

Trabalho Prático do Curso de Redes de Computadores - Pós - 2017

Universidade Federal de Mato Grosso do Sul

Hana Karina Salles Rubinsztein

1 Introdução

O objetivo deste trabalho é fazer um programa que implemente algumas das funções do protocolo TCP. Mais especificamente você vai implementar um protocolo confiável tipo Go-Back-N para transferência de arquivos, usando o protocolo UDP para estabelecer uma conexão de um processo a outro.

2 Funcionamento do Programa

Você deve fazer um aplicativo para transferir um arquivo. O cliente solicita um arquivo ao servidor, que então o entrega de forma confiável. Arquivos podem ser de texto, imagem, áudio ou vídeo. Este aplicativo deve implementar as funções de transferência confiável de informação, que por sua vez usará o protocolo UDP para conectar cliente e servidor, conforme mostra a figura abaixo.

Ao iniciar, o programa de transferência de arquivo deve receber o nome do arquivo a ser transferido, e informações do servidor. O programa implementa o protocolo Go-Back-N conforme descrito no livro-texto.

Você também deverá implementar um módulo de perdas/atrasos que implementa as funções descritas abaixo:

- Dividir a informação a ser transmitida em segmentos cujo tamanho máximo deve ser igual a máxima “carga de dados” (*payload*) do IP. Cada segmento deve conter um *header*. No *header*, você deve colocar o número de sequência do pacote e quaisquer outras informações que julgar necessário.



Figura 1: Diagrama dos módulos envolvidos no trabalho

- Você poderá testar tanto o cliente quanto o servidor rodando-os numa mesma máquina ou em uma rede local. Nestes dois cenários, não existem perdas ou atrasos. Dessa forma, você deverá implementar um módulo (ver Figura 1) que emulará atrasos e perdas de pacotes aleatórios.

Este módulo, recebe um segmento da camada UDP e o atrasa de um valor aleatório igual a $RTT/2 + X$, sendo RTT fixo e X uma variável aleatória com distribuição exponencial e média $E[X]$ (procure saber como você implementa uma distribuição exponencial a partir da função de randomização de C ou outra linguagem). O módulo também escolhe se entrega ou não um pacote à camada acima. Isto é, um pacote é perdido com probabilidade p (procure também saber como você implementa essa probabilidade).

Para implementar os atrasos aleatórios, você deve criar uma lista de segmentos da forma descrita a seguir. Quando um segmento é recebido, o tempo atual t é lido. A este tempo somamos $RTT/2 + X$, onde X é aleatório. O novo tempo $t_n = t + RTT/2 + X$ é colocado numa lista de tempos com um indicador do segmento a que o tempo está associado. A lista é organizada por ordem de tempo. Os pacotes são entregues ao módulo superior quando o tempo atual for igual a t_n .

Você pode saber quando entregar os segmentos de duas formas: (1) Criando um loop infinito no seu programa que verifica a cada t unidades de tempo se o tempo atual é igual a t_n e se for, entrega o segmento. (2) Disparando um *timer* com o valor de t_n cada vez que houver uma inclusão de um elemento na primeira posição da lista. Quando o *timer* expirar, o segmento deve ser entregue ao módulo superior. A solução

(1) é mais simples mas é computacionalmente mais custosa.

Desta forma, a camada de emulação do protocolo confiável receberá segmentos com atrasos aleatórios e também, segmentos podem ser perdidos. Por outro lado o mesmo processo de atraso e perda deve existir para o recebimento de acks.

- O programa de transmissão confiável deve ler de um arquivo: o valor da janela do Go-Back-N e o valor do *timeout* de transmissão. O programa de emulação de perdas deve ler de um arquivo o valor de p , o valor de RTT e o valor de $E[X]$.
- O seu programa deve ter como saída a vazão (*throughput*) da transferência de arquivo e o número de segmentos retransmitidos. O objetivo é traçar os seguintes gráficos: vazão versus p , vazão versus RTT e vazão versus tamanho da janela.

O arquivo de teste deve ser longo para que possamos ter uma ideia do comportamento do protocolo de janela. O seu relatório deve ter conclusões sobre o trabalho, baseadas nas medidas acima. Para isso você deve variar p , RTT e o tamanho da janela. Faz parte do projeto a escolha coerente destes parâmetros de forma a se conseguir fazer um estudo do comportamento protocolo de janela.

3 Relatório

O relatório deverá ser composto por:

- Introdução: Nesta seção você deve descrever em linhas gerais o objetivo do seu trabalho.
- O Programa: Aqui você deve descrever em linhas gerais como funciona o seu programa, explicando as decisões adotadas no protocolo. Exponha também se algo no trabalho não foi implementado, ou se há alguma funcionalidade extra. Indique como executar os programas.
- Resultados: Neste item você deve fazer um estudo do Go-Back-N variando o tamanho da janela, RTT e p , e plotar gráficos conforme indicado acima.
- Conclusão: Nesta seção você deve expor suas conclusões sobre o trabalho.

Obs: A qualidade de seu relatório é tão importante quanto o bom funcionamento de seu programa.

4 Entrega

O trabalho deverá ser submetido eletronicamente pelo moodle, onde consta também a data de entrega.

Cada grupo (de até 2 participantes) deve entregar um arquivo tipo zip ou tar.gz, contendo os arquivos fontes do trabalho (e o makefile) e um relatório. São aceitos trabalhos em C/C++, Java ou Python.

Observações:

- Aqueles que optarem por Java ou Python serão avaliados com mais rigor, em especial quanto a interface do programa e facilidade de utilização.