



UNIVERSIDADE  
ESTADUAL de LONDRINA

---

JADER GOMES CARDOSO JUNIOR  
LAÍS TIE TAKAKI

**TRAB-03 - TRANSFERÊNCIA DE STRINGS VIA UDP**

LONDRINA - PR  
2023

---

## Trabalho de Laboratório – Relatório final 10/10/2023

Esta documentação detalha a aplicação desenvolvida para a transferência de strings entre dois computadores utilizando o protocolo UDP. A aplicação atende aos requisitos e faz uso das seguintes funções e módulos nos scripts *'sender.py'* e *'receiver.py'*.

### UTILIZAÇÃO

Requisitos: Python 3.x, fpdf.

Antes de executar os arquivos *sender.py* e *receiver.py*, é necessário configurar a randomização do hash, por meio da execução da instrução **export PYTHONHASHSEED=0** no terminal.

#### Sender (sender.py)

O Sender é responsável por enviar os dados para o destinatário. Ele implementa as seguintes funcionalidades:

- `send_message(message, checksum, dest)`: esta função envia o pacote de dados já devidamente parametrizados.
- `send(data, dest)`: esta função é responsável por enviar um pacote de dados para o destinatário. O processo envolve a divisão dos dados em pacotes parametrizados (tamanho 500B, 1000B ou 1500B) e o cálculo do header e checksum, parâmetros a serem enviados juntamente com o payload no pacote. Além disso, a função também gera um relatório detalhado da transmissão e encerra o socket após a conclusão.
  - Os tamanhos de cada parte do pacote são arbitrários; na implementação, foi utilizado um tamanho 2 para o header e 4 para o checksum, sendo este os 4 últimos dígitos de um número dado por uma função hash.

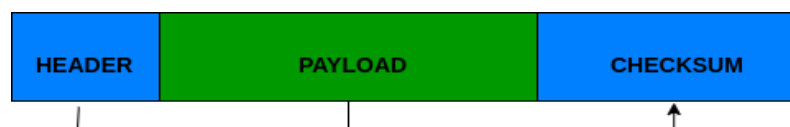


Figura 1: estrutura simplificada dos pacotes

- `send_report(data, dest)`: responsável por enviar ao usuário informações relevantes para a geração do relatório ao fim do programa. Esse processo envolve o estabelecimento de uma conexão TCP, o envio dos dados e o fechamento da conexão.

## Receiver (receiver.py)

O Receiver é responsável por receber e validar os pacotes do remetente. Ele implementa as seguintes funcionalidades:

- `receive_message()`: esta função é encarregada de receber os pacotes de dados enviados pelo remetente. Ela desempenha um papel importante na validação da integridade dos pacotes, calculando o checksum para validar os dados recebidos.
- `receive_report()`: esta função lida com a recepção e análise do relatório de transmissão enviado pelo remetente. Ela estabelece uma conexão TCP, extrai as informações do relatório e exibe os resultados em um PDF gerado automaticamente.

## Comunicação

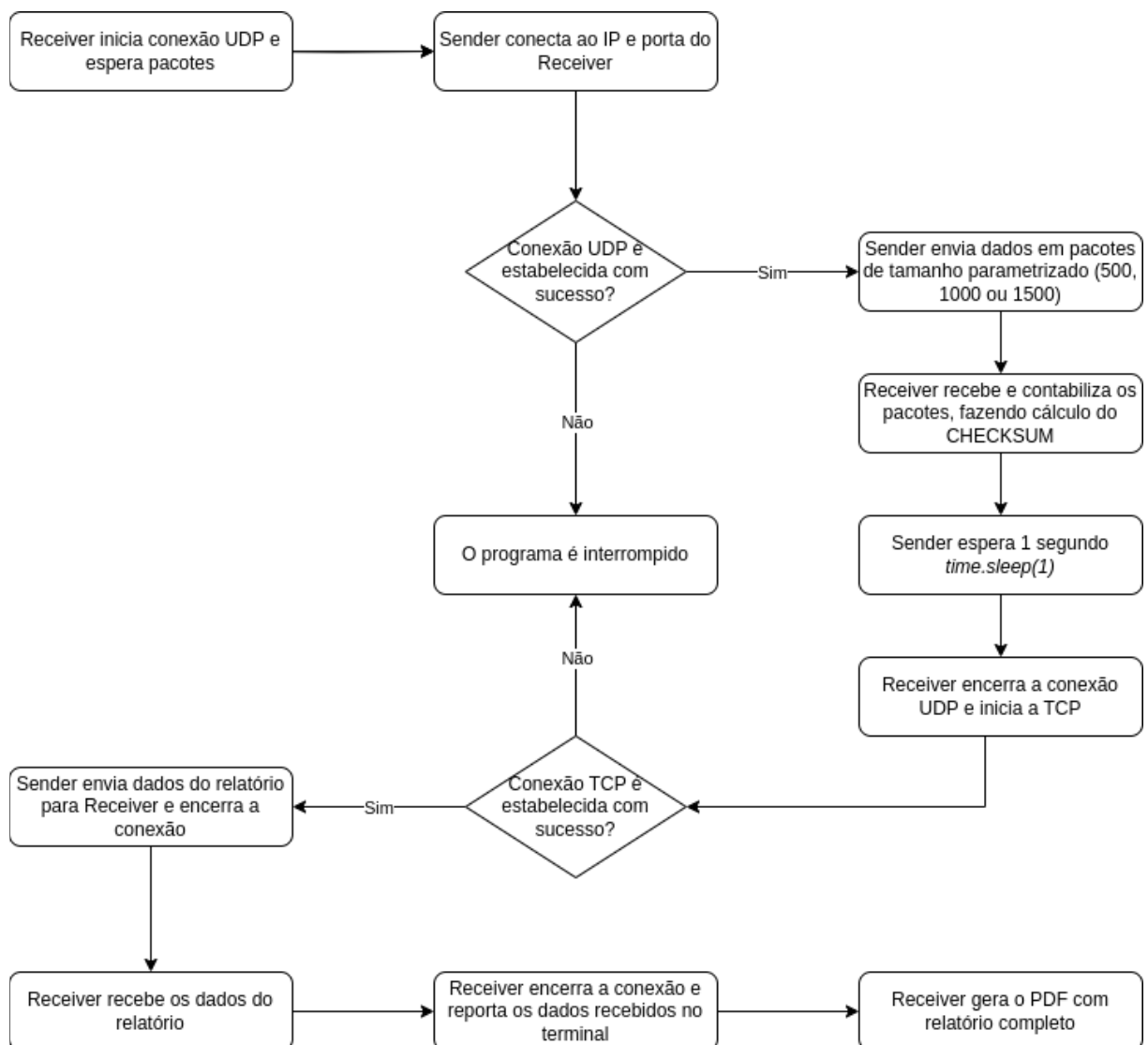


Figura 2: fluxograma operacional

A partir da figura acima, é possível observar de forma simplificada como a aplicação funciona, utilizando conexões UDP e TCP como meio de comunicação entre os computadores. Nesse sentido, a comunicação envolve a transmissão de dados do remetente para o destinatário via UDP, com verificação de integridade, seguida da geração e envio de um relatório detalhado via TCP. Assim, esse processo permite a transferência confiável de dados entre os computadores e o registro das estatísticas da transmissão, mesmo se tratando do protocolo UDP.

### **Observações**

Essa aplicação é uma demonstração prática e esclarecedora dos conceitos essenciais da comunicação em redes de computadores. Ela oferece uma visão abrangente das operações fundamentais, como o uso dos protocolos UDP e TCP para transferência de dados, a fragmentação da informação em pacotes, a aplicação de checksums para garantir a integridade dos dados e a avaliação do desempenho da rede.

Além disso, a aplicação não apenas aborda esses conceitos, mas também os coloca em prática de maneira coerente e interativa. Ela permite que os usuários observem como os dados são divididos, transmitidos e verificados, fornecendo uma compreensão prática da dinâmica por trás da comunicação em redes.

Assim, esta aplicação vai além do seu papel como ferramenta educacional, pois também se apresenta como uma base sólida para iniciar investigações em tópicos avançados relacionados à transferência de dados em redes de computadores.