

Definição do Trabalho

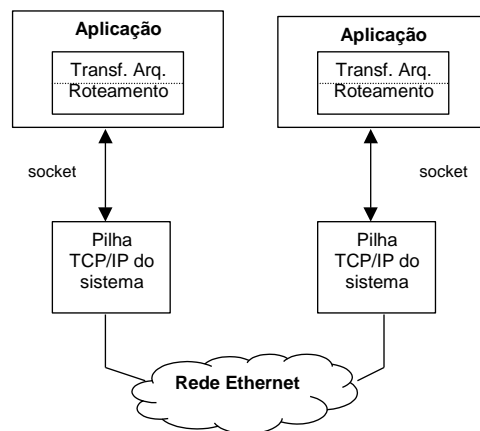
Objetivo

O trabalho prático da disciplina tem como objetivo exercitar os seguintes conceitos:

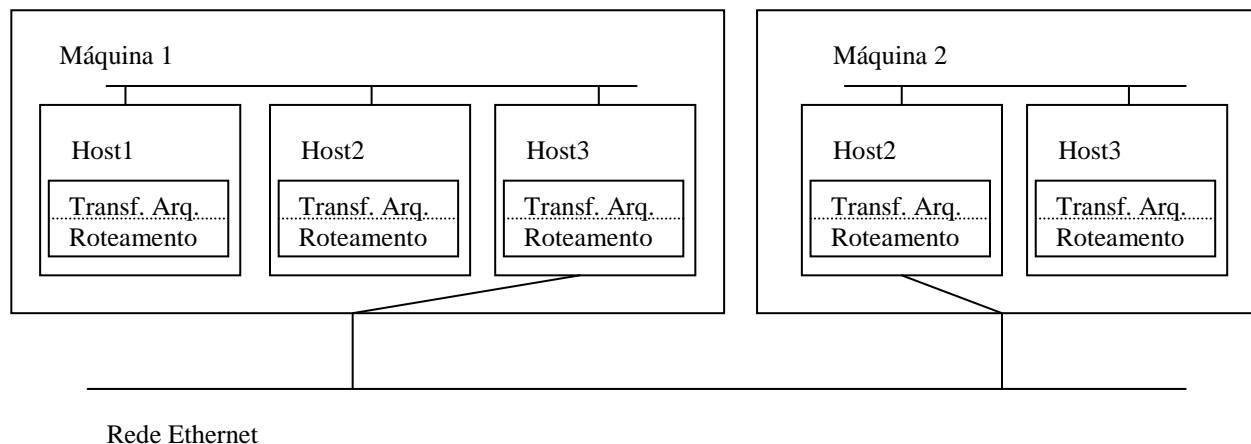
- comunicação entre níveis;
- controle de erro;
- endereçamento;
- troca de mensagens;
- roteamento.

Descrição

Implementar um programa que faça a transferência de um arquivo através de uma rede simulada. Para a transferência funcionar, será preciso implementar um protocolo para transferir arquivos e um protocolo que fará o roteamento das mensagens entre os hosts. Os dois rodarão sobre UDP, portanto será preciso verificar se os pacotes estão chegando corretamente. Neste caso, uma técnica de controle de erro deve ser implementada.



Uma máquina poderá ter vários processos de transferência de arquivos, onde cada processo será considerado uma máquina diferente na rede simulada. Quando um arquivo precisar ser entregue para uma outra rede, ele precisará ser enviado para o processo que faz o papel de roteador e este deverá enviar o pacote para uma outra máquina.



As tabelas de roteamento poderão ser configuradas dinamicamente e deve-se poder visualizá-las. Todos os *hosts* da rede devem possuir uma tabela de roteamento.

O endereço das máquinas da rede simulada deve corresponder a um par <IP,Porta>. Nesse par <IP, Porta>, **IP** é o endereço verdadeiro da máquina onde o módulo que implementa um *host* está instalado e **Porta** é o endereço pela qual são enviados/recebidos os dados, ou seja, a porta que o *socket* real utiliza para ler e escrever da rede.

Quando um arquivo precisar ser enviado, o controle de erro deve ser calculado e incluído na mensagem. Além disso, deve-se verificar o IP de destino da mensagem. Caso este seja o mesmo da máquina que vai enviar a mensagem, deve-se verificar a porta de destino, para que a entrega seja feita para o processo correto. Caso o endereço de destino seja diferente, a mensagem deve ser enviada para o host que faz o papel de roteador. Somente um host na rede simulada deve fazer o papel de roteador.

Os pacotes que circulam na rede devem possuir um cabeçalho com as informações necessárias na transferência. Essas informações são: endereço IP origem, endereço IP destino, porta origem, porta destino e controle de erro.

Todo o processo de recepção e roteamento das mensagens deve estar impresso na tela. Dessa forma será possível visualizar todo o caminho por onde a mensagem passou. Além disso, quando o destino receber uma mensagem, o controle de erro dessa mensagem deve ser feito e o arquivo recebido deve ser salvo na máquina.

Não existirá um número fixo de *hosts*, eles poderão ser incluídos a qualquer momento na rede.

A rede que suportará a simulação é uma rede de alta velocidade e com taxa de erros muito baixa, de forma que os pacotes gerados pela aplicação não serão espontaneamente perdidos. Assim sendo, deve haver um módulo de inserção de falhas que force as estações a enviar uma mensagem com “erro”. Essa função deve inserir falhas aleatoriamente durante a transmissão e o protótipo deverá operar em dois modos distintos: **com** e **sem falhas**.

Regras Gerais

Grupos: Individualmente ou grupos de no máximo 3 alunos.

Entrega e Apresentações: 17 de junho

Entrega final:

- Texto descrevendo a estrutura da solução dada, envolvendo estruturas de dados, threads, procedimentos (ou classes), mecanismos de sincronização utilizados
- Código fonte comentado

Apresentação:

- **todos** participantes devem estar presentes

Visualização dos Resultados:

- demonstração deverá acontecer em mais de uma máquina (no mínimo 2);

- um módulo de testes deve permitir a edição de um pacote a ser enviado a algum destino (ip, porta, arquivo a ser enviado);
- na chegada de um pacote ao destino, seu conteúdo deve ser salvo;
- deve ser possível visualizar o roteamento realizado;
- devem ser estabelecidos procedimentos para contabilização de estatísticas como: número de pacotes de dados enviados, recebidos, perdidos e retransmitidos para cada máquina;
- deve haver uma indicação quando a mensagem for recebida com erro.