

Comunicações por Computador

MIEI - 3º ANO - 2º SEMESTRE
UNIVERSIDADE DO MINHO

TP1 - PROTOCOLOS DA CAMADA DE TRANSPORTE

Grupo 13:

João Vieira, A76516
João Leal, A75569
Manuel Monteiro, A74036

February 25, 2019

Comando usado (aplicação)	Protocolo de Aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte em bytes
Ping	-	-	-	-
Traceroute	-	UDP	33434	8
Telnet	Telnet	TCP	23	20
FTP	FTP	TCP	21	20
TFTP	TFTP	UDP	69	22
Browser/HTTP	HTTP	TCP	80	20
nslookup	-	UDP	53	39
SSH	SSH	TCP	22	20
Outras:	-	-	-	-
SFTP	SSH	TCP	22	20

1 Questões e Respostas

1.1 Questão 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:

- Exemplo de cálculo do Overhead para UDP (traceroute):

```

7 2.060898 192.168.222.129 193.136.19.254 UDP 74 Source port: 49881 Destination port: traceroute

▶ Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▶ Ethernet II, Src: Vmware_41:a8:35 (00:0c:29:41:a8:35), Dst: Vmware_e3:f0:78 (00:50:56:e3:f0:78)
▶ Internet Protocol Version 4, Src: 192.168.222.129 (192.168.222.129), Dst: 193.136.19.254 (193.136.19.254)
▼ User Datagram Protocol, Src Port: 49881 (49881), Dst Port: traceroute (33434)
  Source port: 49881 (49881)
  Destination port: traceroute (33434)
  Length: 40
  Checksum: 0x5074 [validation disabled]
▶ Data (32 bytes)

```

$$\text{UDP Overhead} = 40 (\text{length UDP}) - 32 (\text{data}) = 8 \text{ bytes}$$

- Exemplo de cálculo do Overhead para TCP (telnet):

```

1 0.000000 192.168.222.129 193.136.9.33 TCP 74 59348 > telnet [SYN] Seq=0 Win=14600

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▶ Ethernet II, Src: Vmware_41:a8:35 (00:0c:29:41:a8:35), Dst: Vmware_e3:f0:78 (00:50:56:e3:f0:78)
▶ Internet Protocol Version 4, Src: 192.168.222.129 (192.168.222.129), Dst: 193.136.9.33 (193.136.9.33)
▼ Transmission Control Protocol, Src Port: 59348 (59348), Dst Port: telnet (23), Seq: 0, Len: 0
  Source port: 59348 (59348)
  Destination port: telnet (23)
  [Stream index: 0]
  Sequence number: 0 (relative sequence number)
  Header length: 40 bytes
▶ Flags: 0x002 (SYN)
  Window size value: 14600
  [Calculated window size: 14600]
  Checksum: 0x6a02 [validation disabled]
▶ Options: (20 bytes)

```

$$\text{TCP Overhead} = 40 (\text{length TCP}) - 20 (\text{Options}) = 20 \text{ bytes}$$

1.2 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.

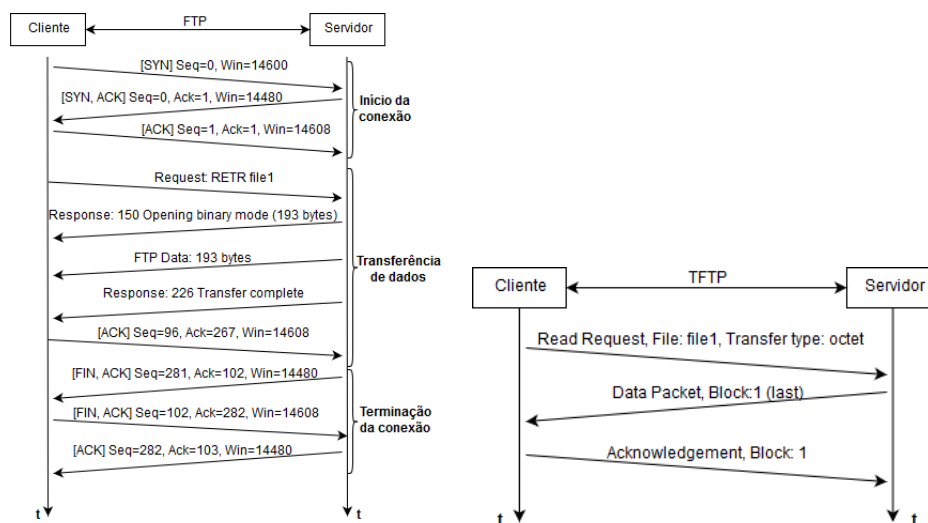


Figure 1: Diagrama FTP; Diagrama TFTP.

- Exemplo de captura para TFTP:

11	46.252333	10.4.4.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
12	46.256200	00:00:00_aa:00:16	Broadcast	ARP	42 Who has 10.1.1.254? Tell 10.1.1.1
13	46.256231	00:00:00_aa:00:12	00:00:00_aa:00:16	ARP	42 10.1.1.254 is at 00:00:00:aa:00:12
14	46.260100	10.1.1.1	10.4.4.1	TFTP	239 Data Packet, Block: 1 (last)
15	46.260395	10.4.4.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1

1.3 Questão 3

Com base nas experiências realizadas, distinga e compare sucinatamente 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;

As quatro aplicações de transferência de ficheiros usadas nas experiências realizadas nas aulas são: FTP, TFTP, SFTP e HTTP.

- **FTP:** Este protocolo utiliza TCP para o transporte da informação a transmitir. O desenvolvimento deste protocolo não foi feito com considerações de segurança e autenticação em mente, sendo que a data transmitida está descriptada e, portanto, vulnerável a qualquer tipo de interceptação da mensagem. Contudo, este protocolo mostra-se eficiente no transporte de ficheiros simples com grande tamanho, de A para B (visto que não existe autenticação ou encriptação/descriptação), não mostrando detalhes dos possíveis problemas que poderão ocorrer na transmissão destes mesmos ficheiros. Por fim, o FTP é um protocolo complexo visto que usa duas conexões separadas para o controlo de sessão e a transferência de dados.
- **TFTP:** O TFTP utiliza UDP para o transporte da informação, portanto, à partida, apresenta uma menor complexidade que o FTP. Relativamente à segurança, o TFTP não oferece qualquer tipo de autenticação ou segurança aquando da transferência de ficheiros, contribuindo para a baixa eficácia deste, tornando o seu uso perigoso em transferências pela Internet. Contudo, em computadores e dispositivos que não apresentem grande armazenamento, este protocolo pode ser facilmente implementado, uma vez que ocupa pouco espaço.
- **SFTP:** Este protocolo usufrui de TCP para o transporte da informação entre o cliente e o servidor, apresentando uma maior complexidade do que o FTP, uma vez que oferece mais funcionalidades, como por exemplo, o acesso, transferência e gestão dos ficheiros a transmitir. Em termos de segurança, este protocolo, em si, não oferece autenticação nem encriptação aquando da transferência de dados, recorrendo ao protocolo de aplicação (SSH) para tratar destes pontos, oferecendo assim a encriptação e a autenticação necessária para a viabilidade das transmissões. No entanto, uma vez que recorre a outro protocolo, este torna-se menos eficiente.
- **HTTP:** O HTTP é um protocolo que usa TCP no transporte de mensagens. Apresenta uma maior eficiência que o FTP, uma vez que usufrui de pipelining, o que torna a transferência de múltiplos ficheiros do mesmo servidor mais rápido. Além disso, o HTTP comprime automaticamente o ficheiro a transmitir, fazendo com que menos data seja enviada. Assim sendo, este protocolo apresenta uma maior complexidade que os acima mencionados, visto oferecer mais funcionalidades. Contudo, possui um baixo nível de segurança pois não encripta a informação a transmitir, o que impulsionou o desenvolvimento do HTTPS, em prol de corrigir esta falha de segurança.

1.4 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).

Ao nível de transporte pode-se dizer que o protocolo de transporte UDP funciona da mesma maneira numa conectividade com atrasos, perdas e duplicações, uma vez que este protocolo não faz controlo de fluxo e congestão, ou seja se o recetor não receber o pacote, ou recebe-lo com erros, o pacote é descartado, não sendo reenviado. Este protocolo serve para transferir pacotes em que a sua perda não prejudica o funcionamento da aplicação.

Ao nível de TCP, como já existe controlo de fluxo e congestão, quando nos encontramos numa conectividade com atrasos, perdas e duplicações, sendo os pacotes enviados por ordem, o remetente terá de reenviar o pacote de novo, caso este não tenha sido recebido, se forem encontrados erros, ou até quando o pacote é recebido fora de ordem, o que provoca uma maior confiança nos dados enviados através deste protocolo, porém também aumenta o tempo de transferência devido aos controlos de erros feitos, comparando com o protocolo UDP.

Por fim, o protocolo de aplicação é que deve escolher qual o protocolo de transporte que deverá ser utilizado, dependendo principalmente do objectivo da aplicação.

- Exemplo de trama ACK duplicada:

```
9 19.311507 10.3.3.1 10.1.1.1 TCP 66 49970 > ftp [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=3139609
10 19.311508 10.3.3.1 10.1.1.1 TCP 66 [TCP Dup ACK 9#1] 49970 > ftp [ACK] Seq=1 Ack=1 Win=14608 Len=0
```

2 Conclusões

Com este trabalho ficámos a perceber algumas diferenças entre os vários protocolos de aplicação existentes, como também, o método de funcionamento dos protocolos de transporte, nomeadamente TCP e UDP.

Além disso, vimos as principais distinções entre os protocolos de aplicação de transferência de dados, como os protocolos usados na camada de transporte, a eficiência de cada um, a sua complexidade e segurança.

Ainda foram feitos testes para verificar o comportamento dos diferentes protocolos numa conectividade mais perto da realidade (com perdas e duplicações de pacotes), para verificar a abordagem dos diferentes protocolos.

Em suma, este trabalho ajudou a aprimorar os conhecimentos que o grupo tinha acerca dos diferentes protocolos de aplicação e de transporte, abordados pelo mesmo.