Redes de Computadores (2014 / 2015)
Trabalho Prático nº 1

Sumário

Este trabalho consiste na implementação de um protocolo para suportar transferências de ficheiros em UDP usando um protocolo de janela deslizante com "Go-Back-N" no emissor e janela igual a 1 no receptor. O servidor deve ser capaz de fazer a transferência para o cliente fornecido pelos docentes. O trabalho também inclui uma comparação com o protocolo TCP. Os estudantes têm nesse caso que desenvolver o servidor TCP capaz de receber um ficheiro enviado pelo cliente TCP fornecido pelos docentes.

1. Introdução

Neste trabalho pretende-se implementar dois conjuntos cliente/servidor para a transferência fiável de ficheiros de um cliente para um servidor. O primeiro par usará UDP. Neste caso o cliente é dado pelos docentes, que corresponde ao trabalho anterior já proposto. O servidor deverá ser desenvolvido pelos estudantes. O servidor transmitirá ficheiros para o cliente. O segundo par usará o protocolo TCP. Neste caso o cliente é dado pelos docentes e os estudantes deverão desenvolver o servidor mas é o cliente que transmite o ficheiro para o servidor.

O primeiro cliente e servidor usarão um protocolo de janela deslizante com recuperação de falhas do tipo "Go-Back-N" e *timeouts* sobre UDP.

Procure obter o melhor desempenho possível quando o seu servidor responde a um pedido de transferência do cliente fornecido. O cliente fornecido tem o nome "FTUDPClient.java" (código Java incluído no arquivo "Trabalho1-20142015.zip" que está acessível no CLIP). Note que o cliente UDP fornecido tem janela de recepção de dimensão igual a 1 e apenas retransmite o ACK do último pacote corretamente recebido.

Para ajuda ao seu desenvolvimento é fornecido um servidor mais simples que transmite ficheiros usando o protocolo "Stop&Wait". O código Java deste servidor simples tem o nome FTUDPServer.java e está também incluído no arquivo "Trabalho1-20142015.zip" que está acessível no CLIP).

O cliente TCP é fornecido pelos docentes e está disponível no ficheiro "FTTCPClient.java" (também incluído no arquivo "Trabalho1-20142015.zip"). Devem realizar a transferência do mesmo ficheiro através do protocolo TCP para um servidor a desenvolver pelos estudantes e de nome "FTTCPServer.java".

Os testes a realizar serão feitos sempre no mesmo computador, ou entre computadores distintos geralmente ligados através de um canal *wireless*. Para tornar os testes mais realistas, quando se testa a transferência através do cliente que usa UDP dentro no mesmo computador, será utilizada uma classe especial de

Sockets Datagrama que simula o funcionamento de uma rede com perca de pacotes e tempo de transito e *jitter* significativos. Essa classe tem por nome "MyDatagramSocket.java" (também incluída no arquivo "Trabalho1-20142015.zip").

2. Requisitos e opções de funcionamento

2.1 Especificação dos cliente e servidor UDP

Tal como o fornecido, o servidor FTUDPServer dos estudantes será invocado sem parâmetros.

Com o servidor a desenvolver, o ficheiro deve ser transferido em blocos de 512 bytes, ou outro valor, usando uma janela de 10 blocos, por exemplo. No caso de a transferência terminar com sucesso, o servidor deve afixar a seguinte informação:

Total de bytes transferidos (dimensão do ficheiro)
Número total de pacotes enviados com dados (incluindo repetições)
Número total de pacotes recebidos com ACKs
Tempo total que durou a transferência
Valor médio do RTT calculado (opcional)

O cliente FTUDPClient fornecido não aceita opções mas poderá estendê-lo para passar a suportá-las. Neste caso deverá ser invocado da seguinte forma:

java FTUDPClient filename host [block_size [timeout]]

onde:

<filename> é o nome do ficheiro a descarregar do servidor
<host> é o nome do computador onde executa o servidor
<block_size> tamanho do bloco que se propõe usar;
na sua ausência o valor será de 512 bytes
<timeout> valor do *timeout* inicial a usar pelo servidor; na sua ausência
deverá ser usado o valor de 2 segundos

A implementação das opções *block_size* e *timeout* no cliente, a utilização de um valor de *timeout* adaptativo no servidor e a variação da dimensão da sua janela são requisitos opcionais mas valorativos. Outra opção valorativa diz respeito à forma como o servidor está estruturado e será apresentada a seguir.

2.2 Especificação do servidor TCP

O servidor TCP a desenvolver deverá ser capaz de receber um ficheiro transmitido pelo cliente TCP fornecido pelos docentes, e deverá chamar-se necessariamente "FTTCPServer.java". O servidor deve obrigatoriamente escrever o ficheiro recebido no disco.

2.3 Sugestão de metodologia sugerida para o desenvolvimento do trabalho

Para a realização do trabalho sugere-se que siga as seguintes fases, que deve tentar desenvolver sequencialmente.

Comecem por estudar o cliente e servidores fornecidos de forma perceberem como o protocolo "Stop&Wait" está implementado, assim como a forma como o servidor aceita opções no pedido inicial de leitura.

Se quiserem podem começar imediatamente a implementar as opções no cliente mas é preferível concentrarem-se na parte obrigatória do trabalho que consiste em introduzir no servidor UDP o protocolo de janela deslizante.

Para esse efeito comecem a implementar uma solução onde cada bloco enviado é guardado numa estrutura que representa a janela. Em cada momento em que é possível enviar blocos, enviam-se todos os blocos existentes na "janela" e que ainda não foram transmitidos. A cada ACK recebido, todos os blocos até ao confirmado pelo ACK são removidos da janela e novos blocos serão colocados na mesma e enviados até voltar a "encher a janela". A cada *timeout*, devem retransmitir os blocos na "janela". Testem o correto funcionamento usando o cliente fornecido.

Opcionalmente, melhorem o servidor introduzindo um mecanismo de cálculo de um valor dinâmico de *timeout* e melhorem o cliente introduzindo as opções.

Numa fase posterior poderão realizar uma versão paralela do servidor (paralela no sentido em que é realizada usando *threads)* para tentar optimizar o seu funcionamento, usando estruturas de dados partilhadas entre os diferentes *threads*. Adicionalmente poderão igualmente implementar uma versão em que a dimensão da janela varia de forma dinâmica, inspirando-se para esse efeito na estratégia de controlo da saturação do protocolo TCP.

Um servidor que utilize o protocolo de janela deslizante é o requisito mínimo deste trabalho. Todas as outras extensões são valorativas.

Finalmente, implemente o servidor TCP, "FTTCPServer.java", para o cliente fornecido que, usando o protocolo TCP, permite a transferência de ficheiros do cliente para o seu servidor. Como pode ver no código fornecido, o cliente envia primeiro, pelo canal TCP, o nome do ficheiro representado numa sequência de caracteres ASCII terminada por '\0' (byte com valor zero), seguindo-se depois todos os bytes no ficheiro, sendo a conexão fechada quando a transferência termina.

3. Medidas a realizar

Avalie o tempo de transferência de um ficheiro com 1 Mbyte para ambos os protocolos, nos cenários indicados abaixo.

Cenário 1 – Clientes e servidores no mesmo computador sem usar a classe MyDatagramSocket.

Cenário 2 – Clientes e servidores em computadores diferentes sem usar a classe MyDatagramSocket. Os computadores devem estar ligados através da rede sem fios da FCT/UNL.

Cenário 3 – Cliente FTUDPClient (COM classe MyDatagramSocket) e servidor FTUDPServer no mesmo computador. Teste usando o servidor com janelas de dimensão 1 (equivalente ao "stop & wait"), de 10 e de 20 blocos. Os parâmetros a usar na classe MyDatagramSocket são:

```
private static final int _PROB_MSG_LOSS     = 50;  // 5%
private static final int _IF_BANDWIDTH      = 10000000 ; // 10 Mbps
private static final double _PPT_JITTER     = 0.10 ; // jitter factor
private static final double _PPT_AVERAGE  = 10 ; // average propagation time
```

Cenário 4 – Idem cenário 3 mas os parâmetros a usar na classe MyDatagramSocket são:

```
private static final int _PROB_MSG_LOSS     = 100;  // 10%
private static final int _IF_BANDWIDTH      = 1000000 ; // 1 Mbps
private static final double _PPT_JITTER     = 0.10 ; // jitter factor
private static final double _PPT_AVERAGE  = 100 ; // average propagation time
```

Para a realização das medidas poderão desligar todas as eventuais mensagens de *debug* afixadas na consola pois a existência das mesmas poderá atrasar a transferência pois apenas interessam as medidas finais apresentadas.

Deverão recolher pelo menos uma medida para cada caso. No entanto, caso desejem ser mais rigorosos, deverão repetir cada medida umas 5 a 10 vezes e calcular os valores médios obtidos.

4. Relatório do trabalho

Os resultados deverão depois ser apresentados no formato e da forma indicada no modelo de relatório.