Projeto de Desenvolvimento com *Sockets*

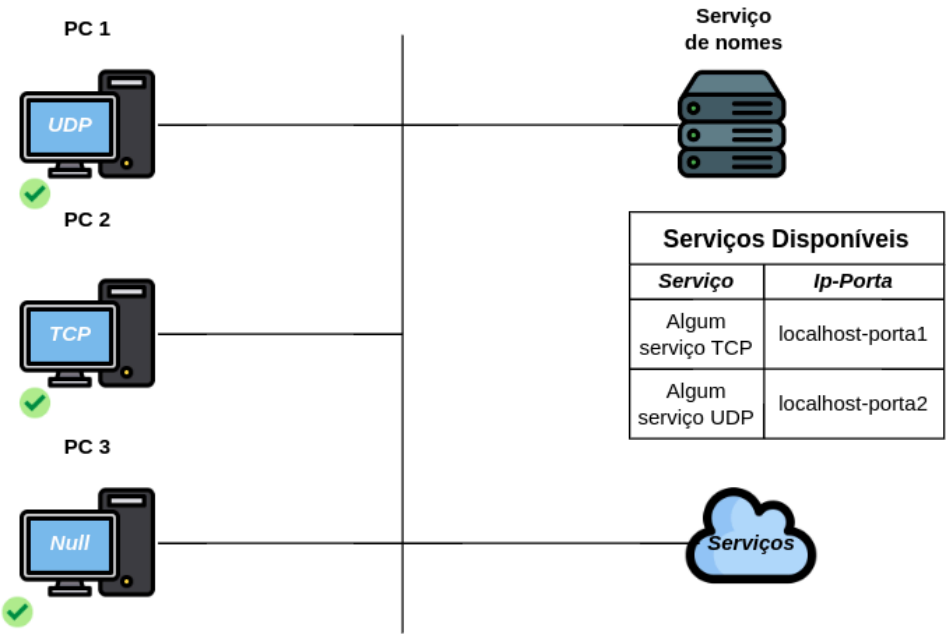

Figura 1: Visão geral do projeto.

Instruções

1) Criar duas aplicações cliente-servidor (1 TCP e 1 UDP);
    a) Criar a mesma aplicação utilizando os protocolos diferentes;
    b) O serviço final é livre;
    c) Medir o tempo total e o tempo de envio;

2) Crie seu próprio Servidor de Nomes: As aplicações quando inicializadas devem ser registradas num serviço de nomes para ficar disponível para os clientes;
    a) Quando o cliente solicitar um serviço, deve ser retornado o conjunto (ip, porta);
    b) Caso o serviço não exista deve ser retornado uma mensagem “Not Found”.

3) Quando um cliente quiser um serviço, ele deve enviar uma solicitação para o serviço de nomes e verificar a disponibilidade, o experimento deve conter 3 clientes;
    a) Um dos clientes deve solicitar o serviço UDP;
    b) Um dos clientes deve solicitar o serviço TCP;
    c) Um dos clientes deve solicitar um serviço inexistente;

Orientações

1) O serviço final é livre, mas a arquitetura ***cliente-servidor*** é obrigatória e a solução deve permitir a mudança de protocolo na camada de transporte entre TCP e UDP;
    a) Como o serviço a ser proposto é livre, recomenda-se elaborar uma ***aplicação cliente-servidor simples,*** mas bem estruturado, como:
        i) Calculadora Remota;
        ii) Sistema de busca (restaurantes, pontos turísticos, jogos digitais)
        iii) Cálculo de notas (Cadastro de alunos, retorno de notas, armazenamento de condição {Reprovado, Aprovado});
        iv) Sistema de tarefas a serem realizadas;
        v) Entre outros.

2) Utilize o primeiro protocolo TCP ou UDP para construir toda a aplicação. Uma vez que funcione com o primeiro protocolo, adapte a sua aplicação para funcionar corretamente com o segundo protocolo.

3) Ambas as aplicações (TCP e UDP) devem estar funcionando simultaneamente, ou seja, processos independentes ao operando ao mesmo tempo.

    a) Utilizar uma versão simplificada, criando dois códigos separados onde cada um deve ser executado em um terminal;
    b) Sendo assim, ambas ao serem inicializadas devem ser registradas no servidor de nomes e disponíveis para os clientes;
    c) Uma vez que ambas as aplicações (TCP e UDP) estiverem disponíveis, os clientes podem iniciar a solicitação de requisições;
    d) Quando estas forem encerradas, devem ser removidas do servidor de nome;

4) Tente criar um diagrama de Fluxo ou UML para melhor visualização do funcionamento total do projeto.

5) Obtenha todos os pacotes (entre o servidor de nomes e os servidores, entre o cliente e o servidor, entre o servidor de nomes e o cliente) em uma única captura Wireshark.

6) Note que, a captura do tempo de envio e recebimento está atrelada à análise de desempenho da camada de transporte, então só deve ser capturado o tempo dos pacotes da aplicação cliente-servidor;
    a) Quando o cliente enviar o pacote, deve-se iniciar a contagem do tempo. Quando o cliente receber a resposta final, o tempo será armazenado.
    b) Para evitar interferências nos resultados, sua aplicação não pode receber valores via input da linha de comando. Portanto, deixe as solicitações pré-determinadas via código (utilização de listas, mapas ou variáveis);

Entregas

1) Deve ser elaborado um relatório explicando o serviço desenvolvido, a captura do Wireshark, e uma análise comparativa entre os tempos capturados utilizando cada protocolo da camada de transporte (UDP e TCP);
2) Os códigos devem ser armazenados ser enviados junto com o relatório no formato .zip.
    a) Criar um README contendo as informações de como executar os códigos;

3) **O Relatório, os códigos e a captura do Wireshark devem ser anexados nesta atividade do Classroom.**