



UnB

Departamento de
Ciência da Computação

Redes de Computadores
Profa. Priscila Solís Barreto

PROJETO 1

Grupos de 03 pessoas
Data de Entrega: 29/07/2024

1. INTRODUÇÃO

A compreensão do desenvolvimento de aplicações cliente servidor e da compreensão dos protocolos de rede pode ser aprofundada ao “ver os protocolos em ação”, observando a sequência de mensagens trocadas entre duas entidades de protocolo, investigando os detalhes da operação do protocolo e fazendo com que os protocolos executem certas ações e, em seguida, observar essas ações e suas consequências. Isso pode ser feito em cenários simulados ou em um ambiente de rede “real”, como a Internet.

Neste projeto o objetivo é desenvolver uma aplicação na arquitetura cliente servidor e executá-la em uma rede em diferentes cenários usando seu próprio computador e uma rede privada. Este projeto requer a utilização de *sockets*, linguagens de programação e o aplicativo WIRESHARK, assim como da definição de um ambiente simples que utilize a arquitetura cliente servidor. O objetivo é aprofundar os conhecimentos na camada de aplicação e verificar e avaliar como os pacotes são enviados e recebidos entre um conjunto de clientes e um servidor, enquanto é utilizada uma aplicação de rede.

2. DESCRIÇÃO GERAL DA APLICAÇÃO A SER DESENVOLVIDA

Deverá ser escolhida um tipo de aplicação para desenvolver e implementar seu próprio protocolo. No relatório devem ser descritos tipo, formato e tamanho de mensagens, além da sequência de transmissão ao delas para a implementação das funcionalidades requeridas. As aplicações podem ser: (1) Streaming de vídeo (2) Chat em grupo ou (3) Utilitário de sincronização de arquivos.

Requisitos funcionais das Aplicações

Streaming de áudio	Chat em grupo	Utilitário de sincronização de arquivos
Cliente deve poder recuperar a lista de vídeos no servidor	Qualquer usuário pode se conectar no servidor de chat com o uso de e-mail válido.	O utilitário é simultaneamente cliente e servidor (<i>peer-to-peer</i>)
Cliente deve poder clicar para assistir um vídeo hospedado no servidor	Cada usuário deverá ter um profile indicando seu nome, e-mail e localização, que poderá ser consultado por qualquer participante do servidor de chat.	Diferentes clientes devem ser capazes de se descobrir em uma rede local
Se o cliente tentar tocar a música e ela não estiver em cache local, buscar no servidor	Os usuários devem ser capazes de listar os grupos de chat e usuários	O usuário deve ser capaz de adicionar uma pasta às pastas a serem sincronizadas
		O cliente onde a pasta foi adicionada, deve montar uma lista de arquivos,

O servidor deve transmitir o vídeo em blocos de 30 segundos	disponíveis para conversar e pedir para ingressar em diversos grupos. Cada grupo de chat terá um nome único.	datas de criação e modificação, além de calcular o hash criptográfico SHA1
O cliente deve poder pausar o vídeo, o que deve interromper a bufferização	Um usuário pode criar um grupo e convidar pessoas para participar do grupo. O criador do grupo será o administrador.	Outros clientes baixarão periodicamente uns dos outros listas de pastas compartilhadas
Se o cliente retomar a execução do ponto parado ou reiniciar o vídeo, o buffer local deve ser consumido	Um usuário pode sair do grupo quando quiser.	Outros clientes verão as pastas compartilhadas e poderão selecionar uma para iniciar a sincronização
Diferentes clientes devem ser capazes de se descobrir em uma rede local	O usuário pode enviar no grupo imagens, texto e arquivos de áudio.	O cliente requisitará a cópia mais recente disponível em seus pares, checará seu hash criptográfico, e caso o arquivo esteja íntegro, o salvará em uma pasta local
Clientes devem ser capazes de assistir um vídeo em um cliente remoto		

IMPORTANTE: Materiais adicionais para programação com sockets

Repositório com exemplos: <https://github.com/Gabrielcarver/Redes-de-Computadores-UnB/>
 Vídeo aula 1: <https://youtu.be/vavtMib9Uxo>
 Vídeo aula 2: <https://youtu.be/k9XP5aK0gk>
 Vídeo aula 3: https://youtu.be/Fq_hJek21aY

3. DESCRIÇÃO DA VERIFICAÇÃO E AVALIAÇÃO DA APLICAÇÃO

Após a definição da arquitetura da rede privada em que a sua aplicação deverá funcionar, que conste de um host servidor e de dois ou mais hosts do tipo cliente, esse ambiente deve permitir que os hosts de tipo cliente se conectem ao host tipo servidor por rede sem fio ou rede cabeada. O servidor pode estar em um notebook ou desktop por exemplo. Os clientes podem ser smartphones, tablets ou qualquer outro dispositivo que permita uma fácil interação com o servidor.

Comandos como ipconfig (para Windows) e ifconfig (para Linux / Unix) estão entre os pequenos utilitários mais úteis, especialmente para depurar problemas de rede. A ferramenta nslookup está disponível na maioria das plataformas Linux / Unix e Microsoft.

Caso utilize na sua rede um servidor DNS, pode executar o comando nslookup. Para executá-lo no Windows, usar prompt de comando e digitar nslookup na linha de comando.

Captura de Pacotes na rede

1. Inicie o serviço do lado do servidor e conecte dois ou mais dos seus clientes nesse servidor.
2. Inicie o Wireshark e antes de capturar, digite o nome do serviço na janela de especificação do filtro de exibição, de modo que apenas as mensagens do serviço de interesse sejam exibidas posteriormente na janela de listagem de pacotes.
3. Espere um pouco mais de um minuto, e em seguida, comece a captura de pacotes Wireshark.
4. Use seus clientes para enviar ou solicitar informação do servidor. Faça várias interações.
5. Pare a captura de pacotes do Wireshark.

Quadro 1

Ao observar as informações nas mensagens entre o servidor e o cliente, responda às seguintes perguntas. **Nas suas respostas, deve constar o a tela das mensagens enviadas e recebidas e indicar em que parte da mensagem foram encontradas as informações.**

- A. Identificação da versão ou tipo da aplicação no servidor que está executando.
- B. Qual é o endereço IP dos clientes? Qual o endereço IP do servidor?
- C. Qual o protocolo de transporte usado pelos clientes e pelo servidor? TCP ou UDP?
- D. Qual a porta de destino do cliente? E qual é a porta de origem do cliente?
- E. Identifique a carga útil dos pacotes entre cliente e servidor. Correspondem ao que é esperado que seja transmitido conforme o funcionamento da sua aplicação desenvolvida? Justifique e demonstre a sua resposta.
- F. Ao inspecionar os dados brutos na janela de conteúdo do pacote, identifique os cabeçalhos de todas as camadas e faça uma analogia deste material com os conceitos teóricos que foram estudados em sala de aula sobre encapsulamento.

4. Instruções de Entrega do Relatório

Importante: utilizar o formato definido no Aprender3 (Definição do Formato Geral de Relatórios de Projetos).

Deve ser elaborado **um único relatório** (em formato pdf), a ser entregue na plataforma Aprender3 (um único relatório para todos os integrantes do grupo) que deve conter:

1. Capa com identificação dos integrantes (matrícula e nome).
2. As seções definidas no formato de relatório disponível no Aprender3 da disciplina, no qual na seção experimental/análise de resultados deve conter:
 - a. Para a parte 2, uma descrição (com figuras explicativas e comentadas) da concepção da solução adotada, justificativa das escolhas na camada de transporte, da linguagem, das bibliotecas e uma descrição da interface de usuário.
 - b. Para a parte 3, uma descrição (com figuras explicativas e comentadas) da rede definida, descrição dos elementos básicos da configuração do cliente e servidor e exemplificação do funcionamento dos serviços. Para o Quadro 1, as respostas justificadas dos itens de A até E, com as imagens/telas impressas que demonstrem os resultados obtidos. Importante observar que cada resposta deve estar justificada com base nas telas e imagens capturadas.
 - c. Para as partes a e b, devem ser incluídas referências do material consultado.
3. O relatório deve conter um link para um vídeo de demonstração pelos integrantes do grupo (max 8 min) da execução da seção 3, em que seja demonstrado o funcionamento do servidor, a sua implementação e a obtenção e análise dos resultados do Quadro 1.