

Trabalho prático

Nomes: Felipe Campolina, Leandro Guido, Marcelo Augusto e Pedro Soares

[Vídeo relatório](#)

Objetivo

O objetivo principal deste projeto é implementar uma aplicação de um sistema de comunicação entre um servidor e clientes utilizando os protocolos TCP e UDP. O sistema foi desenvolvido com o intuito de ilustrar a funcionalidade básica de comunicação entre servidores e clientes, utilizando multithreading para garantir a eficiência e a robustez da aplicação e pode ser utilizado para diversas finalidades, como transmissão de dados em redes locais ou através da Internet.

Estrutura da Aplicação

A aplicação consiste em dois componentes principais:

- Servidor: Implementa servidores TCP e UDP para aceitar conexões de clientes, enviar e receber mensagens.
- Cliente: Implementa clientes TCP e UDP que se conectam aos servidores, enviam comandos e recebem respostas.

Servidor

Descrição Geral

- O servidor é capaz de lidar com conexões TCP e UDP simultaneamente. Ele escuta em portas específicas e responde aos comandos recebidos dos clientes.

Funcionalidade do Servidor TCP:

- Porta: 9999
- Operação: O servidor aceita conexões de múltiplos clientes simultaneamente. Cada cliente conectado recebe um número aleatório (entre 0 e 10) enviado pelo servidor a cada segundo.

- Implementação: O servidor cria um socket TCP, escuta por novas conexões e, ao aceitar uma conexão, cria uma nova thread para lidar com a comunicação com o cliente.

Funcionalidade do Servidor UDP:

- Porta: 9998
- Operação: O servidor recebe pacotes UDP contendo comandos de transação (comprar ou vender) e um preço. Ele verifica a validade do comando e responde com "success" se o comando for válido, ou "failure" caso contrário.
- Implementação: O servidor cria um socket UDP e aguarda pacotes de dados. Ao receber um pacote, ele processa a mensagem e envia uma resposta ao cliente.

Cliente

Descrição Geral:

- O cliente, desenvolvido em Java Swing, oferece uma interface gráfica amigável para que os usuários possam enviar comandos de compra e venda de ações ao servidor. A interface permite a interação fácil e intuitiva, facilitando a realização das operações desejadas pelos usuários. Os clientes TCP e UDP se conectam aos respectivos servidores, enviam comandos e processam as respostas recebidas.

Funcionalidade do Cliente TCP:

- Conexão: Conecta-se ao servidor TCP na porta 9999.
- Operação: Envia comandos digitados pelo usuário para o servidor e exibe as respostas do servidor. O usuário pode encerrar a conexão digitando "bye".
- Implementação: O cliente cria um socket TCP, conecta-se ao servidor, lê comandos do usuário e envia ao servidor, aguardando e exibindo as respostas.

Funcionalidade do Cliente UDP:

- Conexão: Conecta-se ao servidor UDP na porta 9998.
- Operação: Envia um comando de transação fixo ("buy 100") para o servidor e exibe a resposta ("success" ou "failure").
- Implementação: O cliente cria um socket UDP, prepara e envia um pacote de dados contendo o comando de transação, aguarda a resposta do servidor e a exibe.

A aplicação demonstra a implementação básica de servidores e clientes TCP/UDP, permitindo a comunicação bidirecional e simultânea entre eles. O servidor TCP envia dados continuamente para os clientes conectados, enquanto o servidor UDP processa e responde a comandos de transação. Os clientes foram projetados para interagir de forma simples com os servidores, exemplificando o envio e recebimento de dados em uma rede.

Configurações de Rede

Para a configuração da rede, foram utilizados dois roteadores e dois computadores, interconectados por cabos de par trançado com conector RJ45. As redes foram configuradas conforme as especificações de endereçamento IP fornecidas, garantindo a comunicação eficaz entre os dispositivos.

Ferramentas Utilizadas

O desenvolvimento do projeto foi realizado utilizando o Cisco Packet Tracer para a configuração da rede e o Wireshark para a avaliação do tráfego de rede, assegurando a correta implementação e funcionamento da aplicação.

Resultados

1. **Eficiência na Comunicação:** A aplicação demonstrou comunicação bidirecional e simultânea entre servidor e clientes, com o servidor TCP enviando dados continuamente e o servidor UDP processando e respondendo a comandos de transação de forma eficaz.
2. **Interface Gráfica Amigável:** A interface gráfica desenvolvida em Java Swing para os clientes permitiu uma interação fácil e intuitiva, facilitando a realização das operações desejadas pelos usuários.
3. **Configuração de Rede:** A utilização do Cisco Packet Tracer e do Wireshark permitiu a configuração adequada da rede e a avaliação do tráfego, assegurando a correta implementação e funcionamento da aplicação.
4. **Multithreading:** A utilização de multithreading no servidor permitiu a gestão eficiente de múltiplas conexões simultâneas, garantindo a robustez e a responsividade da aplicação.

5. **Versatilidade do Sistema:** O sistema desenvolvido pode ser adaptado e expandido para aplicações mais complexas, onde a comunicação eficiente e confiável entre servidores e clientes é crucial.

Cisco Packet Tracer

No nosso projeto, utilizamos o Cisco Packet Tracer para simular a infraestrutura de rede necessária para a comunicação entre o servidor e os clientes. Esta ferramenta permitiu a configuração e visualização de uma rede completa, composta por roteadores e computadores interconectados, essencial para testar a implementação dos protocolos TCP e UDP.

A simulação envolveu a configuração detalhada dos roteadores, a atribuição de endereços IP aos dispositivos e a conexão física entre eles usando cabos de par trançado. Com o Cisco Packet Tracer, pudemos garantir que todas as conexões estavam corretas e que a comunicação entre os dispositivos fluía como esperado, permitindo uma análise precisa e eficiente do comportamento da aplicação em um ambiente controlado.

Teste 1: Conectividade Usando Ping

Para verificar a conectividade entre os dispositivos na rede simulada no Cisco Packet Tracer, realizamos testes de ping entre os computadores, que estão em redes diferentes. Aqui estão os resultados do teste de conectividade:

Ping pc0 para para pc1

```
C:\>ping 172.16.0.2

Pinging 172.16.0.2 with 32 bytes of data:

Reply from 172.16.0.2: bytes=32 time=4ms TTL=126
Reply from 172.16.0.2: bytes=32 time=5ms TTL=126
Reply from 172.16.0.2: bytes=32 time=5ms TTL=126
Reply from 172.16.0.2: bytes=32 time=1ms TTL=126

Ping statistics for 172.16.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 5ms, Average = 3ms
```

Ping pc0 para para pc1

```

C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=5ms TTL=126
Reply from 192.168.1.2: bytes=32 time=5ms TTL=126
Reply from 192.168.1.2: bytes=32 time=6ms TTL=126
Reply from 192.168.1.2: bytes=32 time=4ms TTL=126

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 6ms, Average = 5ms

```

Teste 2: Tabela de Rotas dos Roteadores

Para garantir que os roteadores na rede simulada estão configurados corretamente e são capazes de encaminhar pacotes de dados entre diferentes sub-redes, realizamos uma verificação das tabelas de rotas em cada roteador. Este teste é crucial para validar a comunicação entre os dispositivos em diferentes segmentos da rede.

Tabela rotas R0:

```

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/30 is directly connected, Serial0/1/0
L       10.0.0.1/32 is directly connected, Serial0/1/0
      172.16.0.0/24 is subnetted, 1 subnets
S       172.16.0.0/24 [1/0] via 10.0.0.2
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0/0

```

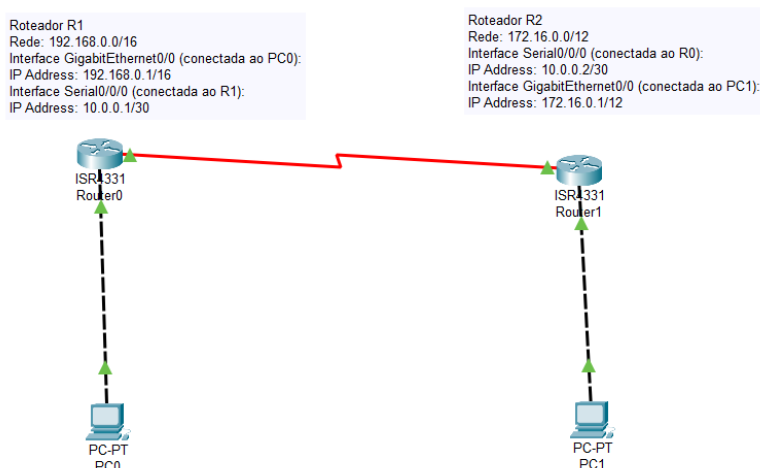
Tabela rotas R1

```

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/30 is directly connected, Serial0/1/0
L       10.0.0.2/32 is directly connected, Serial0/1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/24 is directly connected, GigabitEthernet0/0/0
L       172.16.0.1/32 is directly connected, GigabitEthernet0/0/0
S       192.168.1.0/24 [1/0] via 10.0.0.1

```

Interface Cisco Packet Tracer:



Conclusão

O projeto atingiu com sucesso seu objetivo principal de implementar um sistema de comunicação entre servidor e clientes utilizando os protocolos TCP e UDP. A aplicação desenvolvida demonstrou a funcionalidade básica de comunicação entre servidores e clientes, utilizando multithreading para garantir a eficiência e robustez da aplicação. Este sistema pode ser aplicado para diversas finalidades, como transmissão de dados em redes locais ou através da Internet.

A implementação do servidor mostrou-se capaz de lidar com conexões TCP e UDP simultaneamente, respondendo adequadamente aos comandos recebidos dos clientes. A utilização de sockets e threads permitiu a comunicação contínua e a interação em tempo real entre servidor e clientes.

Por outro lado, a implementação dos clientes, com a interface gráfica desenvolvida em Java Swing, facilitou a interação dos usuários com o sistema, permitindo o envio de comandos e a recepção de respostas de maneira intuitiva e eficiente. O cliente TCP foi capaz de enviar comandos e receber respostas contínuas do servidor, enquanto o cliente UDP processou transações de compra e venda de forma rápida e precisa.

