

# TP1: Camada de Enlace

Fernanda Tempesta e Victor Caram

2 de maio de 2018

## 1 Introdução

O presente trabalho consiste na implementação de um emulador para a camada de enlace para uma rede fictícia denominada *DCCNET*.

A camada de enlace é uma das sete camadas componente do modelo OSI e possui como principais características a detecção de erro, e, eventualmente, sua correção. Entretanto, este trabalho não cobria a correção de erro, apenas a retransmissão do pacote caso fosse detectada alguma inconsistência nos dados.

Basicamente, o emulador desenvolvido trata da codificação, enquadramento, detecção de erros, sequenciamento e retransmissão dos dados.

## 2 Decisões de implementação

O trabalho foi implementado utilizando a linguagem de programação *Python3*, bem como uma conexão TCP. O emulador desenvolvido funciona como transmissor e receptor simultaneamente.

Para a inicialização do programa, são necessário alguns parâmetros com informações básicas para sua execução. A partir do segundo parâmetro, é possível identificar se o emulador deverá se comportar na ponta de forma passiva, ou seja, como o servidor, ao receber o parâmetro *-s*, ou de forma ativa, como cliente, ao receber o parâmetro *-c*. Esta distinção é feita objetivando identificar quais serão os próximos parâmetros a serem lidos para cada um dos casos, bem como a criação de suas conexões. Os quadros abaixo mostram o modelo de inicialização para cada um dos casos, servidor e cliente, respectivamente.

```
./dcc023c2 -s <PORT> <INPUT> <OUTPUT>
```

```
./dcc023c2 -c <IP>:<PORT> <INPUT> <OUTPUT>
```

Inicialmente, é preciso estabelecer uma conexão entre o transmissor e o receptor, assim como foi feito no primeiro trabalho da disciplina. Após se certificar de que esta conexão foi criada, o ambiente estará preparado para começar a transmissão dos dados.

### 2.1 Transmissão de dados

Para a transmissão de dados são necessários alguns passos:

1. Ler os dados a serem enviados, através do arquivo passado como parâmetro, e, a partir dele, montar o quadro a ser transmitido.
2. A função `enquadra(mensagem, id, flag)` faz o enquadramento dos dados seguindo as seguintes etapas:
  - (a) `sincFinal` - Cria o campo de sincronização (`dcc023c2`)
  - (b) `length` - Calcula o tamanho do dado
  - (c) `checksum` - Inicialmente o checksum é marcado como zero
  - (d) `idFinal` - Inicialmente o id de envio é setado para 0

- (e) `flagFinal` - Inicialmente a flag de envio é setada para 0
- (f) `dadosHEX` - E por fim os dados a serem enviados

O retorno desta função é a concatenação de cada uma das variáveis citadas acima gerando, assim, o enquadramento de envio.

Entretanto, antes de transmitir o quadro, ainda são necessários mais alguns ajustes de modo a garantir que o receptor possa identificar uma possível ocorrência de erros durante o envio dos dados. Desta maneira, é calculado o *checksum* do quadro gerado previamente. Para fins de esclarecimento, o *checksum* é um artifício inserido no quadro de envio com o intuito de verificar a integridade dos dados recebidos. Antes do envio de um pacote, é calculada a soma de verificação dos dados contidos no quadro e este valor é armazenado no campo *checksum* do mesmo. Quando este quadro chega ao receptor, a soma é realizada novamente para verificar a consistência dos dados recebidos.

Após o cálculo do *checksum* e a substituição deste valor no quadro gerado anteriormente, tem-se os dados enquadrados e prontos para serem transmitidos.

## 2.2 Recepção dos dados

A recepção foi implementada por meio dos seguintes passos:

1. Os dados são recebidos por meio da função `recv`
2. É calculado o *checksum* do quadro recebido para saber se o mesmo foi corrompido durante a transmissão
3. Dois quadros modelos foram criados, o `enquadramento0` (para ID = 0) e o `enquadramento1` (para ID = 1). Eles são compostos pela concatenação dupla dos campos de sincronização, pela variável `length`, pelo valor do checksum, pelo ID e pela flag, ou seja menos os dados. Estes modelos são comparados com os respectivos bytes no quadro recebido para verificar a consistência dos dados transmitidos.
4. Se essa comparação do quadro modelo com o recebido resultar em valores iguais, então os dados foram entregues com sucesso e não houve corrompimento durante a transmissão. Neste caso, um `ACK` é enviado para o remetente da mensagem confirmando o êxito da conexão. (O `ACK` é um quadro sem dados)
5. Após o envio do `ACK`, o quadro recebido anteriormente é escrito no arquivo de saída
6. Caso o quadro recebido seja diferente dos dois modelos, então um erro ocorreu durante o envio dos dados. Nenhuma confirmação é recebida e o receptor fica aguardando pela retransmissão do quadro.

## 2.3 Retransmissão de dados

Caso os dados tenham sido corrompidos durante o envio, o receptor não receberá um quadro de `ACK` e, portanto, será necessário a retransmissão do quadro.

## 2.4 Múltiplos clientes

O servidor pode se conectar a múltiplos cliente simultaneamente. Esta eventualidade foi tratada por meio do uso de *threads*.

O servidor fica esperando pela conexão com algum cliente. Quando isto acontece, uma nova *thread* é criada para tratar esta conexão, permitindo assim que o servidor continue disponível para estabelecer novas transmissões.

# 3 Considerações Finais

Este trabalho foi importante para complementar o conteúdo aprendido em sala de aula. Ao conhecer a teoria, já é possível ter uma boa noção do funcionamento das camadas de redes, entretanto, o trabalho prático é, sem dúvida, um mecanismo excelente para visualizar, de forma concreta, a maneira com que os processos acontecem, mais especificamente, na camada de enlace

A maior dificuldade encontrada diz respeito a compreensão da especificação, uma vez que, apesar de conter muitas informações, ela deixava alguns temas em aberto, o que gerou a possibilidade de múltiplas interpretações. Além disso, alguns outros tópicos também incitaram muitas dúvidas, como por exemplo a função checksum e os arquivos disponibilizados para teste. Foi necessário certa dedicação e tempo para compreensão e correção dos erros gerados decorrentes das questões citadas acima. Contudo, após o esclarecimento das dúvidas foi possível concluir satisfatoriamente o trabalho.