

## Definição do Trabalho

A implementação do trabalho consiste em simular na rede local o funcionamento de uma rede em anel. O programa deverá implementar a transmissão de mensagens entre as máquinas que compõem o anel, com o protocolo UDP como transporte. Deve ser implementada uma fila de mensagens e somente uma mensagem pode ser transmitida por token (pacote especial que circula na rede e permite a transmissão de mensagens) recebido.

O programa deverá possuir dois tipos de mensagens: o token e os dados. Ao iniciar o programa, o usuário deverá informar o endereço IP da máquina que está a sua direita, um apelido e o tempo do token e dos dados. Ao enviar uma mensagem, o usuário irá identificar o destinatário através de um apelido. Deverá ser possível especificar o tempo que a estação irá permanecer com as mensagens (para fins de depuração) em segundos. Tais informações devem ser inseridas na aplicação através de um arquivo de configuração. O arquivo de configuração deverá seguir o seguinte formato:

```
<ip_destino_token>:<porta>  
<apelido>  
<tempo_token>
```

Exemplo:

```
192.168.1.12:5000  
Bob  
1
```

Apenas uma determinada máquina será responsável por gerar o token a primeira vez. Esta máquina deve enviá-lo para a máquina que está a sua direita no anel. Caso a máquina que recebeu o token não tenha dados para transmitir (fila de mensagens vazia), o token será enviado para a próxima máquina do anel (máquina a direita). Caso contrário, a primeira mensagem é retirada da fila de mensagens e é enviada para a máquina a sua direita, ou seja, os dados também devem seguir a ordem do anel.

Os dados enviados deverão retornar à máquina origem e somente depois disso o token poderá ser enviado para a próxima estação.

Quando a máquina origem enviar um pacote de dados, um campo no cabeçalho do pacote deverá ser marcado como “nãocopiado”. Então, se esse pacote retornar para a máquina origem com o campo em “nãocopiado”, significa que a máquina destino não se encontra na rede ou está desligada. Caso o pacote seja retornado com o campo em “erro”, significa que a máquina destino identificou um erro no pacote e o mesmo deverá ser retransmitido pela origem. Caso o pacote retorne com o campo em “OK”, significa que o pacote foi recebido corretamente pela máquina destino.

Desta forma, deve haver um módulo de inserção de falhas que force as estações a marcar os pacotes com “erro” pela máquina destino. Este módulo deve trabalhar com alguma probabilidade e informar que a mensagem recebida chegou com “erro” ou “OK”.

Além disso, deverá ser implementada uma fila de mensagens para cada máquina. Esta fila poderá estar vazia ou não. A fila poderá conter até 10 mensagens e essas mensagens deverão ser configuradas pelo usuário, assim como o endereço da máquina destino.

Os serviços de envio de dados oferecidos devem contemplar duas formas de transmissão:

- Unicast: envia o pacote para um único destino;
- Broadcast: envia o pacote para todas as máquinas da rede usando o apelido TODOS. Neste caso, o módulo de inserção de falhas deve manter a mensagem em “nãocopiado”.

## 1. Descrição das Mensagens

A implementação deve seguir fielmente o formato das mensagens descrito a seguir. Isso é necessário para garantir a interoperabilidade das implementações. Ou seja, implementações de diferentes grupos devem ser capazes de participar da mesma rede em anel. A correta interação entre as diferentes implementações fazem parte da avaliação do trabalho.

### 1.1 Token

O token será formado por uma sequência numérica em formato string e terá o valor 1234, como mostra o exemplo a seguir:

1234
------

### 1.2 Mensagem de dados

Uma mensagem de dados é formada por outra sequência numérica em formato string e terá o valor 2345. Nesta caso, o valor 2345 será seguido de um ‘;’ e da mensagem que será composta pelos campos: <controle de erro>:<apelido de origem>:<apelido do destino>:<mensagem>. Exemplo:

4066;naocopiado:Bob:Alice:Oi Mundo!
-------------------------------------

Ao receber uma mensagem de dados, a estação verifica se a mensagem é endereçada a ela (verificando o apelido do destino). Caso não seja, esta mensagem deve ser enviada para seu vizinho da direita. Caso a mensagem seja para ela, a aplicação deve imprimir o apelido origem e a mensagem, e deve também enviar a mensagem de volta, alterando o campo naocopiado para OK ou erro.

Caso a mensagem de dados seja recebida por quem a originou (o apelido de origem é igual ao seu), será necessário verificar o controle de erro. Ao receber uma mensagem com este campo em naocopiado ou OK, um token deve ser enviado para seu vizinho da direita. Caso a mensagem venha com erro, a mensagem deve ser retransmitida apenas uma vez na rede, trocando o erro por naocopiado, e enviando a mensagem para a máquina a sua direita.

## 2. Visualização e depuração

A demonstração deverá acontecer, no mínimo, em 3 máquinas.

Deve ser possível:

- Especificar, a qualquer momento, a mensagem a ser enviada por uma máquina;
- Visualizar onde o token e o pacote de dados se encontram durante a execução do programa;
- Avisar quando houver retransmissões;
- Saber o que está acontecendo.

### Regras Gerais

---

Grupos: Até 3 componentes.

Data de entrega e apresentação: **30/10**

Obs.: Todos participantes devem estar presentes

Entrega final:

- Texto descrevendo a estrutura da solução dada, envolvendo estruturas de dados, threads, classes, mecanismos de sincronização utilizados, etc.
- Código fonte comentado.