UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO: TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO
REDES DE COMPUTADORES II
PROF. ROBERTO VITO RODRIGUES FILHO
SARA FERRAZ MATEUS

Atividade 1

A escolha do protocolo de aplicação definida foi o TCP, melhor alternativa de acordo com a confiabilidade de transporte de dados de uma aplicação de chat. A lista de operações que o cliente deve realizar, junto com o protocolo de aplicação correspondente é a seguinte:

Lista de Operações	Protocolo de Aplicação	Como fazer
LOGIN	Envio do nome de usuário ao servidor. O cliente envia uma mensagem de inicialização contendo o nome de usuário após estabelecer a conexão TCP. O servidor envia uma mensagem de notificação ("<nome do usuário> entrou no chat") para todos os clientes ao receber uma nova conexão.	O cliente envia uma string com o nome de usuário para o servidor logo após a conexão.
SEND	Envio de mensagens para o servidor. O cliente envia uma mensagem de texto para o servidor, que então a retransmite para todos os clientes conectados (broadcast), com a identificação do remetente.	O cliente envia mensagens de texto para o servidor, que as retransmite para todos os outros clientes.
RECEIVE	Recebimento de mensagens do servidor. O servidor retransmite (broadcast) todas as mensagens para todos os clientes, exceto para o remetente. O cliente, ao receber, exibe a mensagem na interface.	O cliente recebe e exibe mensagens enviadas por outros usuários através da thread "receive_messages".
LOGOUT	Encerramento da conexão com o servidor. Quando um cliente se desconecta, o servidor envia uma mensagem de notificação ("<nome do usuário> saiu do chat") para todos os clientes restantes.	O cliente envia "sair" para encerrar a conexão e depois fecha o socket. O servidor então notifica os outros usuários sobre a saída.

Para a aplicação de chat, o protocolo definido foi o TCP (Transmission Control Protocol) é a escolha da camada de transporte devido à sua integridade e a ordem de todos os dados. Além de garantir que as mensagens sejam entregues corretamente, o TCP ajusta a velocidade de envio de dados conforme as condições da rede. Como é um protocolo orientado à conexão, ele permite a comunicação entre cliente e servidor, facilitando também a notificação de entradas e saídas de

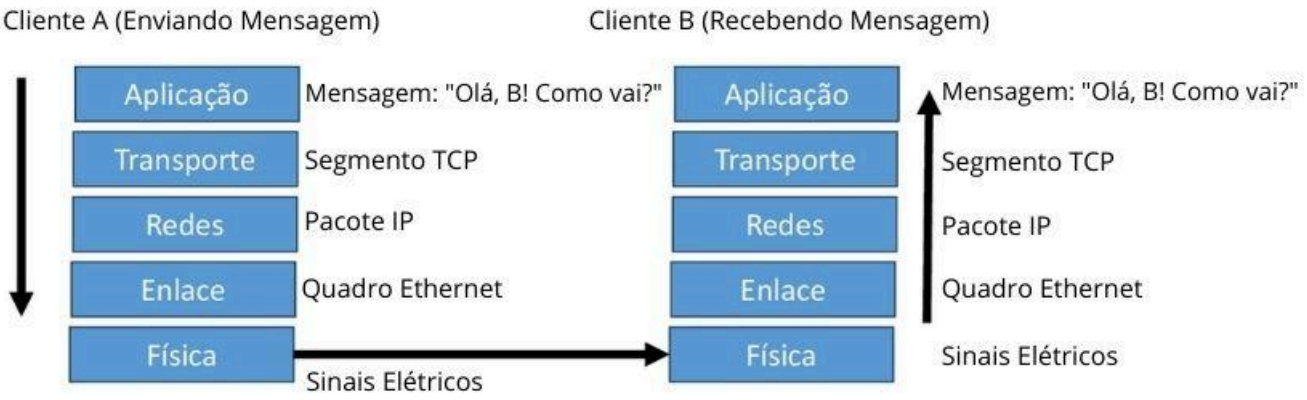
usuários, o que é importante em um chat. Outro ponto a favor é que a implementação em sockets TCP no Python é simples e eficiente. Portanto, a confiabilidade e a necessidade de manter a conexão ativa fazem do TCP a melhor escolha em comparação com protocolos como o UDP, que não oferece garantias de entrega, poderia ser usado em aplicações de chat em tempo real ou com necessidade de baixa latência, mas, para esse caso onde a integridade das mensagens é prioritária, o TCP é a melhor escolha.

O funcionamento do códigoé garantido pelo servidor que gerencia conexões de clientes, recebe e retransmite mensagens entre eles, e lida com a entrada e saída dos usuários. O cliente conecta-se ao servidor, envia e recebe mensagens, e pode se desconectar quando desejado. As threads são usadas para permitir a comunicação simultânea sem bloquear a execução principal do programa.

O servidor “server.py” aceita conexões de múltiplos clientes, gerenciando a comunicação entre eles. Inicialmente, ele cria e configura um socket TCP, vinculando-o ao IP local e à porta 5555, e começa a escutar novas conexões. Quando um cliente se conecta, o servidor cria uma nova thread para lidar com esse cliente, recebendo o nome de usuário e notificando os outros clientes sobre a entrada. Em um loop contínuo, o servidor recebe mensagens dos clientes e as retransmite para todos os outros, além de lidar com desconexões e notificações de saída.

O cliente “client.py” conecta-se ao servidor, envia seu nome de usuário, e permite ao usuário enviar e receber mensagens. Após a conexão, o cliente envia o nome de usuário e inicia uma thread para receber mensagens do servidor. O usuário pode enviar mensagens de texto, que são retransmitidas pelo servidor para todos os outros clientes. Para sair do chat, o cliente envia o comando "sair", fecha a conexão e encerra o programa.

Atividade 2



Para descrever o que acontece em cada camada da pilha de protocolo quando um cliente A envia a mensagem “Olá, B! Como vai?” para um cliente B em uma rede local cabeada (Ethernet), consideraremos a pilha de protocolos TCP/IP e os seguintes passos:

1. Camada de Aplicação:
    - Cliente A: O cliente A cria a mensagem “Olá, B! Como vai?” na aplicação e a envia para o servidor.
    - Cliente B: O cliente B está esperando mensagens do servidor para exibir na aplicação.
2. Camada de Transporte (TCP):
    - Cliente A: A mensagem é passada para a camada de transporte, onde o protocolo TCP divide a mensagem em segmentos e adiciona informações de controle (como números de sequência e confirmação). O segmento TCP é então enviado para a camada de rede.
    - Cliente B: O cliente B recebe o segmento TCP através da camada de rede e reagrupa os segmentos se necessário. A camada de transporte reassembla a mensagem original e a passa para a camada de aplicação.
3. Camada de Rede (IP):
    - Cliente A: O segmento TCP é encapsulado em um pacote IP. O protocolo IP adiciona cabeçalhos que contêm informações de

endereçamento (endereço IP de origem e destino). O pacote IP é então enviado para a camada de enlace.
    - Cliente B: O pacote IP é recebido na camada de enlace e é decapsulado. O endereço IP de destino é verificado e, se estiver correto, o pacote é passado para a camada de transporte.
4. Camada de Enlace (Ethernet):
    - Cliente A: O pacote IP é encapsulado em um quadro Ethernet. O protocolo Ethernet adiciona cabeçalhos que contêm endereços MAC de origem e destino. O quadro é então transmitido pela rede local.
    - Cliente B: O quadro Ethernet é recebido e verificado. O cabeçalho Ethernet é removido, e o pacote IP é passado para a camada de rede.
5. Camada Física:
    - Cliente A: O quadro Ethernet é convertido em sinais elétricos e transmitido através do cabo Ethernet.
    - Cliente B: Os sinais elétricos são recebidos e convertidos de volta em um quadro Ethernet.

Portanto, quando o Cliente A envia uma mensagem, ela é processada pela camada de aplicação e empacotada em um segmento TCP. Este segmento é encapsulado em um pacote IP e, por fim, em um quadro Ethernet. O quadro é convertido em sinais elétricos e transmitido pela rede. No Cliente B, os sinais elétricos são convertidos de volta em um quadro Ethernet, que é decapsulado para revelar o pacote IP. O pacote IP é então passado para a camada de transporte e, finalmente, para a camada de aplicação, onde a mensagem é exibida. Cada camada da pilha de protocolos desempenha a função de encapsular e descapsular os dados, adicionar informações de controle e garantir a entrega correta da mensagem ao destino.

REFERÊNSCIAS BIBLIOGRÁFICAS

ALURA. *Quais as diferenças entre o TCP e o UDP*. Disponível em: https://www.alura.com.br/artigos/quais-as-diferencas-entre-o-tcp-e-o-udp?. Acesso em: 17 set. 2024.