

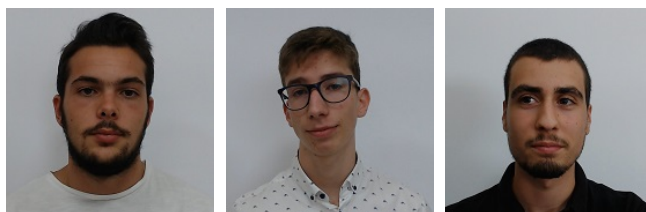


Universidade do Minho  
Mestrado Integrado em Engenharia Informática  
3ºano - 2º Semestre

Comunicações por Computador

## Trabalho Prático Nº.1 – Protocolos da Camada de Transporte

Grupo 1 - PL4



a83732 – Gonçalo Rodrigues Pinto  
a84197 – João Pedro Araújo Parente  
a84829 – José Nuno Martins da Costa

4 de Março de 2020

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Questões e Respostas</b>	<b>3</b>
2.1	Uso da camada de transporte por parte das aplicações . . . . .	3
2.2	Instalação, configuração e utilização de serviços de transferência de arquivos . . . . .	4
2.2.1	Representação num diagrama temporal das transferências da file1 por FTP e TFTP . . . . .	5
2.2.2	Distinção e comparação das quatro aplicações de transferência de arquivos utilizadas . . . . .	7
2.2.3	Influência de situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis . .	8
<b>3</b>	<b>Conclusão</b>	<b>9</b>

## Lista de Figuras

1	Topologia de Rede (backbone, acesso e local). . . . .	4
2	Diagrama temporal da transferência da file1 por FTP. . . . .	5
3	Captura Wireshark da transferência da file1 por FTP. . . . .	5
4	Diagrama temporal da transferência da file1 por TFTP. . . . .	6
5	Captura Wireshark da transferência da file1 por TFTP. . . . .	6

# 1 Introdução

O objectivo deste trabalho foi estudar os Protocolos da Camada de Transporte.

O serviço dos Protocolos de Transporte é disponibilizar uma ligação lógica entre aplicações (processos) que estão a ser executadas em Sistemas Terminais diferentes.

Os protocolos de transporte são executados nos Sistemas Terminais, na medida em que o emissor parte a mensagem gerada pela aplicação em segmentos que passa à camada de rede por outro o lado o receptor junta os diferentes segmentos que constituem uma mensagem que passa à respectiva aplicação.

A Camada de Transporte é fulcral pois fornece uma comunicação lógica entre processos e usa e melhora os serviços disponibilizados pela camada de Rede.

## 2 Questões e Respostas

### 2.1 Uso da camada de transporte por parte das aplicações

Executando numa linha de comando: *sudo wireshark*. Foi capturado o tráfego em determinados instantes que se considerou adequados, observando a forma como as várias aplicações utilizam os serviços da camada inferior. Foi possível preencher a seguinte tabela.

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)
<i>Ping</i>				
<i>tracert</i>		UDP	33434	8 (40-32)
<i>telnet</i>	TELNET	TCP	23	20
<i>ftp</i>	FTP	TCP	21	20
<i>Tftp</i>	TFTP	UDP	69	8
<i>brower/http</i>	HTTP	TCP	80	20
<i>nslookup</i>	DNS	UDP	53	8
<i>ssh</i>	SSH	TCP	22	20
<i>Outras:</i>				

## 2.2 Instalação, configuração e utilização de serviços de transferência de ficheiros

Nesta fase utilizou-se como o recomendado a topologia CC-Topo-2020.imn que se apresenta na figura 1 que descarregou-se da plataforma *elearning*.

Pretendeu-se transferir o mesmo ficheiro usando 4 serviços diferentes: SFTP, FTP, TFTP e HTTP, capturando todos os pacotes trocados durante a transferência utilizando o software Wireshark.

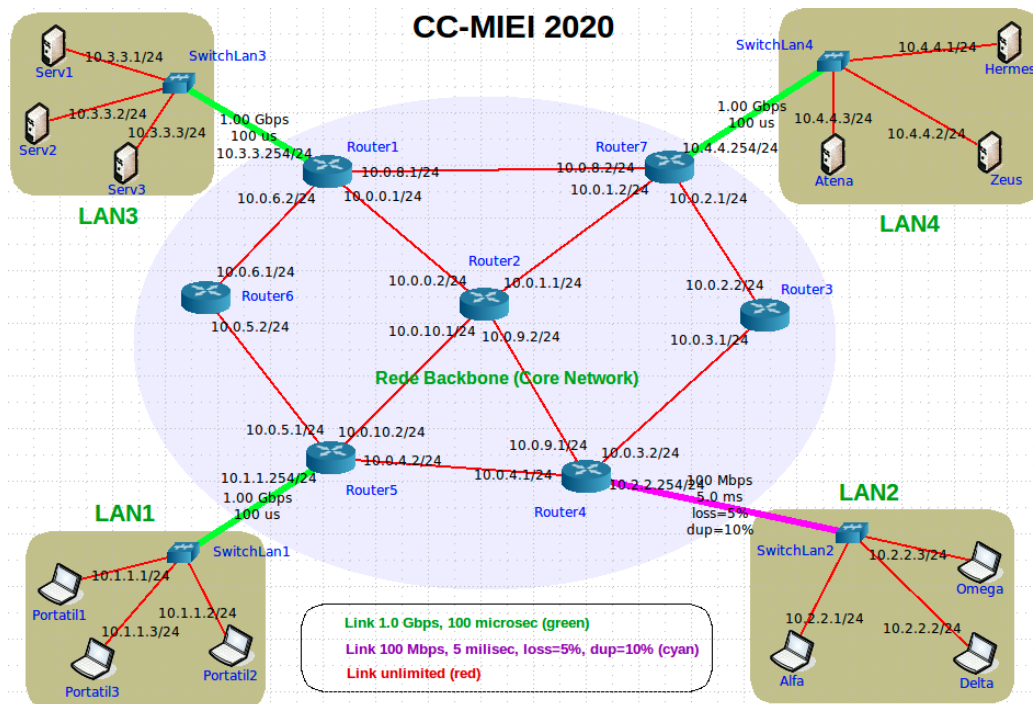


Figura 1: Topologia de Rede (backbone, acesso e local).

De forma atingir esse objectivo foi necessário verificar se o software (cliente e servidor) estava instalado; preparar uma pasta com os ficheiros a transferir (um ficheiro de texto e um ficheiro binário); executar o core com a topologia virtual CC-Topo-2020.imn.

Após as acções acima descritas foi possível transferir o mesmo ficheiro usando 4 serviços diferentes, como o pretendido.

## 2.2.1 Representação num diagrama temporal das transferências da file1 por FTP e TFTP

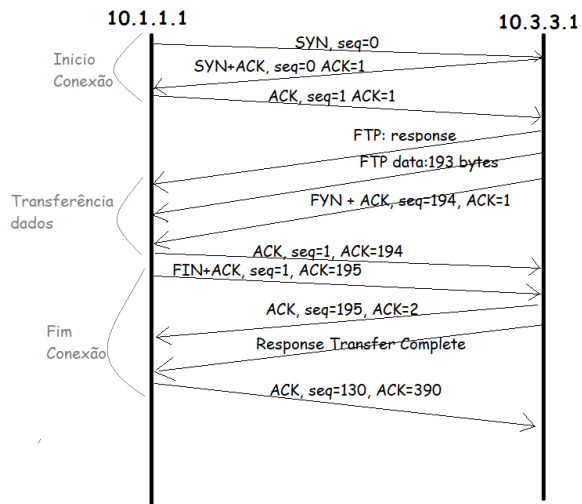


Figura 2: Diagrama temporal da transferência da file1 por FTP.

54	74.244474	10.1.1.1	10.3.3.1	FTP	74 Request: TYPE I
55	74.244972	10.3.3.1	10.1.1.1	FTP	97 Response: 200 Switching to Binary mode.
56	74.245255	10.1.1.1	10.3.3.1	FTP	89 Request: PORT 10.1.1.1,223,189
57	74.245860	10.3.3.1	10.1.1.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
58	74.246273	10.1.1.1	10.3.3.1	FTP	78 Request: RETR file1
59	74.246961	10.3.3.1	10.1.1.1	TCP	74 29 - 57277 [SYN] Seq=0 Win=14608 Len=0 MSS=1460 SACK_PERM=1 TSval=490655 TSecr=0 WS=16
60	74.247063	10.1.1.1	10.3.3.1	TCP	74 57277 - 29 [SYN, ACK] Seq=0 Ack=1 Win=14608 Len=0 MSS=1460 SACK_PERM=1 TSval=490655 TSecr=490655 WS=16
61	74.247206	10.3.3.1	10.1.1.1	TCP	66 29 - 57277 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=490655 TSecr=490655
62	74.247238	10.3.3.1	10.1.1.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
63	74.247240	10.3.3.1	10.1.1.1	FTP-DATA	259 FTP Data: 193 bytes (PORT) (RETR file1)
64	74.247280	10.3.3.1	10.1.1.1	TCP	66 29 - 57277 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=490655 TSecr=490655
65	74.248190	10.1.1.1	10.3.3.1	TCP	66 57277 - 29 [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=490655 TSecr=490655
66	74.248263	10.1.1.1	10.3.3.1	TCP	66 57277 - 29 [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=490655 TSecr=490655
67	74.248483	10.3.3.1	10.1.1.1	TCP	66 29 - 57277 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=490655 TSecr=490655
68	74.249294	10.3.3.1	10.1.1.1	FTP	90 Response: 226 Transfer complete.
69	74.250048	10.1.1.1	10.3.3.1	TCP	66 55898 - 21 [ACK] Seq=130 Ack=390 Win=14608 Len=0 TSval=490656 TSecr=490655

Figura 3: Captura Wireshark da transferência da file1 por FTP.

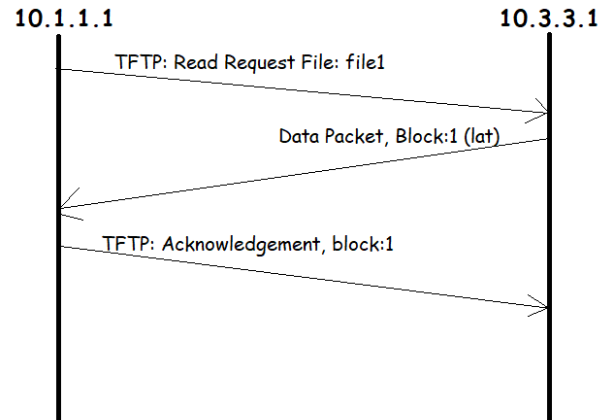


Figura 4: Diagrama temporal da transferência da file1 por TFTP.

54	74.244474	10.1.1.1	10.3.3.1	FTP	74 Request: TYPE I
55	74.244972	10.3.3.1	10.1.1.1	FTP	97 Response: 200 Switching to Binary mode.
56	74.245255	10.1.1.1	10.3.3.1	FTP	89 Request: PORT 10.1.1.1,223,189
57	74.245860	10.3.3.1	10.1.1.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
58	74.246273	10.1.1.1	10.3.3.1	FTP	78 Request: RETR file1
59	74.246961	10.3.3.1	10.1.1.1	TCP	74 20 -> 57277 [SYN] Seq=0 Win=14608 Len=0 MSS=1460 SACK_PERM=1 TSval=490655 TSecr=0 WS=16
60	74.247883	10.1.1.1	10.3.3.1	TCP	74 57277 -> 20 [SYN, ACK] Seq=0 Ack=1 Win=14608 Len=0 MSS=1460 SACK_PERM=1 TSval=490655 TSecr=490655 WS=16
61	74.247206	10.3.3.1	10.1.1.1	TCP	66 20 -> 57277 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=490655 TSecr=490655
62	74.247238	10.3.3.1	10.1.1.1	FTP	150 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
63	74.247240	10.3.3.1	10.1.1.1	FTP-DATA	259 FTP Data: 193 bytes (PORT) (RETR file1)
64	74.247268	10.3.3.1	10.1.1.1	TCP	66 20 -> 57277 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=490655 TSecr=490655
65	74.248100	10.1.1.1	10.3.3.1	TCP	66 57277 -> 20 [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=490655 TSecr=490655
66	74.248263	10.1.1.1	10.3.3.1	TCP	66 57277 -> 20 [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=490655 TSecr=490655
67	74.248463	10.3.3.1	10.1.1.1	TCP	66 20 -> 57277 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=490655 TSecr=490655
68	74.249294	10.3.3.1	10.1.1.1	FTP	90 Response: 220 Transfer complete.
69	74.250848	10.1.1.1	10.3.3.1	TCP	66 55898 -> 21 [ACK] Seq=130 Ack=399 Win=14608 Len=0 TSval=490656 TSecr=490655

Figura 5: Captura Wireshark da transferência da file1 por TFTP.

### 2.2.2 Distinção e comparação das quatro aplicações de transferência de ficheiros utilizadas

- **SFTP**

O *Secure File Transfer Protocol* (SFTP) é uma versão mais pesada e encriptada do FTP, que utiliza o protocolo de comunicação TCP. Por isso mesmo este protocolo é menos rápido que o FTP mas, em termos de segurança é superior.

- **FTP**

FTP (File Transfer Protocol) é um protocolo de transferência de ficheiros rápido e seguro que lhe permite através de um programa, enviar ficheiros e páginas web para o servidor. É um protocolo cliente/servidor que permite a transferência de ficheiros de um computador para outro, através de uma rede TCP/IP. Com o FTP, o utilizador tem à disposição uma configuração para login de acesso e senha e permite enviar directórios de documentos inteiros e de uma única vez, não tendo a menor necessidade de enviar documento por documento, o que leva a ganhar em velocidade e eficiência. Algo que torna a velocidade de transferência rápida em relação aos outros protocolos é que o FTP não coloca meta-data/overhead aos ficheiros enviadas é puro binário. Uma desvantagem do FTP em relação aos outros protocolos é cria duas conexões TCP, uma para envio de comandos e a segunda para o envio de a informação em si.

- **TFTP**

Trivial File Transfer Protocol (TFTP) é um protocolo que devido ao seu design simples é fácil de implementar, é um FTP mais simples que não proporciona autenticação de utilizadores e que usa UDP e por isso mesmo é um protocolo mais susceptível a perda de informações.

- **HTTP**

O Hypertext Transfer Protocol ao contrário do FTP reutiliza conexões TCP. Em transferência de ficheiros pequenos o HTTP é bastante eficiente, até mais eficiente do que o FTP, o mesmo não acontece em ficheiros de tamanho consideravelmente grande. Ao contrário de o FTP, o HTTP não requer nenhuma autenticação.



### **2.2.3 Influência de situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis**

As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. A perda de pacotes ocorre quando um determinado pacote não chega ao destino, isto pode acontecer devido a várias causas, se estivermos a falar de uma aplicação que use o protocolo de transporte TCP, como é o caso do FTP, este deteta as perdas, e se identificar uma, encarrega-se de fazer o pedido de retransmissão, isto leva a que bastantes pacotes sejam retransmitidos e assim sendo possivelmente provocar impacto no desempenho da aplicação, contudo é algo necessário para manter a informação fiável. No caso de uma aplicação que utilize UDP se existir uma perda de pacotes, não é possível recuperá-lo, isto em algumas aplicações como o caso do streaming não faz muita diferença mas em relação a aplicações fiáveis pode ser relevante.

Algo que também influencia o desempenho global de aplicações fiáveis são os pacotes duplicados. Este fator pode acontecer por várias razões, um cenário possível pode ser por exemplo termos um emissor e um recetor, o emissor envia dois pacotes para o recetor, o pacote 1 vai por um caminho na rede e o pacote 2 vai por outro caminho devido a routing, e por acaso o pacote 2 chega primeiro que o pacote 1, o recetor ao receber o pacote 2 antes do 1 deteta que lhe falta o pacote 1 e envia um ACK para notificar o emissor que lhe falta o pacote 1. O recetor ao receber este ACK envia de novo o pacote 1 tendo assim enviado dois pacotes do pacote 1. Se tivermos a falar de uma aplicação que use o protocolo de transporte TCP o recetor ao receber pela segunda vez o pacote 1, deteta que é duplicado e ignora-o simplesmente, se não for este caso e usar o protocolo de transporte UDP, o protocolo não tem forma de detetar que se trata de um pacote duplicado e entrega-o como informação válida á aplicação, o que claramente para aplicações fiáveis não é aconselhável.

### **3 Conclusão**

O presente relatório descreveu, de forma sucinta, a resolução das questões propostas utilizando os softwares disponibilizados pelos docentes.

Após a realização deste trabalho, ficamos conscientes dos vários aspectos do Protocolos da Camada de Transporte.

Consideramos que os principais objectivos foram cumpridos.

Sentimos que a realização deste trabalho prático consolidou os nossos conhecimentos dos Protocolos da Camada de Transporte..