

# Cliente e Servidor UDP com transmissão confiável de dados

Davi Maciel Chamma Santos<sup>1</sup>, Flávio Augusto Muller Shinzato<sup>1</sup>

<sup>1</sup>Faculdade de Computação – Universidade Federal do Mato Grosso do Sul (UFMS)

davichamma97@gmail.com, f.m.a.shinzato@gmail.com

**Resumo.** *Este documento contém noções de implementação e características dos programas de cliente e servidor baseados em UDP com transmissão confiável de dados. O documento também possui informações sobre o desempenho dos programas.*

## 1. Introdução

O User Datagram Protocol(UDP) é um dos principais integrantes do modelo conceitual de internet que conhecemos hoje. Possui como características principais:

- Utilização de indicadores de porta de origem e porta de destino no cabeçalho.
- Não possui handshaking nem controle de fluxo.
- Possui um cabeçalho de tamanho fixo de 8 bytes.

Já o Transmission Control Protocol(TCP), o qual também integra o modelo conceitual de internet, possui como características:

- Transmissão garantida de dados.
- Possui handshaking, controle de fluxo e controle de congestionamento.
- Possui algoritmo de janela deslizante para retransmissão de dados perdidos.

O intuito do trabalho é fazer com que o datagrama UDP seja transmitido de maneira confiável. Dessa forma, o arquivo será encapsulado em um segmento TCP, porém, transmitido em um datagrama UDP.

## 2. Informações importantes

### 2.1. Decisões de implementação

O trabalho foi dividido em dois arquivos, um para o servidor e outro para o cliente. No arquivo cliente, o cabeçalho é gerado antes de ser encapsulado junto ao segmento, uma thread fica responsável por escutar os ACK's recebidos do servidor. O arquivo servidor cria o registro do novo arquivo a ser recebido e salva os bytes nesse registro.

## **2.2. Requisitos para funcionamento dos programas**

O arquivo a ser transferido deve estar no mesmo diretório que o programa Client. O arquivo a ser recebido não deve existir antes de rodar o Server, o que pode ocasionar em loop.

### **2.2.1. Executar o programa Client.java com os parâmetros**

- fn: nome do arquivo a ser enviado
- sip: endereço IP do servidor
- sport: porta UDP do servidor
- wnd: tamanho da janela do transmissor e receptor em bytes

### **2.2.2. Executar o programa Server.java com os parâmetros**

- fn: nome do arquivo a ser recebido e gravado em disco
- sport: porta UDP que o servidor deve escutar
- wnd: tamanho da janela do transmissor e receptor em bytes

## **3. Funcionalidades não implementadas**

- Simulação da perda de pacotes
- Fazer um timeout de tempo variável
- Calcular o tamanho da janela de acordo com o tempo de envio do pacote

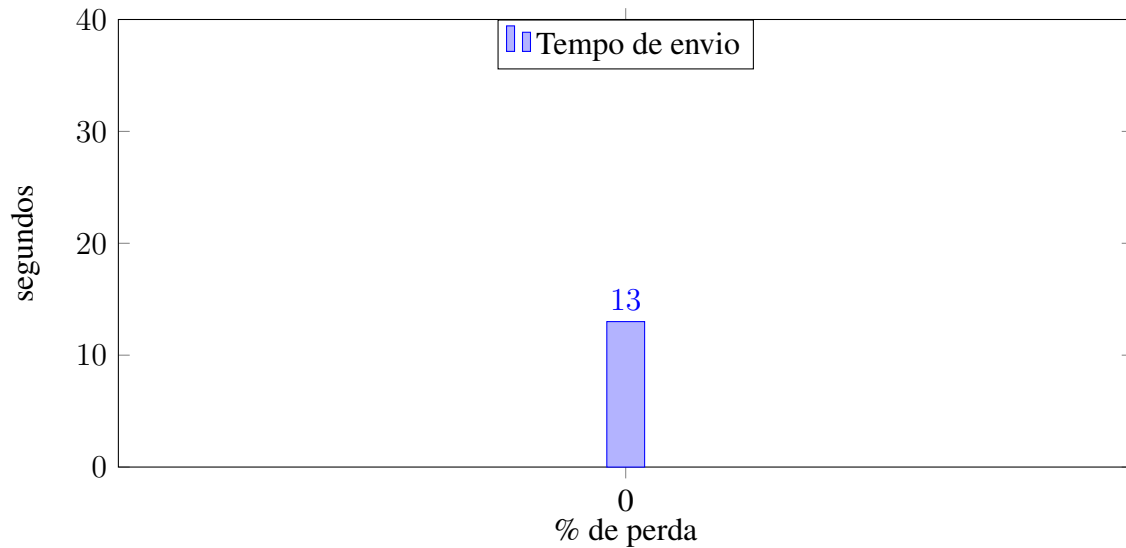
## **4. Problemas enfrentados na implementação**

- Dificuldade de lidar com os pacotes no envio de dados
- Dificuldade em debugar erros devido a quantidade de informações enviadas, muitos pacotes que são muito pequenos, difíceis de acompanhar
- Adaptação do protocolo UDP

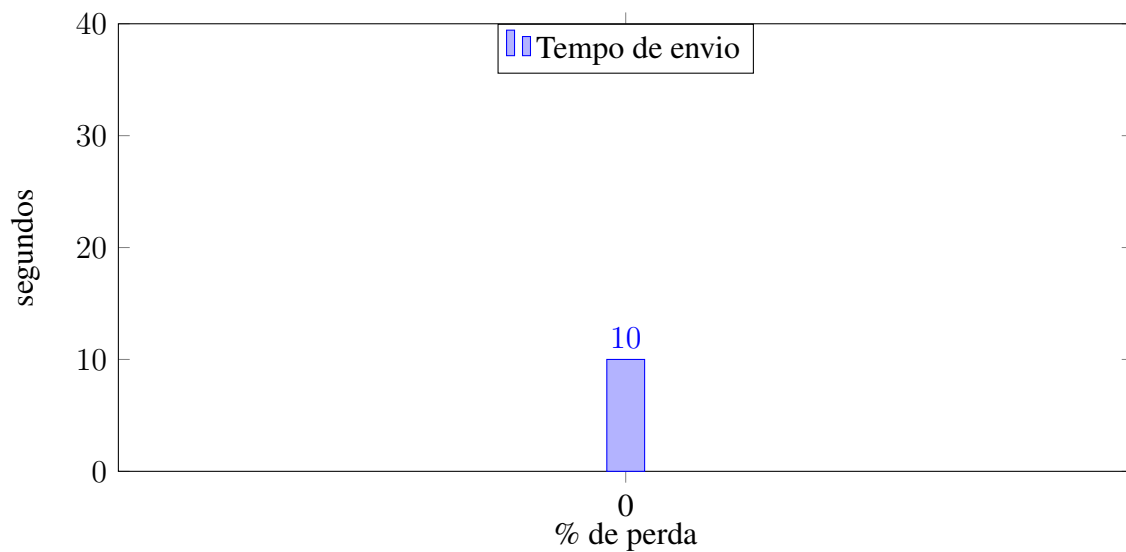
## 5. Avaliação experimental do protocolo

### 5.1. Experimentos

MSS 512 :



MSS 1024 :



### 5.2. Cenário usado

Computador 1 (Cliente) - Macbook late 2009

- Processador Intel Core 2 Duo de 2,26GHz com 3MB de cache L2 compartilhado no chip em par com a velocidade do processador
- SDRAM DDR3 de 2GB (duas SO-DIMMs de 1GB); dois slots SO-DIMM com suporte para até 4GB
- Rede sem fio Wi-Fi interna AirPort Extreme2 (de acordo com a especificação IEEE 802.11n); compatível com IEEE 802.11a/b/g
- Gigabit Ethernet 10/100/1000BASE-T (conector RJ-45)

Computador 2 (Servidor) - Dell Inspiron 14r5437

- Processador : Intel Core i5-4200U
- Memória RAM : 8gb DDR3 1600mhz
- Wireless LAN : 802.11 b/g/n
- Forma de comunicação : Wi-Fi
- Velocidade da rede : 10Mbps
- Rede utilizada : Rede local