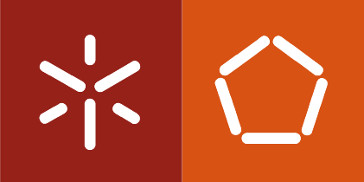
Comunicações por Computador

Mestrado Integrado em Engenharia Informática

Universidade do Minho



Trabalho prático nº1:

Protocolo da Camada de Transporte

PL4 Grupo nº 16

Gonçalo Costa de Almeida a88292

Maria Sofia Martinho Gonçalves Jordão Marques a87963

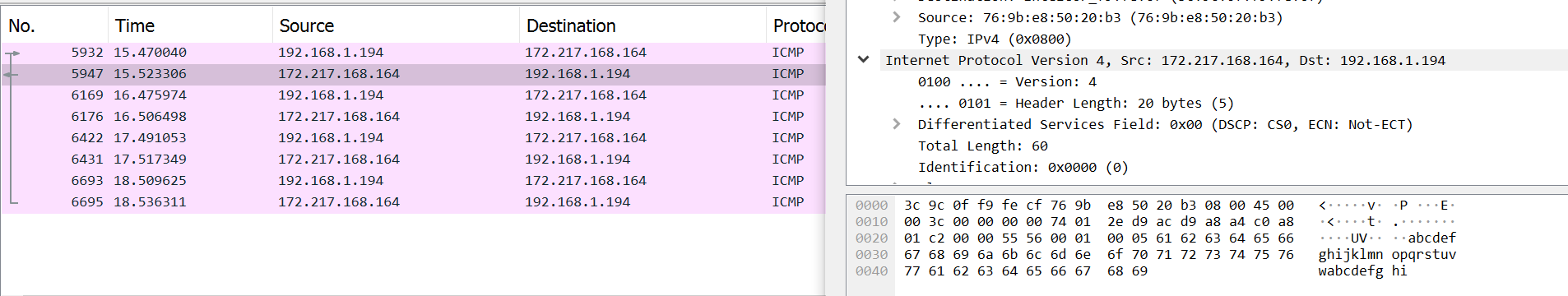
Pedro Nogueira Pereira a89232

**Resumo:** Análise do usa da camada de transporte por parte de aplicações

1. **Questões e Respostas**

**1.1. Inclua 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:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| Traceroute | DNS | UDP | 33446 | 8 Bytes |
| Telnet | TELNET | TCP | 23 | 20 Bytes |
| FTP | FTP | TCP | 21 | 20 Bytes |
| TFTP | TFTP | UDP | 69 | 8 Bytes |
| Browser/http | HTTP | TCP | 80 | 20 Bytes |
| nslookup | DNS | UDP | 53 | 8 Bytes |
| SSH | SSH | TCP | 22 | 20 Bytes |
| Outras: | **-** | **-** | **-** | **-** |



Uma imagem com mesa

Descrição gerada automaticamente

Figura 2 - Captura do traceroute

Figura 1 - Captura do Ping

Uma imagem com mesa

Descrição gerada automaticamente

Figura 1 - Captura do Telnet

Uma imagem com texto

Descrição gerada automaticamente

Figura 2 - Captura do FTP

Uma imagem com mesa

Descrição gerada automaticamente

Figura 5 - Captura do TFTP

Uma imagem com texto

Descrição gerada automaticamente

Figura 6 - Captura do Browser/http

Uma imagem com texto

Descrição gerada automaticamente

Figura 3 - Captura do nslookup

Uma imagem com texto

Descrição gerada automaticamente

Figura 4 - Captura do 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.**

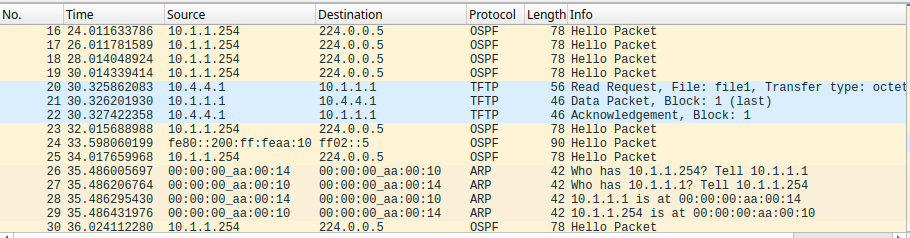


Figura 5 - Captura da transferência via TFTP

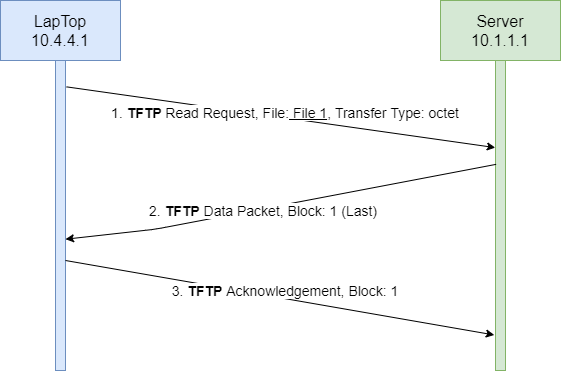


Figura 6 - diagrama temporal TFTP

Uma imagem com mesa

Descrição gerada automaticamente

Figura 11 - Captura da transferência via FTP

[

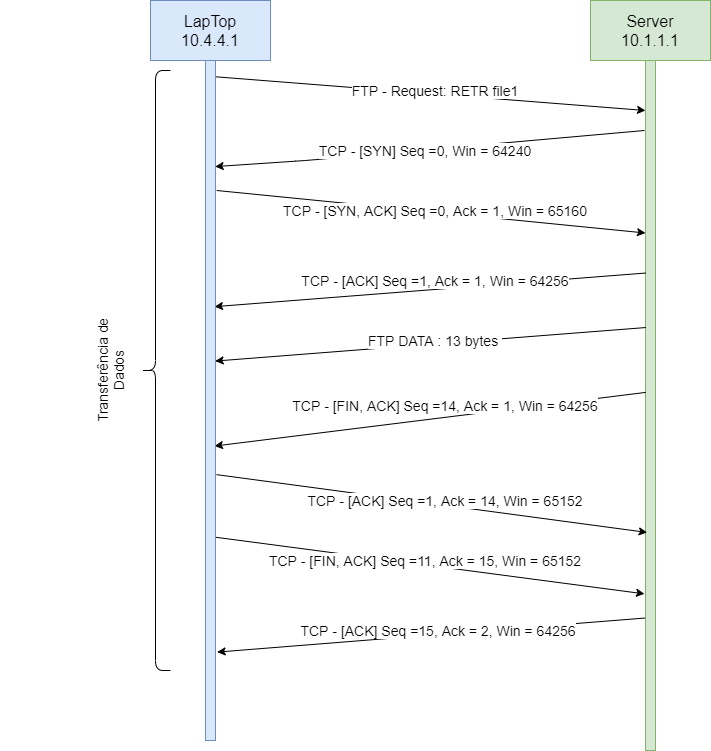


Figura 12 - Diagrama temporal FTP

Uma imagem com mesa

Descrição gerada automaticamente

Figura 13 - Captura ampliada do 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.**

**(i) Uso da camada de transporte**:

**SFTP** – Utiliza o protocolo de transporte: TCP.

**FTP** – Utiliza o protocolo de transporte: TCP.

**TFTP** – Utiliza o protocolo de transporte: UDP.

**HTTP** – Utiliza o protocolo de transporte: TCP.

**(ii) eficiência de transporte:**

Assumimos para este tópico que a eficiência se traduz tanto na capacidade de transmissão segura de ficheiros como também na velocidade da mesma, assim tendo isso em conta avaliamos da seguinte forma os seguintes protocolos aplicacionais:

**SFTP** – É um protocolo muito seguro, tal como o nome indica (*Secure File Transfer Protocol*), na transferência de ficheiros, no entanto essa segurança acarreta alguns custos a nível de velocidade de transmissão.

**FTP** – É um protocolo de transferência de ficheiros simples, no entanto em comparação com o protocolo SFTP não implementa nenhuma segurança adicional, quanto a velocidade de transmissão é bastante mais lenta que os restantes apresentados.

**TFTP** – No que toca à velocidade este é bastante melhor que os dois primeiros descritos em cima, no entanto e dado que usa o protocolo UDP, não é fiável aquando da transferência de ficheiros. Ainda assim trata-se de um protocolo eficiente.

**HTTP** – Este protocolo é bastante vulnerável à interceção de dados, dai não ser um protocolo muito seguro. Quanto à velocidade este é bastante eficiente.

**(iii)** **complexidade:**

**SFTP** – trata-se do protocolo mais complexo, um dos motivos é a sua elevada segurança, os utilizadores podem configurar passwords e os dados são encriptados, outro motivo é o uso do TCP em vez de UDP.

**FTP** - relacionando também com a sua segurança, este protocolo apresenta uma complexidade intermédia, pois apesar de existir autenticação, não existe qualquer tipo de encriptação (utiliza TCP em vez de UDP o que aumenta a sua complexidade).

**TFTP** - é o protocolo que apresenta menor complexidade, pois não apresenta autenticação nem encriptação, e para além disso é o único protocolo que utiliza UDP o que faz com que não haja qualquer tipo de confirmação por parte do recetor para o transmissor, tornado assim este protocolo ainda menos complexo.

**HTTP** - tal como FTP apresenta uma complexidade intermédia, pois usa autenticação, porém não apresenta encriptação, e também utiliza TCP em vez de UDP.

**(iv) Segurança:**

**SFTP** – este protocolo garante uma segurança elevada uma vez que os dados são encriptados através do protocolo SSH, apresentando também métodos de autenticação, um utilizador pode configurar passwords privadas para o pacote de dados enviado.

**FTP** – este protocolo, apesar de apresentar um método de autenticação é inseguro visto que não apresenta qualquer tipo de encriptação, a transmissão de dados é feita em formato texto.  Desta forma se alguém intercetar a mensagem transmitida terá acesso a toda a informação.

**TFTP** - não apresenta qualquer tipo de encriptação ou autenticação, estando assim mais suscetível a ataques, podemos concluir assim que se trata de um protocolo inseguro.

**HTTP** – tal como o protocolo FTP, este, apesar de possuir sistema de autenticação, não apresenta encriptação sendo os dados transmitidos em formato texto, o que faz com que qualquer pessoa que aceda à mensagem enviada tenha acesso a toda a informação, podendo assim fazer alterações indesejadas.

**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).**

Quando o nosso objetivo recai sobre o desenvolvimento de uma aplicação fiável é de extrema importância a escolha apropriada do protocolo de transporte.

Desta forma existem duas opções possíveis, o protocolo TCP ou o protocolo UDP.

No entanto quando nos referimos a aplicações fiáveis o nosso primeiro pensamento é o TCP, pois este fornece um serviço garantido, no qual a informação enviada pela aplicação alcançará, sem erros, duplicação e desordenamento, a camada correspondente na máquina destino. Isto porque este protocolo possui para além de mecanismos de controlo de erros, controlo de fluxo e congestionamento. O mecanismo de controlo de fluxo tem um papel importante no que toca a proteger o recetor de uma subcarga de pacotes, e o controlo de congestionamento contribui para a proteção da rede.

No entanto quando nos encontramos perante uma rede fraca a probabilidade de congestionamento e consequentes atrasos na transferência e perdas de pacotes provoca a necessidade de retransmissão de pacotes e mensagens de erro o que para alem de afetar diretamente o desempenho da aplicação também leva a duplicação de pacotes por vezes desnecessários.

Deste modo é fácil de ver a grande influência que as caraterísticas da rede têm neste protocolo.

Por outro lado, também pode ser usado o UDP. A sua vantagem cai no facto de este ser um protocolo direcionado a velocidade de transferência e não propriamente a fiabilidade de transmissão. Dessa forma perante uma rede fraca quando ocorre perdas de pacotes estas não afetam de forma drástica o funcionamento da aplicação. Como não há verificação na transferência de pacotes por parte do protocolo não há a possibilidade de reenvio dos mesmos o que significa que não vai haver pacotes duplicados, não sobrecarregando a rede, nem problemas quanto a congestão dados.

1. **Conclusão**

Na realização deste trabalho, foram realizados vários testes com a ajuda da topologia core e o *Wireshark* que nos permitiram observar os diferentes protocolos quer de aplicação quer de transporte.

Ao longo do trabalho, após as configurações e instalações necessárias, tivemos a oportunidade de utilizar diferentes serviços de transferência de ficheiros, sendo estes SFTP, FTP, TFTP e o HTTP, conseguindo assim entender melhor o seu funcionamento. Ficamos também a perceber os conceitos/diferenças entre *TCP* e *UDP* e as vantagens/desvantagens de cada um.

Assim resumidamente, consideramos este trabalho importante, na medida em que nos permitiu consolidar os conhecimentos sobre o funcionamento deste nível de rede e o modo como as diversas aplicações recorrem aos serviços da camada inferior, consolidando a matéria lecionada nas aulas teóricas.