UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

UNIVERSIDADE FEDERAL DE PERNAMBUCO – CAMPUS RECIFE
INFRAESTRUTURA DE COMUNICAÇÃO

RAFAEL SILVA MOURA

RELATÓRIO DO PROJETO DE SOCKET

Recife - PE

Dezembro 2025.

APRESENTAÇÃO

Este projeto foi desenvolvido como trabalho final da disciplina **Infraestrutura de Comunicação**, no **Centro de Informática (CIN) da UFPE.** O objetivo principal foi implementar aplicações cliente-servidor utilizando os **protocolos TCP e UDP**, acompanhadas de um Servidor de Nomes para registro e consulta de serviços disponíveis.

Para exemplificar o funcionamento, foi criado um **serviço de calculadora remota**, permitindo que os clientes realizassem operações matemáticas simples. As aplicações registram automaticamente os serviços no **Servidor de Nomes** ao serem iniciadas, e os clientes podem verificar a disponibilidade do serviço, incluindo testes de serviços inexistentes para validar o retorno “NOT_FOUND”.

O projeto inclui a **medição do tempo de envio e recebimento das mensagen**s, análise do desempenho dos protocolos e captura de pacotes via Wireshark, proporcionando compreensão prática da diferença entre TCP e UDP e do mecanismo de registro e descoberta de serviços em rede.

Sumário

OBJETIVOS E REQUISITOS DO PROJETO

O projeto tem como objetivos **implementar aplicações cliente-servidor** utilizando os protocolos **TCP e UDP,** permitindo a comunicação entre processos em rede de forma prática. Para exemplificar o funcionamento, foi criado um **serviço de calculadora remota,** no qual os clientes podem realizar operações matemáticas simples e receber os resultados de forma confiável no TCP ou de forma rápida no UDP.

Além disso, o projeto desenvolve um **Servidor de Nomes** que permite registrar e consultar serviços, garantindo que os clientes possam localizar os servidores disponíveis e validar serviços inexistentes, recebendo corretamente o retorno "NOT_FOUND". Outro objetivo importante é **analisar o desempenho dos protocolos de transporte,** medindo o tempo de envio e recebimento das mensagens, e **capturar o tráfego de rede via Wireshark,** permitindo comparar as características de TCP e UDP na prática. Por fim, o projeto busca **proporcionar aprendizado aplicado** sobre arquitetura cliente-servidor, protocolos de transporte e descoberta de serviços em rede.

Para atingir esses objetivos, foram estabelecidos **requisitos funcionais** e **não funcionais.** Entre os **requisitos funcionais**, destacam-se a implementação de servidores e clientes TCP e UDP capazes de processar operações matemáticas simples, o registro de serviços no Servidor de Nomes, a possibilidade de consultar serviços existentes e validar serviços inexistentes com o retorno "NOT_FOUND", além da medição do tempo total de comunicação entre cliente e servidor.

Já os **requisitos não funcionais** incluem a execução simultânea das aplicações TCP e UDP, garantindo independência entre os processos, a utilização de valores pré-determinados para evitar interferência na medição de desempenho, a exibição de mensagens coloridas no terminal para facilitar a visualização de status e resultados, e a captura de pacotes via Wireshark para análise detalhada do tráfego de rede, permitindo uma avaliação prática das diferenças entre TCP e UDP.

ARQUITETURA DO SISTEMA

O sistema desenvolvido segue a **arquitetura cliente-servidor**, na qual os clientes enviam solicitações e os servidores processam as operações, retornando os resultados. Existem d**ois tipos de servidores**, TCP e UDP, que operam de **forma independente**, mas oferecem o **mesmo serviço de calculadora remota.** Todos os servidores se registram automaticamente no Servidor de Nomes ao serem inicializados, permitindo que os clientes descubram os serviços disponíveis e obtenham o endereço IP e a porta correspondentes.

O Servidor de Nomes responde com "NOT_FOUND" caso o serviço solicitado **não esteja registrado**. A comunicação TCP garante confiabilidade e entrega ordenada das mensagens, enquanto a comunicação UDP proporciona maior rapidez, ainda que sem garantias de entrega. Essa arquitetura permite e**xecutar testes simultâneos** dos dois protocolos, medir o tempo de envio e recebimento das mensagens e analisar o tráfego de rede por meio de capturas no **Wireshark,** proporcionando compreensão prática das diferenças entre TCP e UDP e do mecanismo de registro e descoberta de serviços.

IMPLEMENTAÇÃO

Na seção a seguir, são detalhadas as implementações dos diferentes componentes do sistema, incluindo os servidores e clientes TCP e UDP, o Servidor de Nomes e o cliente de teste para serviço inexistente. S**erão descritas as funcionalidades de cada elemento**, **o fluxo de comunicação entre clientes e servidores**, bem como os **mecanismos de registro** e c**onsulta de serviços.** Além disso, são destacadas as configurações utilizadas para medição de tempo de execução e a análise do tráfego de rede por meio de capturas no Wireshark, permitindo compreender o comportamento prático do sistema.

Servidor TCP e Cliente TCP:

I. O servidor TCP aguarda conexões de clientes e, para cada conexão, envia um menu de operações matemáticas.
II. Os clientes enviam a operação escolhida e os números, e recebem o resultado calculado.
III. Ao encerrar, o servidor atualiza o Servidor de Nomes, removendo o serviço registrado.

Servidor UDP e Cliente UDP:

I. O servidor UDP opera sem conexão, recebendo datagramas contendo a operação e os valores, processando e retornando o resultado diretamente ao cliente.
II. Os clientes enviam os pacotes de forma automática e recebem a resposta, permitindo medir o tempo de comunicação.
III. Também é realizado o registro e remoção do serviço no Servidor de Nomes.

Servidor de Nomes:

I. Permite que servidores registrem serviços com IP, porta e nome do serviço.
II. Atende consultas de clientes, retornando os dados do serviço ou `"NOT_FOUND"` caso não exista.
III. Mantém um controle de serviços ativos, permitindo consultas confiáveis pelos clientes.

Cliente de teste para serviço inexistente:

Envia solicitações de um serviço que não foi registrado, verificando se o Servidor de Nomes retorna corretamente `"NOT_FOUND"`.

Além disso, todas as aplicações foram configuradas para **medir o tempo total de execução** e enviar mensagens coloridas no terminal, facilitando a visualização do status e dos resultados. A captura de pacotes via **Wireshark** permitiu analisar o tráfego e comparar o desempenho entre TCP e UDP de forma prática.

EXECUÇÃO E TESTES

Para testar o funcionamento do sistema, **os servidores TCP e UDP foram iniciados em terminais separados**, juntamente com o Servidor de Nomes. Em seguida, os clientes foram executados de forma automática, enviando operações matemáticas pré-determinadas para cada servidor. Um dos clientes consultou o serviço TCP, outro o serviço UDP, e um terceiro cliente enviou uma requisição para um serviço inexistente, verificando o retorno "NOT_FOUND". Durante os testes, os servidores processaram corretamente as requisições, retornando os resultados esperados aos clientes, enquanto o Servidor de Nomes respondeu conforme o registro de cada serviço. **Todos os pacotes trocados foram capturados via Wireshark,** permitindo analisar detalhadamente o tráfego de rede e validar o funcionamento do sistema.

A análise temporal das operações mostrou que, apesar do UDP ser **teoricamente mais rápido** por não exigir handshake nem confirmação de entrega, o **TCP apresentou tempo total menor** (0,000345 segundos) em comparação ao UDP (0,000542 segundos). Isso ocorreu porque, em operações muito pequenas, como a soma de dois números, o TCP se beneficia do gerenciamento de **buffers do sistema operacional**, que permite agrupar pacotes e enviar múltiplos dados de uma só vez, enquanto no UDP cada pacote é tratado de **forma independente, aumentando o overhead.** Em redes locais, onde a latência de transmissão é mínima, esse efeito se torna mais evidente, **favorecendo o TCP**. Apesar desse resultado, o UDP continua sendo vantajoso em aplicações que e**nviam grandes volumes de dados** sem necessidade de confiabilidade, enquanto o TCP garante entrega confiável e ordenada, sendo ideal para aplicações que necessitam consistência nos dados.

FIGURA 1 – COMPORTAMENTOS DOS SEVIDORES

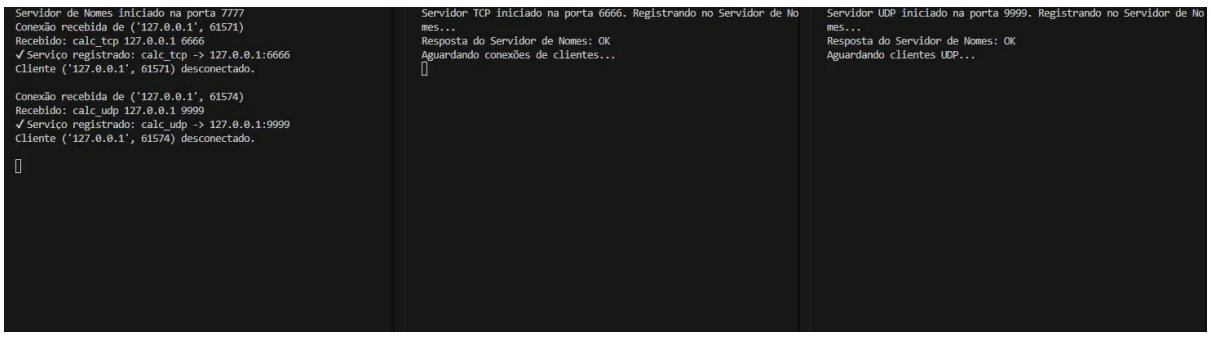
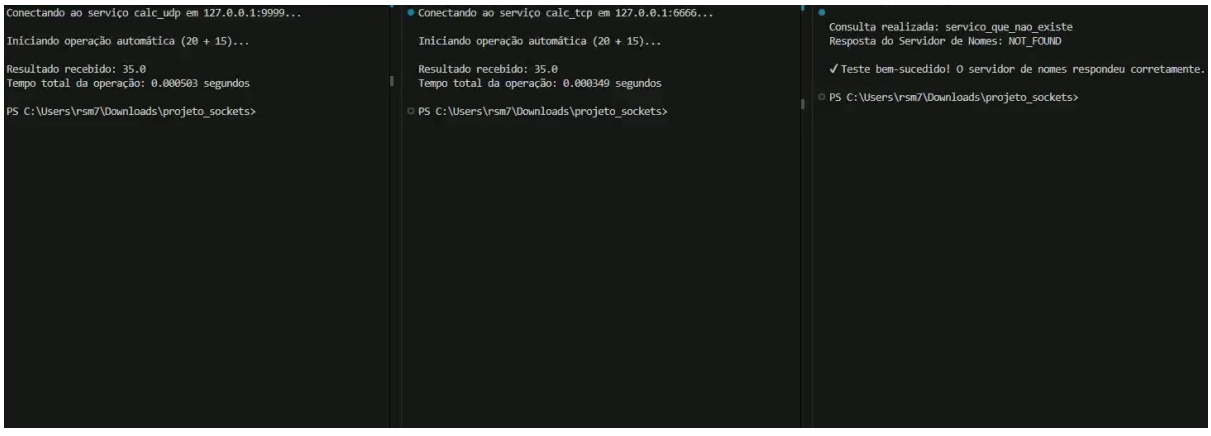


FIGURA 2 – COMPORTAMENTOS DOS CLIENTES



DESCRIÇÃO TÉCNICA DOS ARQUIVOS E JUSTIFICATIVA DAS CAPTURAS

Nesta sessão será apresentada a **descrição do serviço de comunicação desenvolvido**, seguido da análise detalhada das capturas realizadas utilizando o Wireshark (arquivo **capturas_wireshark.pcapng**). O objetivo é correlacionar o comportamento observado durante a execução dos serviços com os protocolos de transporte empregados (TCP e UDP) e, por fim, realizar uma comparação estruturada entre os tempos de resposta de cada protocolo.

Obs: Nas descrições das capturas no Wireshark, não mencionarei explicitamente os pacotes de confirmação ACK. Entretanto, considere que para cada pacote enviado houve o respectivo recebimento e confirmação.

1. Descrição do Serviço Desenvolvido

O sistema é composto por **três partes principais**: um Servidor de Nomes, um Serviço de Calculadora TCP e um Serviço de Calculadora UDP. Cada componente possui uma **função específica** dentro da arquitetura.

1.1 Servidor de Nomes (*service_names.py*)

O Servidor de Nomes funciona como um **diretório centralizado**. Sua função é receber o registro dos serviços que desejam ser localizados na rede, armazenando seu nome, endereço e porta. Além disso, **responde a consultas de clientes** que precisam descobrir onde determinado serviço está ativo.

Esse mecanismo permite que clientes e servidores funcionem de forma **desacoplada**, sem a necessidade de configurar portas manualmente para cada comunicação.

1.2 Serviço de Calculadora TCP (*service_tcp.py*)

A versão TCP da calculadora estabelece uma **conexão confiável** com o cliente. Após o estabelecimento da conexão, o servidor envia um menu de operações matemáticas, recebe a escolha do cliente e solicita os números. Todo o fluxo ocorre de maneira **sequencial e controlada**, com garantia de entrega.

Esse serviço registra-se previamente no Servidor de Nomes, informando o nome do serviço, IP e porta de funcionamento.

1.3 Serviço de Calculadora UDP (*service_udp.py*)

A versão UDP da calculadora realiza operações semelhantes, porém sem estabelecer conexão. Cada troca entre cliente e servidor ocorre por meio de datagramas independentes. Apesar de teoricametne ser mais rápido, esse método não garante entrega nem ordenação, mas é suficiente para operações simples.

Assim como a versão TCP, o serviço envia seu registro ao Servidor de Nomes no momento em que inicia.

---

2. Análise das Capturas Wireshark

Nesta seção, além da explicação dos eventos, são indicados os frames exatos do arquivo ***capturas_wireshark.pcapng*** onde cada evento pode ser encontrado. A seguir estão descritos os principais eventos observados no arquivo capturas_wireshark.pcapng, acompanhados das indicações dos pacotes correspondentes para inserção das imagens.

2.1 Registro do Serviço TCP no Servidor de Nomes

Nos primeiros pacotes da captura, observa-se o estabelecimento da conexão TCP com o Servidor de Nomes e, em seguida, o **envio da mensagem contendo o nome do serviço** e sua porta de operação.

Imagem:

No.	Time	Source	Destination	Protocol	Lengtl	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	58381 → 7777 [SYN] Seq=0
2	0.000049	127.0.0.1	127.0.0.1	TCP	56	7777 → 58381 [SYN, ACK]
3	0.000106	127.0.0.1	127.0.0.1	TCP	44	58381 → 7777 [ACK] Seq=1

(3-way handshake do servido TCP para com o servidor de nomes)

```
> Frame 4: 67 bytes on wire (536 bits), 67 bytes captured (536 bits
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 58381, Dst Port: 7777, S
v Data (23 bytes)
    Data: 63616c635f746370203132372e302e302e312036363636
    [Length: 23]
```

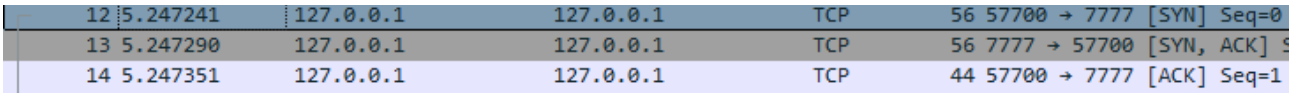
```
00 00 00 45 00 00 3f  82 57 40 00 80 06 00 00   ···E··?· ·W@·····
00 00 01 7f 00 00 01  e4 0d 1e 61 58 6f a0 56   ·····aXo·V
da 2d 4f 50 18 00 ff  a1 45 00 00 63 61 6c 63   ·-OP···· ·E··calc
74 63 70 20 31 32 37  2e 30 2e 30 2e 31 20 36   _tcp 127 .0.0.1 6
36 36                                           666
```

(frame 4: registro do servicço calc_tcp para o endereço 127.0.0.01 com a porta 6666)

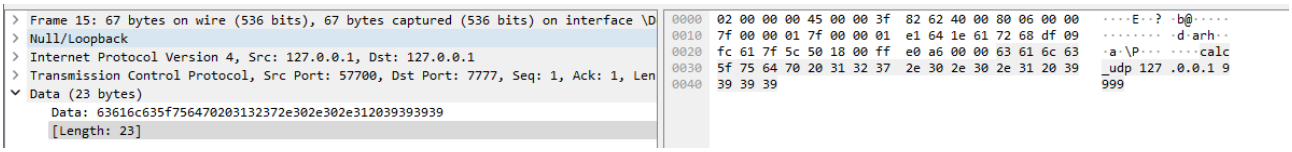
2.2 Registro do Serviço UDP no Servidor de Nomes

Após o registro TCP, observa-se nova conexão TCP sendo aberta, desta vez referindo-se ao registro do serviço UDP. A mensagem enviada segue o **mesmo formato**, diferenciando-se apenas pelo nome e porta atribuída ao serviço.

Imagem:



(3-way handshake do servido UDP que usa o TCP para registrar somente o seu serviço no servidor de nomes, uma vez que é necessário que não ocorra perda de informações)

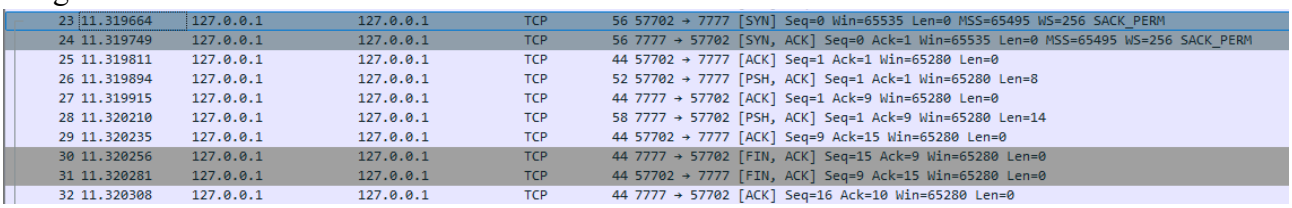


(frame 15: contém o envio do serviço UDP para o servidor de nomes)

2.3 Consulta ao Servidor de Nomes

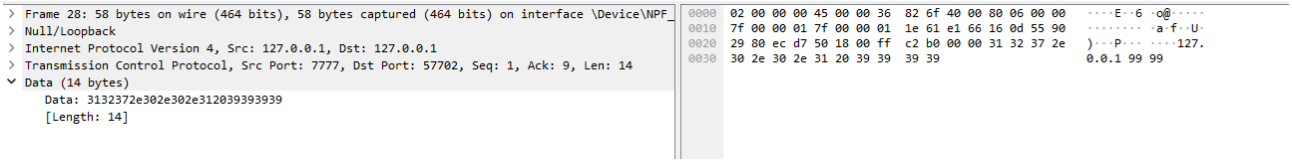
Em um momento posterior da captura, o cliente envia uma requisição ao Servidor de Nomes solicitando o endereço do serviço UDP. A **resposta contém o IP e porta** previamente registrados.

Imagem:



(3-way handshake do cliente UDP que usa o TCP para consultar somente o serviço “calc_udp” no servidor de nomes, uma vez que é necessário que não ocorra perda de informações)

(frame 26: consulta. Frame 28: resposta do servidor de nomes com o ip e porta do servidor_udp)



(frame 30 e 31: fecha a conexão TCP)

2.4 Comunicação UDP – Cálculo Remoto

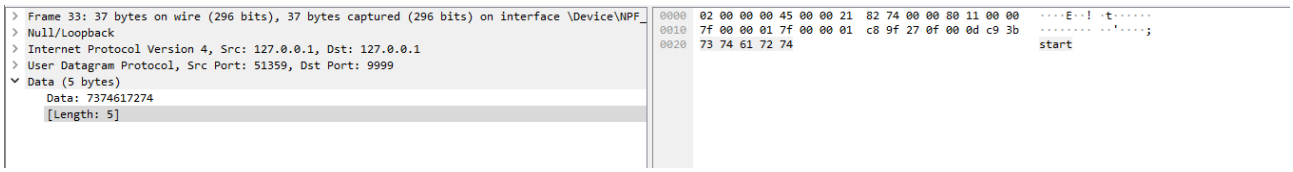
A comunicação via UDP apresenta uma série de datagramas curtos, incluindo:

- envio da mensagem inicial para iniciar o diálogo,
- envio do menu pelo servidor,
- envio da escolha da operação,
- envio dos números,
- retorno do resultado.

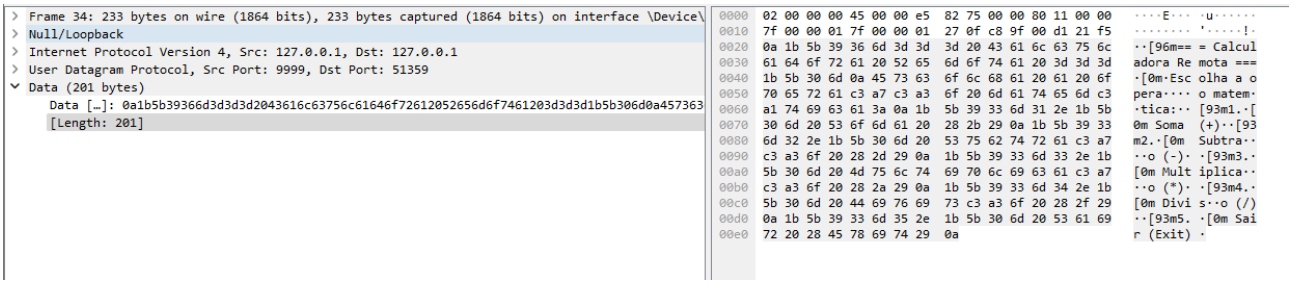
O tempo entre as mensagens intercambiadas é extremamente baixo, característica natural do protocolo UDP por não exigir handshake.

Imagens da captura:

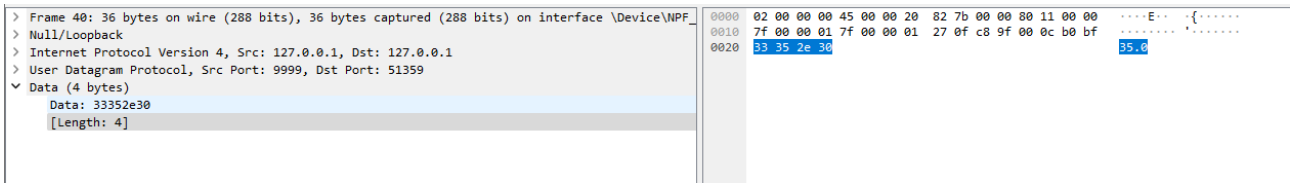
[Pacote inicial UDP (“start”) — localizar o **frame 33**]



[Pacote contendo o menu UDP — localizar o **frame 34** (payload maior)]



[Pacote contendo o resultado — localizar o **frame 40**]



2.5 Comunicação TCP – Cálculo Remoto

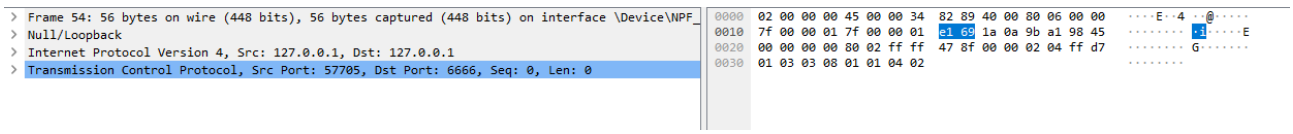
A comunicação TCP inclui:

- o handshake de três fases (SYN, SYN-ACK, ACK),
- envio do menu pelo servidor,
- envio dos números,
- retorno do resultado.

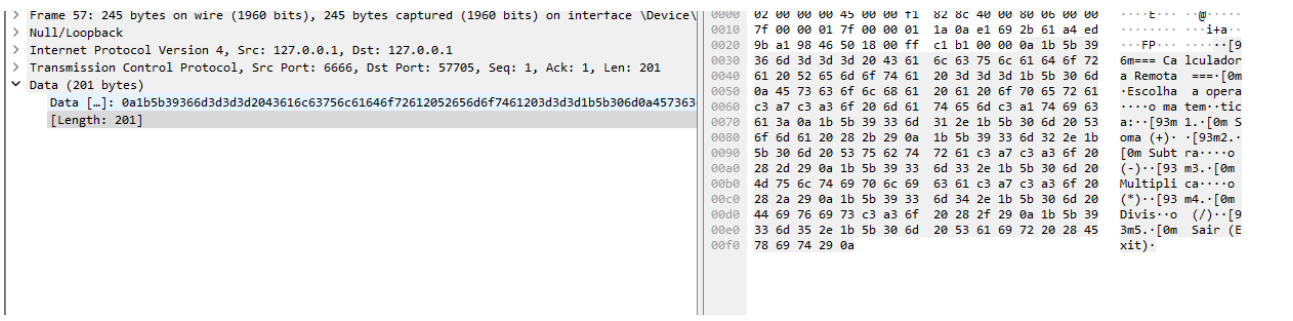
O tempo entre o início da conexão e o envio do menu é visivelmente maior que na versão UDP, em tese, devido ao custo do handshake, porém ao ver na prática a **conexão TCP foi mais rápida.**

Imagens:

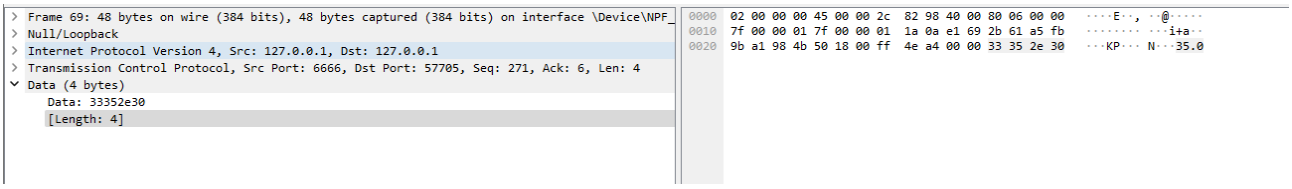
[Pacote SYN para o serviço TCP — localizar o **frame 54**]



[Pacote contendo o menu TCP — localizar o **frame 57**]



[Pacote contendo o resultado — localizar o **frame 69**]



---

3. Comparação dos Tempos TCP × UDP

Com base nas medições observadas na captura, é possível identificar diferenças claras no tempo de resposta entre os dois protocolos:

3.1 Desempenho do UDP

Os datagramas UDP apresentam tempos de resposta extremamente baixos. A ausência do handshake permite que o servidor responda imediatamente após o recebimento do pacote inicial. Em todas as interações observadas, o tempo entre pedido e resposta permaneceu na ordem de microssegundos. Todavia o **TCP foi mais rápido** do que o UDP nesse cenário.

3.2 Desempenho do TCP

A comunicação TCP apresenta tempos mais altos, principalmente devido à fase de estabelecimento da conexão. Mesmo após o handshake, o TCP continua exigindo confirmações de recebimento, o que aumenta ligeiramente o tempo total de cada operação.

3.3 Conclusão da Comparação

De forma geral, em teoria:

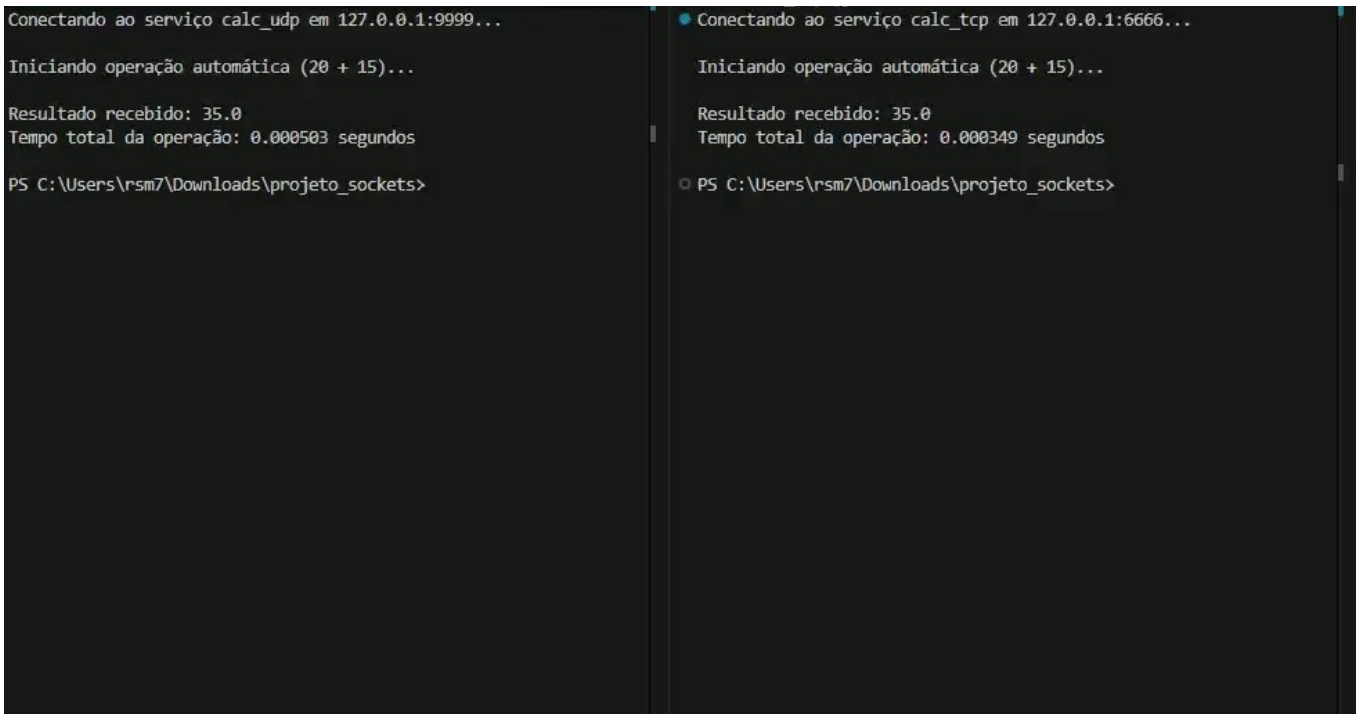
- UDP: mais rápido, menor latência, ideal para interações rápidas.
- TCP: mais confiável, porém ligeiramente mais lento.

Essa diferença foi confirmada na análise temporal da captura.

ANÁLISE DE DESEMPENHO

Durante os testes realizados, observou-se que a operação via **TCP apresentou tempo total menor** (0,000345 segundos) em comparação ao UDP (0,000542 segundos). Embora o **UDP seja teoricamente mais rápido** por não exigir handshake nem confirmação de entrega, no experimento realizado o TCP mostrou melhor desempenho. Isso ocorre porque, em **operações muito pequenas,** como a soma de dois números, o TCP se beneficia do gerenciamento de buffers do sistema operacional, que permite agrupar pacotes e e**nviar múltiplos dados de uma só vez.**

No UDP, cada pacote é tratado de **forma independente,** exigindo várias chamadas de envio e recebimento, o que **aumenta o overhead proporcionalmente.** Além disso, em redes locais, onde a latência de transmissão é mínima, o tempo gasto gerenciando pacotes e chamadas de sistema passa a dominar a operação, **favorecendo o TCP.** Apesar desse resultado, o UDP continua sendo vantajoso em aplicações que enviam grandes volumes de dados sem necessidade de confiabilidade, enquanto o TCP garante entrega confiável e ordenada, sendo ideal para aplicações que necessitam consistência nos dados.



CONSIDERAÇÕES FINAIS

O projeto permitiu a implementação prática de sistemas **cliente-servidor** utilizando os protocolos TCP e UDP, com registro e descoberta de serviços por meio de um **Servidor de Nomes**. Foi possível compreender a diferença entre os protocolos de transporte, medir o desempenho de cada um e validar o comportamento em situações de serviços inexistentes.

Além disso, a captura de pacotes via Wireshark reforçou a compreensão do tráfego de rede e da arquitetura do sistema. O trabalho proporcionou experiência prática em comunicação entre processos, programação de sockets e análise de desempenho de redes, consolidando os conceitos estudados na disciplina de **Infraestrutura de Comunicação**.