

Comunicações por Computador (3° Ano, 2° Semestre)

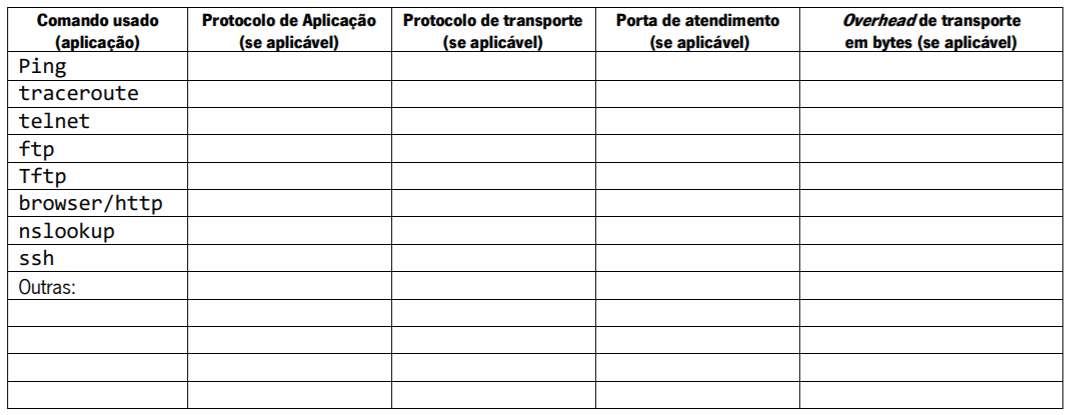
Universidade do Minho, Escola de Engenharia

CC-TP1-PL68

Diogo Araújo A78485

Diogo Nogueira A78957

# Questão 1



8 bytes

8 bytes

Porta 22

Porta 53

Porta 80

Porta 21

Porta 69

FTP

SSH

20 bytes

20 bytes

HTTP

DNS

20 bytes

≈ [Porta 33446 a 33523]

FTP

TCP

TELNET

Porta 23

TCP

20 bytes

8 bytes

UDP

DNS

Não Aplicável

Não Aplicável

Não Aplicável

Não Aplicável

??

TCP

UDP

TCP

UDP

# Questão 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.*

Cliente 1 (10.4.4.1)

Servidor 1 (10.1.1.1)

TFTP

TFTP – RRQ, File: file1

TFTP – Data Packet, Block: 1 (last)

TFTP – Acknowledgement, Block: 1

Como o file1 tem menos de 512 bytes, apenas foi enviado um pacote de dados (Block: 1) que é denominado como sendo então o último pacote. O Cliente faz ACK desse único pacote, terminando a transferência.

Servidor 1 (10.1.1.1)

FTP

Cliente 1 (10.4.4.1)

FTP - Request: RETR file1

TCP – ftp-data [SYN] Seq=0

Estabelecimento de conexão

TCP – ftp-data [SYN, ACK] Seq=0 Ack=1

TCP – ftp-data [ACK] Seq=1, Ack=1

FTP - Response: 150 Opening BINARY connection for file1

Transferência de Dados

FTP Data: 193 bytes

TCP – ftp-data [FIN, ACK] Seq=194 Ack=1

TCP – ftp-data [ACK] Seq=1 Ack=194

Término da Ligação

TCP – ftp-data [FIN, ACK] Seq=1 Ack=195

TCP – ftp-data [ACK] Seq=195 Ack=2

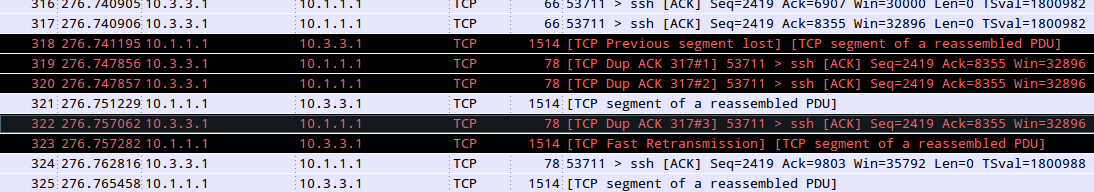
# Questão 3

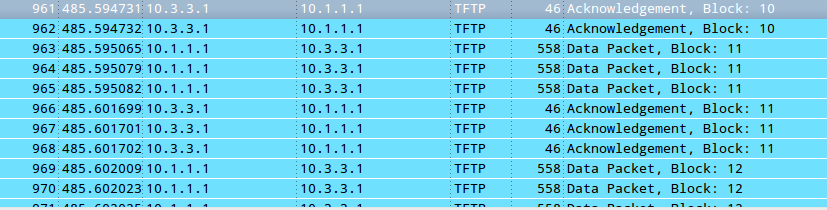
*Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança;*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Camada Transporte | Eficiência da Transferência | Complexidade | Segurança |
| SFTP | TCP | Igualmente confiável como o FTP, adicionado a segurança logo no protocolo | Mais complexo que o FTP devido à utilização do protocolo SSH | Segurança mais elevada porque encripta tanto os seus comandos como os seus dados, protegendo as informações sensíveis |
| FTP | TCP | Mais confiável, uma vez que não se perde a conexão ao transferir os dados (sequencialmente) e lento, dado que existe um controlo paralelo constante | Complexo | Existe segurança tal como utilizador e password, mas esta segurança é enviada sem qualquer encriptação, acabando por se tornar vulnerável a alguns ataques. Além de permitir um login anónimo |
| TFTP | UDP | Rápido dependendo do tamanho do pacote, dado que envia por UDP blocos de dados (aleatoriamente) que necessitam de uma resposta, podendo existir atrasos nessa espera | Mais simples | Não possui segurança ao nível da camada aplicacional |
| HTTP | TCP | Eficiência similar a outros que utilizem o protocolo TCP. Apenas perde uma conexão para o controlo melhorando a eficiência inicial | Menos complexo que o FTP dado que multiplexa a conexão de controlo e de dados | Não possui segurança por omissão, podendo adicionar um protocolo que execute esta propriedade (TLS) |

# Questão 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 influenciam fortemente a qualidade e fiabilidade dos níveis de Transporte e Aplicação. Como visto na topologia fornecida pelo docente no CORE, caso exista uma ligação de rede com perdas e duplicação de pacotes mais ou menos ocasionais, influencia a tecnologia dos protocolos de transporte e consequentemente os protocolos no nível de aplicação. Como exemplo, numa transferência usando *SFTP (Secure File Transfer Protocol)*, que utiliza como protocolo de transporte o *TCP (Transport Control Protocol)*. Dado que o *TCP* providencia um *stream* fiável, ordenado, mas principalmente importante, consegue verificar erros. Como visto no *print* abaixo, quando existe perdas de pacotes para o “*host Alfa”*, neste caso com ACK duplicados, verificamos que o *TCP* reconhece que é um duplicado, dada a sua natureza que verifica a ordenação (através do nº de sequência e também o nº de ACK).

 Usando o *TFTP (Trivial File Transfer Protocol)* como o protocolo aplicacional para uma transferência, notamos aí uma diferença dado que usa o protocolo de transporte *UDP (User Datagram Protocol)*. Como este protocolo não tem nenhum processo de *“hand-shake”*, ou nenhuma garantia e/ou proteção de entrega, duplicação e ordenação de datagramas. Assim, como visto no *print* abaixo temos a consequência destas situações. Ao enviar os blocos de dados através do *TFTP*, existe consecutivamente o envio do mesmo bloco, dado que há pacotes perdidos, neste caso os ACK necessários para ser enviado o próximo bloco de dados.

Para uma aplicação fiável, tem de se pensar no mais importante para o propósito, mas também nas diversas caraterísticas que as ligações de rede dos utilizadores terão. Uma melhor latência em detrimento de não existir garantia e proteção para os vários erros que podem ocorrer, pode não ser vantajoso dado que teria de existir um resguardo nestes pontos de outra forma, normalmente feitos em camadas superiores do modelo OSI.

# Considerações Finais de Aprendizagem

Com a aplicação prática de todos os protocolos abordados consegue-se perceber toda a eficiência, complexidade e funcionalidade de cada uma destas aplicações na transferência de ficheiros. Recorrendo-se a dois ficheiros previamente fornecidos, de tamanhos distintos fizeram-se experiências ao nível da camada de transporte e de aplicação e estabeleceu-se uma distinção entre a maneira como cada protocolo opera, as velocidades comparativamente a outros protocolos e ainda a segurança que oferecem.

A análise precisou de uma captura de tráfego pelo *Wireshark* que depoisteve de ser analisado e filtrado com o intuito de ajudar a perceber como cada protocolo estabelecia uma conexão prévia entre um Cliente e Servidor e a transferência do ficheiro em si. As dificuldades deste trabalho prático residiram sobre este componente – como são processadas e executadas as transferências de um mesmo ficheiro para protocolos aplicacionais opostos. Foi através desta descoberta que se definiram e ficaram a conhecer as regras de cada protocolo, as suas aplicações práticas e em que situações necessitam ser efetivamente utilizados.