

Redes de Computadores (2017/2018)

Trabalho Prático 1

2016/2017

Versão 1.0 de 2 de Outubro de 2017

O protocolo **FTP17**, desenvolvido para a realização deste trabalho, permite transferir um ficheiro de um cliente para um servidor de forma fiável. Este protocolo usa datagramas UDP para transporte de blocos de dados e de mensagens de controlo, de acordo com a seguinte especificação:

O cabeçalho comum a todas as mensagens contém os seguintes campos:

- **Operation Code** - um inteiro de 16 bits (um *short*), com o significado óbvio.
- **Sequence Number** - um inteiro de 64 bits (um *long*), com o significado de um número de sequência cumulativo e discutido em pormenor mais abaixo.
- **ScratchPad** - um total de 16 bytes, de conteúdo livre, que deve ser enviado pelo cliente ao servidor, e que este se limitará a copiar para a eventual mensagem de resposta.

O protocolo **FTP17** compreende 6 tipos de mensagens, identificados pelos seguintes valores do campo **Operation Code**:

- **UPLOAD** - mensagem que inicia uma transmissão, do cliente para o servidor, indicando também obrigatoriamente o nome do ficheiro (uma *string*) como parâmetro suplementar. O valor do campo **Sequence Number** deve ser 0. Esta mensagem é confirmada pelo servidor através de uma mensagem **ACK**.
- **ERROR** - mensagem de erro com o campo suplementar **Error Message** (uma *string*).
- **DATA** - bloco de dados, com o campo suplementar **Data Block**, um *array* de bytes com qualquer dimensão inferior ao permitido pela implementação do servidor.
- **ACK** - mensagem de ACK que contém o campo suplementar **Discarded** (um byte) que tem o valor 0 se a mensagem recebida foi guardada pelo servidor, e um valor diferente de 0 se o servidor a desprezou.
- **FIN** - mensagem que assinala o fim da transmissão, sem nenhum parâmetro suplementar. Em resposta a esta mensagem, caso o servidor não tenha ainda recebido todos os dados, responderá com um **ACK**. Caso contrário, responde com um **FINACK**.
- **FINACK** - mensagem enviado pelo servidor em resposta a um **FIN**.

O campo **Sequence Number** tem 8 bytes (um *long*) e a sua gestão e significado são inspirados do protocolo TCP:

- Nos blocos **DATA**, este campo contém o número de ordem do primeiro byte do bloco contido no pacote.
- Nas mensagens de **ACK**, enviadas pelo servidor, o valor do campo **Sequence Number** corresponde ao número de ordem do próximo byte do ficheiro que o receptor espera. Ou seja, os números de sequência enviados pelo servidor são cumulativos, tal como no protocolo GBN (*Go back N*).

A numeração dos bytes do ficheiro começa em 1. O número de sequência das mensagens **FIN** e **FINACK** são sempre a dimensão do ficheiro mais um. O problema de a dimensão do ficheiro exceder a capacidade de um *long* pode ser ignorado.

Materiais Fornecidos

É fornecido um servidor concorrente (ficheiros **FTP17Server.jar** e **FTP17Packet.java**) que executa o protocolo para receber os ficheiros enviados pelos clientes. O servidor é capaz de dialogar com clientes que enviam um ficheiro usando qualquer uma das seguintes filosofias de protocolo: *Stop & Wait*, *Go Back N*, e *Selective Repeat*

(quando o cliente utilizar para o efeito o campo **ScratchPad**). É também possível implementar outra versão (mais simples ou de maior desempenho) que misture facetas dos dois últimos protocolos.

A dimensão da janela de receção é um parâmetro opcional do servidor. Lançar o servidor com janelas de diferentes dimensões condiciona o seu comportamento perante a receção de blocos de dados fora de ordem ou fora da janela de receção.

É fornecida uma solução do cliente que utiliza o protocolo *Stop & Wait* (**Ftp17Client.java**).

Objectivos do trabalho

O objetivo deste trabalho é desenvolver um cliente que envie ficheiros para o servidor usando um protocolo que permita a transferência mais rápida que possível num conjunto de cenários que serão posteriormente definidos. Os estudantes têm de entregar uma solução que utilize pelo menos o protocolo de janela deslizante na versão *Go Back N* sem otimizações.

Para a testar o trabalho são ainda fornecidos:

- Ficheiros de teste.
- Instruções para realização de testes e para a entrega do trabalho.

Indicações complementares

O cliente a desenvolver deve usar a mesma forma de passagem de parâmetros e valores por omissão que o cliente fornecido. Poderá ser utilizado um valor de *timeout* fixo, no entanto, a utilização de um valor adaptativo, sendo opcional, será valorizado. Poderá ser utilizada uma janela de emissão de dimensão fixa, no entanto, a utilização de um valor ajustado automaticamente, sendo opcional, será valorizado. No caso da transferência terminar com sucesso, o cliente deve apresentar um relatório com a seguinte informação:

- Total de bytes transferidos (ou dimensão do ficheiro em bytes),
- número total de pacotes enviados com dados (incluindo repetições),
- tempo total que durou a transferência em segundos com 3 casas decimais,
- velocidade de transferência média extremo a extremo em Mbps com 3 casas decimais,
- valor (médio) do RTT (e seu desvio padrão) em milissegundos com 3 casas decimais, e
- dimensão da janela de emissão em bytes, ou valor médio da dimensão da janela de emissão em bytes usada se variável.

A classe **Stats** do cliente fornecido facilita a apresentação deste relatório.

A forma como o protocolo for implementado e a qualidade da implementação também serão consideradas para efeitos de avaliação. A nota máxima de um trabalho sem opções (solução *Go Back N* usando apenas com números de sequência cumulativos e sem otimizações, janela e valor do *timeout* fixos) será de 15 valores.

Prazo de entrega, formato do relatório, procedimentos para entrega do trabalho e avaliação com teste

Consultar os avisos no CLIP e ainda as seções **Organização > Calendário > Testes** e **Documentação de apoio > Protocolos**.