

Trabalho Final de Redes de Computadores e de Laboratório de Redes

Nomes: Arthur Sudbrack Ibarra, Miguel Torres de Castro e Felipe Grosze Nipper de Oliveira.

Estrutura do código

O código fonte do trabalho possui 6 arquivos Python, cada um sendo responsável por alguma parte do sistema. Dentre os arquivos mais importantes, podemos citar:

- **configurations.py:** responsável por ler e interpretar os arquivos de configuração.
- **custom_sockets.py:** responsável pela implementação dos sockets UDP de cliente e de servidor.
- **packaging.py:** responsável pela abstração dos dois tipos de pacote: token e dados. Neste módulo, também existem métodos para identificar pacotes, calcular CRC's e inserir propositalmente erros em pacotes de dados.
- **thread_managers.py:** este é o módulo mais importante e extenso do sistema. Responsável por gerenciar a thread de input de usuário e a thread de recebimento/envio de pacotes por sockets. Este módulo faz uso de um socket cliente para enviar dados e de um socket servidor para receber dados. Além disso, é aqui que acontece todo o controle de token e a checagem dos pacotes.

Estrutura de dados

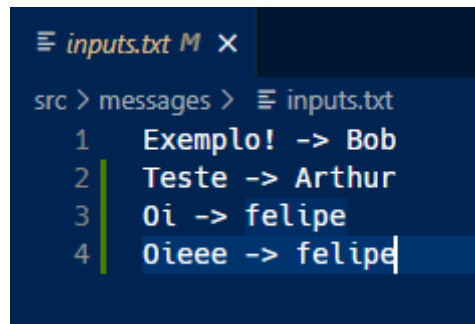
Uma estrutura de dados de fila foi utilizada para armazenar as mensagens que deveriam ser enviadas. A fila utilizada permite a inserção de elementos tanto à direita quanto à esquerda, isto porque quando um usuário informar novas mensagens para colocar na fila, elas deverão ser inseridas à direita, mas quando um NAK é retornado para a máquina origem, a mensagem deve ser inserida de novo no início da fila, ou seja, à esquerda.

Threads

Conforme dito anteriormente, dentro do arquivo **thread managers.py**, 2 threads concorrentes foram utilizadas: uma para receber o input do usuário e outra para gerenciar os sockets cliente e servidor.

- **Thread de input do usuário:** Esta thread olha constantemente para um arquivo chamado 'inputs.txt', onde são escritas as mensagens que queremos

mandar para a fila. Quando uma mudança é detectada no arquivo, a última linha do arquivo é lida e a mensagem para o destino especificado é colocada na fila.



```
inputs.txt M X
src > messages > inputs.txt
1 Exemplo! -> Bob
2 Teste -> Arthur
3 Oi -> felipe
4 Oieeee -> felipe
```

- **Thread de sockets:** Esta thread gerencia os sockets cliente e servidor. É dentro desta thread que recebemos e enviamos pacotes tanto de token quanto de dados. Caso a máquina em que o programa está sendo executado é aquela responsável pelo controle do token, então dentro desta thread também será feito esse controle, com o descarte ou geração de tokens.

Quanto à **sincronização das threads**, não se fez necessário o uso de semáforos, visto que a fila utilizada em Python é thread-safe.

Cálculo do CRC

Para o cálculo do CRC, foi utilizada uma biblioteca nativa de Python chamada 'zlib'. A biblioteca em questão possui uma função chamada 'crc32' que espera uma mensagem em bytes como parâmetro e retorna o CRC calculado em inteiro.

Como executar o programa

Para executar o programa, é preciso ter o Python 3 instalado em seu computador. Com esse requisito cumprido, abra um terminal, vá para o diretório 'src' e digite o comando Python:

```
cd src/
python main.py config/config-4.txt 9000
```

O primeiro parâmetro é o caminho relativo até o arquivo de configuração, e o segundo parâmetro é a porta em que o seu computador estará ouvindo mensagens de sockets clientes.

Exemplos de execução

Abaixo seguem imagens demonstrando a execução do programa. Nessas imagens, é possível visualizar exemplos de ACK, NAK, broadcast, timeout, remoção do token da rede...

Os testes foram realizados em 1 computador, porém utilizando 3 portas diferentes para simular a existência de 3 computadores se comunicando na rede.

```
[Arquivo de configuração]
IP da máquina à direita: 192.168.1.104
Porta da máquina à direita: 9100
Apelido da máquina atual: arthur
Tempo do token: 1
Deve gerar token: True

Para enviar pacotes, escreva no arquivo messages/inputs.txt [sua mensagem] -> [apelido do destino]. Após isso, SALVE O ARQUIVO.
Exemplo: Oi -> Maria

=====

Pressione enter para continuar...

Enviando token para a máquina à direita.

Mensagem 'ola pessoal' para 'TODOS' colocada na fila!
Mensagem 'ola felipe' para 'felipe' colocada na fila!
Mensagem 'ola miguel' para 'miguel' colocada na fila!

Recebi Token: 1111

Enviando próxima mensagem da fila: '2222;maquinanaoexiste:arthur:TODOS:3655262330:ola pessoal'

-----
Recebi Dados: 2222;maquinanaoexiste:arthur:TODOS:3655262330:ola pessoal
Recebi [maquinanaoexiste] para a mensagem 'ola pessoal' - retorno de meu BROADCAST.
Enviando token [1111] para a máquina à direita com IP: 192.168.1.104
-----

Recebi Token: 1111

Enviando próxima mensagem da fila: '2222;maquinanaoexiste:arthur:felipe:-9999:ola felipe'

-----
Recebi Dados: 2222;NAK:arthur:felipe:-9999:ola felipe
Recebi [NAK] para a mensagem 'ola felipe' - colocando o pacote no início da fila para tentar novamente.
Enviando token [1111] para a máquina à direita com IP: 192.168.1.104
-----

Recebi Dados: 2222;NAK:arthur:felipe:-9999:ola felipe
Mesmo após o reenvio do pacote, um [NAK] foi recebido novamente. O pacote não será adicionado na fila novamente.
Recebi Token: 1111

Enviando próxima mensagem da fila: '2222;maquinanaoexiste:arthur:miguel:-9999:ola miguel'

-----
Recebi Dados: 2222;NAK:arthur:miguel:-9999:ola miguel
Recebi [NAK] para a mensagem 'ola miguel' - colocando o pacote no início da fila para tentar novamente.
Enviando token [1111] para a máquina à direita com IP: 192.168.1.104
-----

Recebi Token: 1111

Enviando próxima mensagem da fila: '2222;NAK:arthur:miguel:-9999:ola miguel'

-----
Recebi Dados: 2222;NAK:arthur:miguel:-9999:ola miguel
Mesmo após o reenvio do pacote, um [NAK] foi recebido novamente. O pacote não será adicionado na fila novamente.
Enviando token [1111] para a máquina à direita com IP: 192.168.1.104
-----
```

Recebi Token: 1111

Enviando token para a máquina à direita.

[Timeout] atingido, um novo token será gerado.

Enviando token para a máquina à direita.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;maquinanaoexiste:felipe:arthur:111014304:olaaaaaaaa, arthur
CRC recebido: 111014304
CRC calculado: 111014304

Retornando [ACK] - Origem: felipe, Mensagem: olaaaaaaaa, arthur

Recebi Token: 1111

Enviando token para a máquina à direita.

```
[Arquivo de configuração]

IP da máquina à direita: 192.168.1.104
Porta da máquina à direita: 9200
Apelido da máquina atual: miguel
Tempo do token: 1
Deve gerar token: False

Para enviar pacotes, escreva no arquivo messages/inputs.txt [sua mensagem] -> [apelido do destino]. Após isso, SALVE O ARQUIVO.
Exemplo: Oi -> Maria

=====

Pressione enter para continuar...|
```

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;maquinanaoexiste:arthur:TODOS:3655262330:ola pessoal
Retornando [maquinanaoexiste], a mensagem recebida foi enviada em broadcast.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;maquinanaoexiste:arthur:felipe:-9999:ola felipe

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;NAK:arthur:felipe:-9999:ola felipe

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;maquinanaoexiste:arthur:miguel:-9999:ola miguel
CRC recebido: -9999
CRC calculado: 583008938

Retornando [NAK] - Origem: arthur, o CRC não bate.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;NAK:arthur:miguel:-9999:ola miguel
CRC recebido: -9999
CRC calculado: 583008938

Retornando [NAK] - Origem: arthur, o CRC não bate.

Recebi Token: 1111

Removendo token da rede...

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;ACK:felipe:arthur:111014304:olaaaaaaaa, arthur

Recebi Token: 1111

Enviando token para a máquina à direita.

[Arquivo de configuração]

IP da máquina à direita: 192.168.1.104

Porta da máquina à direita: 9000

Apelido da máquina atual: felipe

Tempo do token: 1

Deve gerar token: False

Para enviar pacotes, escreva no arquivo `messages/inputs.txt` [sua mensagem] -> [apelido do destino]. Após isso, SALVE O ARQUIVO.
Exemplo: Oi -> Maria

Pressione enter para continuar...

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;maquinanaoexiste:arthur:TODO\$3655262330:ola pessoal
Retornando [maquinanaoexiste], a mensagem recebida foi enviada em broadcast.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;maquinanaoexiste:arthur:felipe:-9999:ola felipe
CRC recebido: -9999
CRC calculado: 27499725

Retornando [NAK] - Origem: arthur, o CRC não bate.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;NAK:arthur:felipe:-9999:ola felipe
CRC recebido: -9999
CRC calculado: 27499725

Retornando [NAK] - Origem: arthur, o CRC não bate.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;NAK:arthur:miguel:-9999:ola miguel

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Dados: 2222;NAK:arthur:miguel:-9999:ola miguel

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Token: 1111

Mensagem 'olaaaaaaaa, arthur' para 'arthur' colocada na fila!

Enviando próxima mensagem da fila: '2222;maquinanaoexiste:felipe:arthur:111014304:olaaaaaaaa, arthur'

Recebi Dados: 2222;ACK:felipe:arthur:111014304:olaaaaaaaa, arthur

Recebi [ACK] para a mensagem 'olaaaaaaaa, arthur' - o recebimento do pacote foi confirmado.

Enviando token [1111] para a máquina à direita com IP: 192.168.1.104

Recebi Token: 1111

Enviando token para a máquina à direita.

Recebi Token: 1111