Redes de Computadores
Trabalho: Protocolo de Transferência Confiável de Arquivos

Bruno Tomél[1], **Cláudio Menezes**[1], **Ronan Nunes**[1]

[1]Instituto Federal de Minas Gerais - Campus Formiga – (IFMG)
Rua Padre Alberico, 440 - São Luiz - Formiga MG - 35570-000 - Minas Gerais - Brasil

ibrunotome@gmail.com.br, claudiomenezio@gmail.com, ronannc12@gmail.com

Resumo. *O propósito deste trabalho foi a implementação de um protocolo de transferência confiável de arquivos em rede, seguindo a arquitetura cliente/servidor.*

1. Relatório

O trabalho foi baseado no javachat-v2 feito em sala de aula, visto que bastou alterar de string para um vetor de bytes[] para que qualquer tipo de arquivo pudesse ser transferido pela rede.

A arquitetura usada é a cliente/servidor, onde um servidor multicliente faz conexão com vários clientes ao mesmo tempo. Na camada de aplicação, os sockets TCP fazem o papel de transporte de datagramas de um host para o outro. E na camada de rede temos o endereçamento via IP e porto.

Definimos o buffer com o valor 1472, devido a testes realizados a partir desse tutorial: http://www.dslreports.com/faq/5793, onde tentamos passar a maior quantidade possível de dados sem fragmentação, definindo o tamanho da MTU. Tendo lembrado de acrescentar + 28 pelo cabeçalho do IP e ICMP.

1.1. Pontos fortes

- Conseguimos em um teste realizado no laboratório, com rede cabeada, fazer uma transferência de um vídeo de 288 megabytes em 68 segundos, e observamos que com a rede cabeada as percas eram quase inexistentes.
- O fato de utilizarmos sockets TCP torna a aplicação confiável, ou seja, os dados requisitados pelo cliente sempre chegarão até ele (retransmissão), de forma rápida ou não.
- O item extra relativo a criptografia foi implementado, ou seja, mais segurança na transferência dos arquivos.

1.2. Pontos fracos

- Não fizemos a alteração recomendada pelo professor, de alterar o flush() (descarga dos dados) para uma bufferização que poderia ter resultado em taxas melhores de transferência.
- A criptografia utilizada foi a DES (DES/ECB/PKCS5Padding), que está ultrapassada em relação a outros algoritmos como o AES por exemplo. Arquivos grandes demoram significativamente a serem encriptados/descriptografados.

2. Conclusão

Este foi um ótimo trabalho implementado, em trio pudemos discutir as melhores decisões de projeto, qual o tamanho de buffer, qual decisão tomar e porquê tomá-la. Além de ter sido um trabalho relativamente fácil de ser implementado, pois o java possui as principais funções já implementadas, bastando saber quando e onde usá-las.