

2023_2 - REDES DE COMPUTADORES - TE

[PAINEL](#) > [MINHAS TURMAS](#) > [2023_2 - REDES DE COMPUTADORES - TE](#) > [TAREFAS E MATERIAL DE ESTUDO](#)

> [TRABALHO PRÁTICO 1: PECRC, UM PROTOCOLO DE ENLACE CONFIÁVEL](#)

Trabalho prático 1: PECRC, um protocolo de enlace confiável

Trabalho em dupla ou individual

Prazo de entrega: confira o moodle, no link de entrega

Introdução

Neste trabalho vamos implementar o PECRC, um protocolo simples de enlace confiável por retransmissão, com enquadramento por byte-stuffing.

Objetivo

Seu objetivo será criar um módulo em Python que implementará um protocolo de enlace com enquadramento por sentinelas de bytes e com retransmissão de mensagens não confirmadas.

O princípio de operação

O seu protocolo deve ser definido dentro do módulo PECRC e deve implementar as seguintes funcionalidades:

- enquadramento por byte stuffing
- checksum de 16 bits
- mensagem de confirmação (ACK) enviado para cada quadro recebido corretamente
- retransmissão por temporização de 5 segundos

Especificação do protocolo

O protocolo utiliza um quadro com os seguintes campos:

- um marcador de início de quadro que será o caractere '[' (abre-cholchete);
- um caractere de controle que indica se o quadro contém dados ('D') ou confirmação ('C');
- um caractere para numeração dos pacotes, poderá ser '0' ou '1' (note que não um **byte** com valor 0 ou 1);
- um campo de dados de tamanho variável, com no máximo 1500 bytes;
- um campo de checksum com dois bytes (algoritmo de checksum de 16 bits da internet);
- um marcador de fim de quadro, que será o caractere ']' (fecha-colchete).

O processo de byte stuffing é feito usando-se o ponto de exclamação (!) como caractere de escape; apenas os caracteres de início de bloco, fim de bloco e o próprio ponto de exclamação, caso ocorram entre os marcadores de início e fim do bloco, precisam ser precedidos do caractere de escape. Qualquer byte encontrado depois de um byte de escape deve ser simplesmente tratados como um byte normal, sem significado especial.

Um quadro de confirmação terá todos os campos mencionados, exceto o campo de payload. Um ACK deve sempre carregar no campo de numeração o número do quadro que está sendo confirmado.

Detalhamento da implementação

Junto com este enunciado são publicados quatro arquivos, entregues em um arquivo .zip:

- canal_tp1.py - uma camada que encapsula a interface de sockets (pense nele como uma camada física);
- pecrc.py - uma versão mínima do módulo para a implementação do PECRC, que implementa um protocolo nulo, que não altera os dados que passam por ele;
- envia.py - um programa que lê um arquivo e usa o PECRC para enviá-lo através do enlace;
- recebe.py - um programa que usa o PECRC para receber um arquivo do enlace e salvá-lo no disco.

Apenas o arquivo pecrc.py deve ser alterado e será o único arquivo de código esperado na entrega. Os demais arquivos não podem ser alterados. Durante a avaliação, uma versão mais elaborada do arquivo canal_tp1.py será utilizada. Essa versão poderá fazer a injeção de erros no canal de transmissão (bytes corrompidos, removidos ou inseridos, quadros perdidos ou duplicados). Essa versão não será publicada.

Sobre a execução dos programas:

Os programas `envia.py` e `recebe.py` não devem ser alterados e devem ser sempre executados com os parâmetros de linha de comando como indicados. Devido ao uso de TCP na criação dos links, o programa `recebe.py` deve sempre ser executado primeiro, pois ele executa a abertura passiva do canal.

Sua implementação deve se feita alterando o arquivo `pecrc.py`. Para simplificar a entrega e correção automática, todo o código do protocolo deve estar contido naquele arquivo, que será o único arquivo `.py` esperado na entrega.

A única saída esperada a criação do arquivo recebido por `recebe.py`, com o nome indicado como parâmetro. Caso você inclua mensagens de depuração ou outras informações que sejam exibidas durante a execução, certifique-se de removê-las da saída na execução da versão final.

O código deve usar apenas Python 3, sem bibliotecas além das consideradas padrão. O material desenvolvido por você deve executar sem erros em uma máquina linux usual. A correção será feita em uma máquina com Ubuntu 20.04 LTS e Python 3.8.10.

A correção será feita automaticamente e e programas que não executarem, não seguirem as determinações quanto ao formato da entrada e da saída, ou apresentarem erros durante a execução serão penalizados/desconsiderados (dependendo da gravidade do problema encontrado).

O que deve ser entregue:

Você deve entregar **apenas** o arquivo `pecrc.py`, que deverá ter todo o código fonte do protocolo PECRC. A primeira linha deve conter, como comentário, os nomes da dupla de autores. Não será exigido um relatório separado mas espera-se que o código fonte seja devidamente comentado para destacar as decisões de implementação.

Preste atenção nos prazos: entregas com atraso poderão ser aceitas em casos especiais (converse antes com o professor), mas serão penalizadas.

Sugestões de depuração

Faz parte do exercício desenvolver os testes para o mesmo. Não serão fornecidos arquivos de teste nem detalhes sobre a versão de avaliação do `canal_tp1.py`. Vocês devem criar os seus próprios arquivos e testes e podem compartilhá-los no fórum do exercício. Por outro lado, obviamente não é permitido discutir o princípio de sincronização da solução. Certifique-se que seu protocolo funciona ao enviar arquivos que contenham os caracteres de demarcação ('[', ']', '!') em qualquer lugar.

Dúvidas?

Use o fórum criado especialmente para esse exercício de programação para enviar suas dúvidas. **Não é permitido publicar código no fórum!** Se você tem uma dúvida que envolve explicitamente um trecho de código, envie mensagem por e-mail diretamente para o professor.

◀ [Links úteis: serviços de apoio ao estudante](#)

Seguir para...

[Código fonte para o TP1](#) ▶