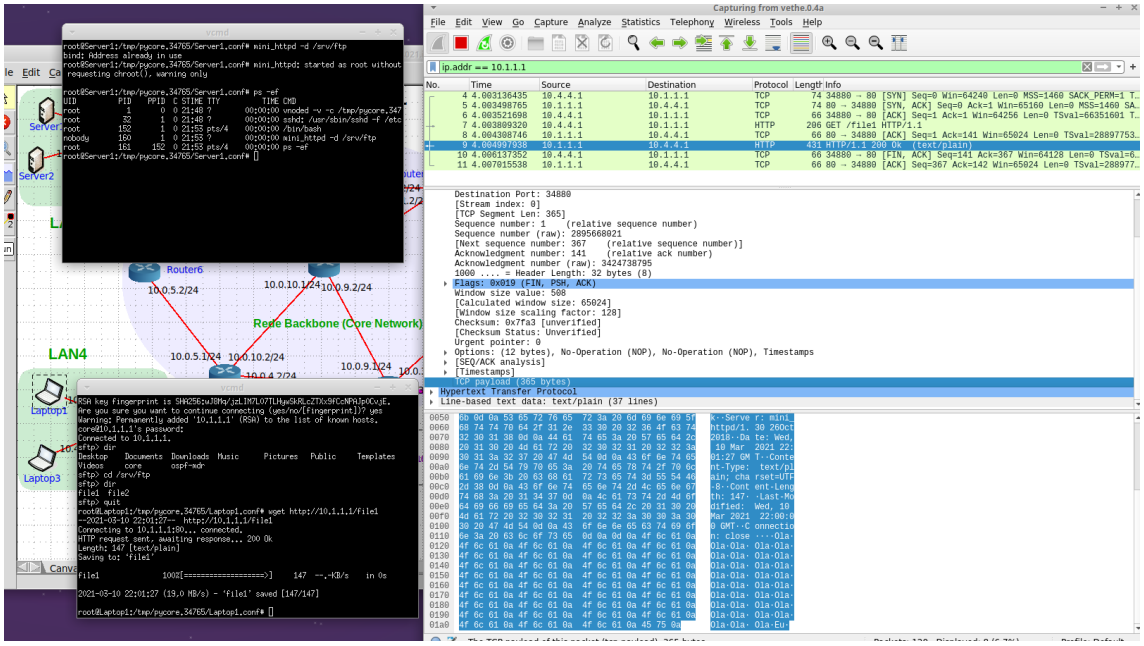

Figura 15: Utilização do HTTP

O HTTP é um protocolo inseguro, uma vez que também permite, a qualquer um, a leitura do conteúdo dos ficheiros, tal como observado na figura 15.

Por fim o TFTP, também é um protocolo bastante inseguro, uma vez que não aplica nenhum algoritmo para ocultar os dados transportados e não possui um serviço de autenticação.

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).

Figura 16: Verificação de perda de pacotes

Para a realização de aplicações fiáveis é necessário, uma atenção especial na escolha dos protocolos de transporte.

O protocolo de transporte a escolher deverá ser o TCP, visto que proporciona uma conexão fiável, garantindo que os pacotes enviados são recebidos no destino sem qualquer tipo de erro e pela ordem correta. De modo a garantir a conexão fiável, o TCP adota uma estratégia de enviar um *Acknowledgment* sempre que um pacote é recebido com sucesso. Uma vez que não existe mensagens de erro é também através do *Acknowledgment* que o emissor tem noção do ocorrido durante o envio.

Uma vez que a perda de pacotes ou a injeção de erros ocorre durante o transporte, as características do link terão influência no decorrer deste protocolo.

Tirando partido da experiência realizada, percebeu-se que devido à menor qualidade do link entre o Corvo e a *Backbone* ocorreram perdas de pacotes, como se pode observar na figura 16. Percebe-se a perda ao observar que o Corvo envia uma série de ACK com valor 39097 apesar de já ter recebido do servidor a sequência 43441. Assim, conclui-se que houve perda das sequências 39097, 40545 e 41993.

Uma vez que é necessário o reenvio do pacote quando este se perde ou chega ao destino com erros, existe uma perda de desempenho, causando mais sobrecarga na rede e atrasos na aplicação.

Por fim, o protocolo UDP, apesar de ser um protocolo não fiável também pode ser utilizado no desenvolvimento de aplicações fiáveis. Contudo passa a ser responsabilidade da aplicação verificar se os dados recebidos estão corretos ou se houve perda de pacotes.

A utilização do UDP, proporciona uma menor latência na ligação, uma vez que não realiza verificações de erros e não envia *Acknowledgment* após a receção de um pacote.

Conclusões

Com a realização do trabalho, conciliou-se as experiências práticas aos conhecimentos adquiridos nas aulas teóricas da disciplina, o que permitiu conciliar a matéria lecionada.

Percebeu-se a principal diferença entre o protocolo de transporte TCP do protocolo de transporte UDP. Uma vez que o TCP é um serviço orientado à conexão, oferece uma conexão fiável, garantindo que os dados que chegam à aplicação estão sem erros e por ordem correta. Ao contrário do TCP existe o UDP que não garante uma conexão fiável, sendo por isso trabalho da aplicação perceber se os dados recebidos estão sem erros e que não foram perdidos nenhuns pacotes pelo caminho. Contudo, o UDP garante uma menor latência e uma menor sobrecarga na rede.

Percebeu-se também, através da análise de casos práticos, o procedimento realizado, pelo TCP, quando existe a perda de pacotes. Identificou-se ainda, que a perda de pacotes é influenciada pela qualidade do *link* que percorre.

Por fim, o trabalho realizado permitiu diferenciar, os diferentes protocolos a nível de Aplicação, através da sua complexidade, segurança e tipo de camada de transporte utilizada.