

**UnB**Departamento de
Ciência da Computação**Redes de Computadores – Turma B**
Profa. Priscila Solís Barreto**PROJETO 1****Grupos de 03 pessoas****Data de Entrega: 16/Janeiro/2023 até 23:55****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 programa 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: servidor e cliente de IRC

O objetivo deste projeto é adquirir experiência no desenvolvimento de aplicativos de rede simultâneos. O projeto exige o desenvolvimento de uma solução para implementar um servidor de bate-papo na Internet usando um subconjunto do protocolo Internet Relay Chat (IRC). O IRC é um sistema de bate-papo global, distribuído e em tempo real que opera na Internet. Uma rede IRC consiste em um conjunto de servidores interconectados. Uma vez que os usuários estejam conectados a um servidor IRC, eles podem conversar com outros usuários conectados a qualquer servidor na rede IRC. O IRC oferece comunicação em grupo, por meio de canais nomeados, bem como comunicação pessoal por meio de mensagens “privadas”. Para obter mais informações sobre IRC, incluindo software cliente disponível e redes públicas de IRC, consultar: <http://www.irchelp.org/irchelp/rfc/> e <http://www.irchelp.org/irchelp/new2irc.html>.

Neste projeto, o solicitado é **implementar um servidor IRC autônomo**. Para tal, deve-se assumir que existe apenas um servidor IRC e todos os clientes estão conectados a esse servidor. As definições para esse servidor são as seguintes:

- **nodeID** – identificador único que identifica um servidor IRC, ou nó. O nodeID para o servidor IRC independente deve ser 1.
- **destino** – apelido ou canal de IRC como uma cadeia de caracteres terminada em nulo. De acordo com o RFC do IRC, os destinos terão no máximo 9 caracteres e não podem conter espaços.
- **Porta IRC** – A porta TCP no servidor IRC que se comunica com os clientes.

O servidor a ser desenvolvido deve implementar um subconjunto do protocolo IRC original. O protocolo IRC original é definido no RFC 1459, entretanto, para facilitar sugere-se fortemente utilizar a versão anotada dessa RFC (<http://www.cs.cmu.edu/~srini/15-441/F06/project1/rfc.html>). As funcionalidades mínimas que o servidor deve implementar são as seguintes:

Comandos Básicos

- **NICK** – Dar um apelido ao usuário ou alterar o anterior. O servidor deve relatar uma mensagem de erro se um usuário tentar usar um apelido já usado.
- **USER** – Especificar o nome de usuário, nome do host e nome real de um usuário.
- **QUIT** – Finalizar a sessão do cliente. O servidor deve anunciar a saída do cliente para todos os outros usuários que compartilham o canal com o cliente que está saindo.

Comandos de canal

- **JOIN** – Começar a ouvir um canal específico. Embora o protocolo IRC padrão permita que um cliente se junte a vários canais simultaneamente, o servidor implementado deve restringir um cliente a ser membro de no máximo um canal. Entrar em um novo canal deve fazer com que o cliente saia implicitamente do canal atual.
- **PART** – Sair de um canal específico. Embora um usuário possa estar em apenas um canal por vez, PART ainda deve lidar com vários argumentos. Se esse canal não existir ou existir, mas o usuário não estiver nesse canal, deve ser feito o tratamento de erro.
- **LIST** – Listar todos os canais existentes apenas no servidor local. O servidor deve ignorar os parâmetros e listar todos os canais e o número de usuários no servidor local em cada canal.

Comandos Avançados

- **PRIVMSG** – Envia mensagens aos usuários. O alvo pode ser um apelido ou um canal. Se o destino for um canal, a mensagem deve ser transmitida para todos os usuários no canal especificado, exceto o originador da mensagem. Se o alvo for um apelido, a mensagem será enviada apenas para esse usuário.
- **WHO** – Consulta informações sobre clientes ou canais. Neste projeto, seu servidor só precisa oferecer suporte a canais de consulta no servidor local. Ele deve fazer uma correspondência exata no nome do canal e retornar os usuários desse canal.

Para todos os outros comandos, o servidor deve retornar `ERR UNKNOWNCOMMAND`. O servidor deve ser capaz de suportar vários clientes simultaneamente. O único limite para o número de clientes simultâneos deve ser o número de descritores de arquivo disponíveis no sistema operacional. Enquanto o servidor espera que um cliente envie o próximo comando, ele deve ser capaz de lidar com as entradas de outros clientes. Além disso, o servidor não deve desligar se um cliente enviar apenas um comando parcial. Em geral, a simultaneidade pode ser alcançada usando threads múltiplos. É importante considerar que o servidor não deve estar vulnerável a um cliente mal-intencionado.

Os detalhes da implementação (linguagem, APIs e interfaces de usuário) ficam a critério da equipe do projeto e devem ser justificados e detalhados no relatório. Esses itens serão considerados na avaliação deste projeto.

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. Em sua operação mais básica, a ferramenta `nslookup` permite que o host que executa a ferramenta consulte um registro DNS em qualquer servidor DNS especificado. O servidor DNS consultado pode ser um servidor DNS raiz, um servidor DNS de domínio de nível superior, um servidor DNS autoritativo ou um servidor DNS intermediário (no livro e nas vídeo aulas são explicados estes conceitos). Para realizar essa tarefa, o `nslookup` envia uma consulta DNS ao servidor DNS especificado, recebe uma resposta DNS desse mesmo servidor DNS e exibe o resultado. Ao executar o `nslookup`, se nenhum servidor DNS for especificado, o `nslookup` enviará a consulta ao servidor DNS padrão. A sintaxe geral dos comandos `nslookup` é: `nslookup -option1 -option2 host-to-find dns-server`

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. Corresponde ao que é esperado que seja transmitido conforme o funcionamento da sua aplicação desenvolvida?
- 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 (matricula 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 adoptada, 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 8min) 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.