



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA (DETI)
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

ITALO AGUIAR DO NASCIMENTO PAULINO

**PERFORMANCE MUSICAL VIA REDE: ESTADO DA ARTE, IMPLEMENTAÇÃO E
ANÁLISE**

FORTALEZA

2023

ITALO AGUIAR DO NASCIMENTO PAULINO

PERFORMANCE MUSICAL VIA REDE: ESTADO DA ARTE, IMPLEMENTAÇÃO E
ANÁLISE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. José Marques Soares.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A229p Aguiar do Nascimento Paulino, Italo.
Performance Musical via Rede : Estado da arte, implementação e análise / Italo Aguiar do Nascimento Paulino. – 2023.
81 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia de Computação, Fortaleza, 2023.
Orientação: Prof. Dr. José Marques Soares.

1. Performance musical via rede. 2. Interface de áudio. 3. latência. I. Título.

CDD 621.39

ITALO AGUIAR DO NASCIMENTO PAULINO

PERFORMANCE MUSICAL VIA REDE: ESTADO DA ARTE, IMPLEMENTAÇÃO E
ANÁLISE

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Computação do Centro de Tecnologia da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de bacharel em
Engenharia de Computação.

Aprovada em: 14 de julho de 2023.

BANCA EXAMINADORA

Prof. Dr. José Marques Soares (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Michela Mulas
Universidade Federal do Ceará (UFC)

Prof. Dr. George André Pereira Thé
Universidade Federal do Ceará (UFC)

À minha família, por sua capacidade de acreditar em mim. Vó e Mãe, seu cuidado e dedicação foi o que deram a esperança para batalhar e criar caminhos frente aos medos que me assolaram.

AGRADECIMENTOS

À minha avó, Maria Lucimar Aguiar do Nascimento, por ter acreditado em mim e nutrido minha vida com tanto carinho, você a razão do meu viver.

À minha mãe, Maria Jarina Aguiar do Nascimento, pelo esforço de sustentar um e ter batalhado por mim sua vida inteira.

Aos amigos, Larissa Beatriz e Silas Façanha, pelo suporte nos últimos anos, por me ajudarem a conhecer quem eu sou e lembrarem sempre de meu potencial.

Aos amigos com quem morei na França, Luiz Sydrião, Renato Antoniassi, Gustavo Paixão, João Belintane, Ilan Vermeren, Leonardo Luquezi e Raul Adams, pelo acolhimento e companhia nessa batalha, pelas conversas sobre a vida conversas e apoio.

Aos amigos e colega de faculdade, pelas conversas, debates e ajudas nessa jornada, por terem me acolhido desde o início do curso e compartilhado conhecimento com bom grado.

À Prof.(a) Dr. Michela Mulas, minha orientadora durante a graduação, pelo acolhimento e ensinamentos desde o início do curso, por enxergado meu potencial e acreditado em mim.

Ao Pesquisador Dr. Gregory Pallone, meu orientador de estágio na França, por me valorizar e pela tamanha confiança em meu potencial e trabalho.

Ao Prof. Dr. José Marques Soares, pela excelente orientação e paciência durante esse trabalho.

Ao Governo da França, pelo apoio financeiro com a manutenção da bolsa de intercâmbio.

"No encontro com o outro, encontramos a nós
mesmos." (AGUIAR Italo, 2021)

RESUMO

Esta pesquisa fornece uma compreensão abrangente dos elementos técnicos essenciais para realizar Performances Musicais à distância. Explora soluções existentes, requisitos do sistema e metodologias de avaliação para enfrentar os desafios enfrentados pelos músicos na colaboração remota e permitir interações síncronas. Neste trabalho são examinados os requisitos técnicos necessários para alcançar comunicação com baixa latência no sistema de performance musical e enfatiza a importância de áudio de alta qualidade e colaboração eficaz. Também considera o impacto da distância de rede na latência e destaca as vantagens potenciais dos avanços em poder computacional, largura de banda de rede e tecnologia 5G. Ao considerar diversos cenários, como ensaios, aulas de música e concertos ao vivo, o artigo demonstra o potencial da interação síncrona em tempo real para promover a inovação musical. No geral, esta pesquisa contribui para a compreensão das Performances Musicais à Distância e oferece visões para o futuro.

Palavras-chave: Performance Musical à Distância, Interface de Áudio, Latência, Drivers

ABSTRACT

This research provides a comprehensive understanding of the technical elements essential for successful Network Musical Performance (NMP). It explores existing solutions, system requirements, and evaluation methodologies to address the challenges faced by musicians in remote collaboration and enable synchronous interactions. The article examines the necessary technical requirements for achieving low-latency communication in NMP and emphasizes the importance of high-quality audio and effective collaboration. It also considers the impact of network distance on latency and highlights the potential advantages of advancements in computational power, network bandwidth, and 5G technology. By considering various scenarios, such as rehearsals, music lessons, and live concerts, the article showcases the potential of synchronous real-time interaction to foster musical innovation. Overall, this research contributes to the understanding of NMP and provides insights for future advancements in network musical performance.

Keywords: Network Musical Performance, Latency, Drivers, Soundcards

LISTA DE FIGURAS

Figura 1 – Setup de testes (ROTTONDI <i>et al.</i> , 2015)	21
Figura 2 – System process that collaborate to over-all latency	24
Figura 3 – Interface de Áudio Scarlett 2i2	25
Figura 4 – Funcionamento do buffer interno de um interface de áudio	26
Figura 5 – Arquitetura ponto-a-ponto	30
Figura 6 – Arquitetura cliente-servidor	31
Figura 7 – Arquitetura ponto-a-ponto <i>multicast</i>	32
Figura 8 – Arquitetura cliente-servidor <i>multicast</i>	32
Figura 9 – Interfaces principal do <i>Jack Audio Connection Kit</i>	39
Figura 10 – Interfaces de configuração do <i>Jack Audio Connection Kit</i>	39
Figura 11 – Execução do <i>Jacktrip</i>	40
Figura 12 – Interfaces de configuração do <i>Jack Audio Connection Kit</i>	40
Figura 13 – Interface principal do <i>Jamulus</i>	43
Figura 14 – Lista de servidores públicos.	43
Figura 15 – Janela de sessões do <i>Jamulus</i>	44
Figura 16 – Janela de configuração do <i>Jamulus</i>	44
Figura 17 – Interface principal do Sonobus.	46
Figura 18 – Interface de configuração do Sonobus.	47
Figura 19 – Menu de conexão do Sonobus.	48
Figura 20 – Configuração da qualidade de áudio do Sonobus.	49
Figura 21 – Configuração do <i>buffer de jitter</i> do Sonobus.	49
Figura 22 – Protocolo de avaliação de latência de uma interface de áudio - Configuração da interface de áudio no computador do usuário	54
Figura 23 – Protocolo de avaliação da latência de uma interface de áudio - Configuração do computador de controle	55
Figura 24 – Protocolo de avaliação da latência de uma interface de áudio - Envio de áudio para o computador do usuário	56
Figura 25 – Protocolo de avaliação da latência de uma interface de áudio	57
Figura 26 – Forma de onda de áudio sintético	57
Figura 27 – Forma de onda de áudio sintético com zoom	58
Figura 28 – Intuição sobre como calcular o atraso entre dois sinais	59

Figura 29 – Valores de latência (ms) para diferentes tamanhos de <i>buffer</i> para a placa de som Scarlett 2i2 de 3ª geração	60
Figura 30 – Valores de latência (ms) para diferentes tamanhos de <i>buffer</i> para a placa de som Behringer UCA202	60
Figura 31 – Protocolo de avaliação do buffer de jitter e conexão de rede	61
Figura 32 – Fluxo de pacotes de rede durante as simulações	63
Figura 33 – Latência de rede usando conexão de rede local calculada usando o áudio capturado pelo computador de controle	65
Figura 34 – Latência de rede usando conexão de rede local calculada usando funções integradas do Jacktrip	65
Figura 35 – Latência global do sistema usando conexão de rede local calculada usando o áudio capturado pelo computador de controle	66
Figura 36 – Localização geográfica do usuário e do servidor	67
Figura 37 – Latência global do sistema usando conexão de rede de fibra óptica (cliente e servidor próximos) calculada usando o áudio capturado pelo computador de controle	68
Figura 38 – Latência global do sistema usando conexão de rede de fibra óptica (cliente e servidor distantes) calculada usando o áudio capturado pelo computador de controle	69
Figura 39 – Latência global do sistema usando conexão de rede 4G calculada usando o áudio capturado pelo computador de controle	70
Figura 40 – Latência de rede usando conexão de rede 4G medida usando funções integradas do Jacktrip.	71
Figura 41 – Latência global do sistema usando conexão de rede 5G comercial calculada usando o áudio capturado pelo computador de controle	72
Figura 42 – Latência de rede usando conexão de rede 5G experimental do Fornecedor1 calculada usando o áudio capturado pelo computador de controle	73
Figura 43 – Latência global do sistema usando conexão de rede 5G experimental do Fornecedor2 calculada usando o áudio capturado pelo computador de controle	73
Figura 44 – Latência global do sistema usando a conexão de rede 5G experimental do provedor1 com um servidor em <i>edge</i> calculada usando o áudio capturado pelo computador de controle	74

Figura 45 – Latência global do sistema usando a conexão de rede 5G experimental do provedor2 com um servidor em <i>edge</i> calculada usando o áudio capturado pelo computador de controle	75
--	----

LISTA DE TABELAS

Tabela 1 – Sincronicidade no tempo, fidelidade e percepção de latência para diferentes intervalos de atraso	22
Tabela 2 – Comparação entre <i>softwares</i> de performance musical remota	51
Tabela 3 – Resultados dos experimentos sobre a influência da conexão de rede em um sistema NMP	75
Tabela 4 – Sincronicidade no tempo, fidelidade e percepção de latência para diferentes intervalos de atraso	76

SUMÁRIO

1	INTRODUÇÃO	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Casos de Uso	18
2.2	Escopo Técnico	19
2.3	Estudo Psicoacústico e Técnico da latência nas Performances via Rede .	19
2.3.1	<i>Efeitos Psicoacústicos da latência nas Performances via Rede</i>	<i>19</i>
2.3.2	<i>Estudo Técnico da latência</i>	<i>22</i>
2.4	Estudo da técnico de elementos de um sistema	24
2.4.1	<i>Cadeia de Aquisição de áudio</i>	<i>25</i>
2.4.2	<i>Drivers</i>	<i>27</i>
2.4.3	<i>Processamento de Áudio</i>	<i>28</i>
2.4.4	<i>Sincronização da stream de áudio</i>	<i>28</i>
2.4.5	<i>Protocolos de Rede</i>	<i>29</i>
2.4.6	<i>Arquitetura de Rede</i>	<i>30</i>
2.4.7	<i>Jitter de Rede</i>	<i>32</i>
2.4.8	<i>Gerenciamento da aplicação</i>	<i>33</i>
2.4.9	<i>Considerações Finais</i>	<i>33</i>
3	METODOLOGIA	35
3.1	Definição da solução NMP existente a ser utilizada	35
3.2	Implementando um Sistema NMP	36
3.3	Análise Objetiva	36
3.4	Análise Subjetiva	37
4	EXPERIMENTOS E RESULTADOS	38
4.1	Avaliação de soluções NMP existentes	38
4.1.1	<i>JackTrip</i>	<i>38</i>
4.1.1.1	<i>Como usar ?</i>	<i>38</i>
4.1.1.2	<i>Arquitetura de rede</i>	<i>41</i>
4.1.1.3	<i>Interfaces e Design</i>	<i>41</i>
4.1.1.4	<i>Codec de Áudio</i>	<i>41</i>
4.1.1.5	<i>Licenças e plataformas suportadas</i>	<i>41</i>

4.1.1.6	<i>Estratégias para lidar com Jitter e Perda de pacotes</i>	41
4.1.1.7	<i>Características adicionais e limitações</i>	41
4.1.2	<i>Jamulus</i>	42
4.1.2.1	<i>Como usar ?</i>	42
4.1.2.2	<i>Arquitetura de rede</i>	45
4.1.2.3	<i>Interfaces e Design</i>	45
4.1.2.4	<i>Codec de áudio</i>	45
4.1.2.5	<i>Licenças e plataformas suportadas</i>	45
4.1.2.6	<i>Estratégias para lidar com Jitter e Perda de pacotes</i>	45
4.1.2.7	<i>Características adicionais e limitações</i>	45
4.1.3	<i>Sonobus</i>	45
4.1.3.1	<i>Como usar ?</i>	46
4.1.3.2	<i>Arquitetura de rede</i>	50
4.1.3.3	<i>Interfaces e Design</i>	50
4.1.3.4	<i>Codec de áudio</i>	50
4.1.3.5	<i>Licenças e plataformas suportadas</i>	50
4.1.3.6	<i>Estratégias para lidar com Jitter e Perda de pacotes</i>	50
4.1.3.7	<i>Características adicionais e limitações</i>	50
4.1.4	<i>Comparação das soluções NMP existentes</i>	51
4.2	<i>Implementando um NMP</i>	51
4.3	<i>Análise Objetiva</i>	52
4.3.1	<i>Influência do tamanho do buffer utilizado em uma interface de áudio . . .</i>	52
4.3.2	<i>Influência do tamanho do buffer de jitter da aplicação utilizado</i>	61
4.3.3	<i>Influência da conexão de rede na latência de um sistema NMP</i>	63
4.3.3.1	<i>Conexão Local</i>	64
4.3.3.2	<i>Fibra óptica</i>	66
4.3.3.3	<i>4G Comercial</i>	69
4.3.3.4	<i>5G Comercial</i>	71
4.3.3.5	<i>5G e Fortinet</i>	74
4.3.3.6	<i>Avaliação da influência da conexão de rede em um sistema NMP</i>	75
4.4	<i>Análise Subjetiva</i>	76
5	<i>CONCLUSÕES E TRABALHOS FUTUROS</i>	78

REFERÊNCIAS	80
--------------------	----

1 INTRODUÇÃO

No contexto da pandemia de COVID-19 e das políticas de distanciamento social, os músicos e suas bandas foram impedidos de tocar juntos, tendo suas atividades interrompidas. Além disso, o aumento do acesso a equipamentos e a popularização de estúdios caseiros associada a dificuldade de transporte de equipamentos, a prática em estúdios compartilhados torna-se cada vez mais difícil. Como as aplicações de videoconferência não são soluções viáveis para cenários síncronos e em tempo real, é necessário buscar outras soluções possíveis.

Com o objetivo de viabilizar a colaboração musical por meio de redes, pesquisas têm sido conduzidas desde os anos 70. A performance musical em rede (*Network Musical Performance*, NMP, sigla em inglês) representa uma interação síncrona mediada entre duas ou mais pessoas.

Vários estudos mostram que, para tocar juntos como se estivessem no mesmo local, os músicos precisam experimentar um atraso unidirecional abaixo de 25 ms a 30 ms ao ouvir um ao outro. Neste trabalho, focamos nos requisitos e soluções para garantir a interação simultânea de múltiplas localizações geográficas, preservando a qualidade do áudio, bem como os métodos de avaliação para estudar a qualidade das soluções.

As interações podem ser categorizadas de maneira diferente dependendo do tipo de conexões de rede utilizadas e da abordagem para conectar os participantes. Como nos NMPs os músicos estarão em regiões geograficamente separadas, eles serão conectados usando uma Rede de Área Ampla (WAN, na sigla em inglês). É importante ter em mente que a distância física é um dos elementos críticos nos sistemas NMP devido ao atraso de propagação que depende da distância percorrida pela informação de um usuário para outro. Em fibra óptica, a informação tem uma velocidade de transmissão de aproximadamente 80% da velocidade da luz, o que significa que se dois músicos estiverem a 720 km de distância (equivalente a um atraso de latência de transmissão de 30 ms de ida), eles não conseguirão tocar juntos adequadamente.

Com o objetivo de viabilizar a colaboração simultânea de vários músicos, enfrentamos vários desafios. Devemos atender aos requisitos de baixa latência para preservar o ritmo e a sincronia entre os músicos. Os equipamentos de áudio devem ser ajustados para garantir qualidade de áudio e baixa latência na captura e reprodução de áudio. Por fim, estratégias para evitar problemas de qualidade de áudio devido à oscilação da rede devem ser utilizadas. A oscilação da rede é a variação no tempo de transmissão dos dados do remetente para o receptor.

Na última década, graças ao aumento do desempenho computacional e da largura

de banda da rede, os NMPs estão se tornando possíveis e algumas soluções já foram propostas, como JackTrip (LABS, 2023), Jamulus (FISCHER, 2023), Sonobus (SONOSAURUS, 2023), Aloha (ELKAUDIO, 2023) e outros. Especialmente no contexto da pandemia, muitas soluções e estudos foram desenvolvidos para aprimorar a qualidade das soluções. Também estamos atentos à evolução do 5G, que pode oferecer vantagens para aplicativos de NMP por meio de melhorias na largura de banda e na evolução da latência.

A possibilidade de interação síncrona em tempo real pode abrir novos caminhos para inovações tecnológicas, embora nos concentremos especialmente no cenário de interação musical, como ensaios, aulas de música e concertos ao vivo.

O trabalho está estruturado da seguinte maneira. No Capítulo 2, apresentamos todos os elementos técnicos necessários para viabilizar a performance musical em rede, juntamente com uma análise das soluções existentes e dos requisitos do sistema. No Capítulo 3, descrevemos a metodologia utilizada para a implementação e análise do sistema, detalhando os passos e as abordagens adotadas. No Capítulo 4, discutimos os experimentos realizados e os resultados obtidos, analisando objetivamente e subjetivamente o desempenho e o funcionamento do sistema em diferentes cenários. Por fim, no Capítulo 5, apresentamos as conclusões alcançadas com base nos resultados dos experimentos e discutimos possíveis trabalhos futuros a serem realizados para aprimorar ainda mais a performance musical em rede. Essa estrutura foi planejada de forma a fornecer uma compreensão abrangente do tema, desde os fundamentos teóricos até a aplicação prática e as implicações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

De início, cumpre ressaltar alguns casos de uso e o escopo técnico do presente trabalho, a fim de melhor compreender a delimitação do objeto de estudo.

2.1 Casos de Uso

Acerca das interações musicais presenciais, estas podem ser categorizadas como: concertos, práticas individuais ou em grupo, ou colaborações musicais. As práticas individuais ou em grupo incluem todas as interações musicais entre instrumentistas e cantores, a exemplo de ensaios, estudos individuais ou em grupo, e sessões de improvisação musical. Sessões de improvisação musical são interações que priorizam o caráter de experimentação e de entretenimento. Concertos são performances musicais em que um ou mais músicos estão tocando para um público. Colaborações musicais incluem todas as interações entre um ou mais indivíduos para a criação de uma música ou arranjo musical. A fim de que a colaboração musical seja viável, algumas ferramentas podem ser necessárias, como *softwares* de notação musical ou estações de trabalho digitais (Digital Work Station - DAW). Vale ressaltar que as composições colaborativas têm sido uma prática cada vez mais comum na indústria musical ao longo das últimas décadas.

Os casos de utilização das performances musicais via rede vêm da categoria de música presencial. Uma prática de grupo via rede é um caso de uso da NMP em que é possível interagir por meio de uma rede como em práticas de grupo presenciais. Isso significa que a propagação do som no ar leva um tempo equivalente ao que a informação leva para chegar através de conexões de rede. Entretanto, os problemas de atraso aparecem quando a transmissão se torna maior do que a propagação do som em tempo real, presencialmente.

Como comparação, tem-se que os concertos via rede são equivalentes a práticas habituais com uma audiência. As colaborações musicais via rede dependem da comunicação entre todos os participantes e do acesso ao software, o que pode ser viabilizado pelo desenvolvimento de software colaborativo. O software colaborativo é uma aplicação para trabalhar em tempo real em uma tarefa comum.

Outro fator para ser levado em consideração nos casos de uso é o tipo de informação que é transmitida via rede. Pode-se transmitir áudio, vídeo ou sinal da Interface Digital de Instrumentos Musicais (MIDI). O áudio é o elemento mais necessário para viabilizar a NMP. Além disso, o vídeo pode ser um recurso interessante para algumas aplicações, como concertos

e aulas. Por fim, sinais MIDI são usados para se comunicar com instrumentos eletrônicos ou aplicações de computador que podem ser utilizados para fins colaborativos. Entretanto, são facilmente substituídos pela saída de áudio de instâncias controladas para práticas em grupo.

No escopo deste trabalho, o foco será dado nos requerimentos para as práticas em grupo via rede. Apenas a transmissão via áudio será considerada para protocolos de *design* e testes.

2.2 Escopo Técnico

A fim de melhor entender o funcionamento das performances musicais via rede, faz-se necessário um estudo e delimitação sobre o escopo técnico analisado.

2.3 Estudo Psicoacústico e Técnico da latência nas Performances via Rede

Para desenvolver uma solução para práticas em grupo via rede, é importante considerar os requisitos necessários para criar um ambiente que se assemelhe o máximo possível às situações reais. Isso envolve estudar a cadeia técnica e identificar as condições necessárias para a implementação adequada. Para estes requisitos, formalmente atribui-se o termo *requisitos psicoacústicos*. Os requisitos psicoacústicos estão relacionados à obtenção de condições sonoras semelhantes às das performances no mundo real, que estão majoritariamente relacionadas ao atraso na propagação do som, já mencionada. Vale também mencionar alguns elementos que podem ser tão importantes quanto, como a qualidade do áudio, que é uma medida do nível de fidelidade ao som original capturado, e o contato físico, que engloba o contato visual, gestos e expressão corporal. Esses não serão levados em consideração para definir as características críticas do sistema, mas podem ser analisados nas performances da solução.

2.3.1 Efeitos Psicoacústicos da latência nas Performances via Rede

Uma das características principais que possibilitará avaliar as performances musicais em rede é a medida do tempo individual e de referência. O tempo representa a velocidade ou ritmo de uma determinada composição musical, valor esse estabelecido como tempo de referência. O tempo individual é uma estimativa do tempo instantâneo ou médio de um indivíduo executando uma composição.

Por sua vez, define-se sincronicidade de tempo como a discrepância do tempo

individual entre dois ou mais músicos. Tal definição implica que, para alcançar a sincronicidade de tempo perfeita, todos os músicos devem manter o mesmo ritmo.

A fidelidade de tempo é a discrepância entre o tempo individual de um músico e o ritmo de referência de uma composição. A sincronicidade e a fidelidade de tempo serão empregadas para avaliar a influência da latência em práticas executadas por meio da rede.

Outro aspecto importante é a perceptibilidade da latência, que descreve a percepção subjetiva dos músicos em relação ao atraso nos sons. Este aspecto está diretamente relacionado ao conforto dos intérpretes. Estas características possibilitarão a definição do principal requisito: uma latência geral que permita fidelidade e sincronização de tempo quase ou não perceptíveis.

É importante ressaltar que a fidelidade de tempo perfeita é alcançada apenas por meio do uso de um metrônomo de referência. Um metrônomo é um dispositivo que produz cliques audíveis e regulares em um ritmo predefinido. Considerando que não se presume necessariamente o uso de um metrônomo durante as interações musicais, a "fidelidade de tempo natural", em oposição à perfeita, pode ser mais adequada à descrição de nosso contexto.

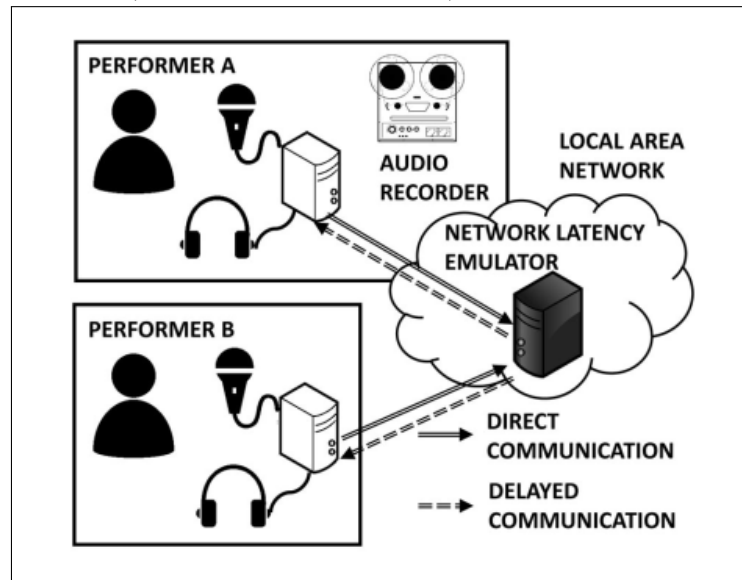
Em performances musicais presenciais, o som propaga-se pelo ar desde sua fonte até os ouvidos dos demais músicos. Considerando que as interações do mundo real implicam em distâncias físicas entre os músicos, naturalmente experimentam-se atrasos sonoros devido ao tempo de propagação. O som tem uma velocidade de propagação de 340m/s em meio aéreo, o que equivale a cerca de 3ms/m. Nota-se que músicos em apresentações presenciais lidam com uma limitação máxima de distância de 8,5m para uma boa manutenção da sincronicidade de tempo sem maestro, conforme indicado em (CARÔT, Germany, 2004) e (SCHUETT, 2002). Entretanto, tais artigos carecem de referência adequada e de protocolo de experimentação adaptado. Esta distância equivale a um atraso de 25ms na propagação causado pela viagem do som.

Enquanto no mundo real o tempo de propagação do som pelo ar indica o atraso da interação, nas performances em rede, o atraso ou a latência da comunicação é definido pelo intervalo de tempo entre a aquisição do som e seu momento de chegada aos ouvidos dos demais intérpretes, utilizando a rede como seu "meio de propagação".

A fim de avaliar a influência da latência nas práticas de grupo via rede, um protocolo de testes amplamente adotado é a simulação da latência da rede entre dois ou mais intérpretes em diferentes locais. Enquanto estes tocam padrões rítmicos ou estruturas mais complexas ao longo de ensaios variando a latência unidirecional, pelo menos o tempo individual e a sincronização de tempo são estimados com esta prática. A latência da rede é iniciada em 0ms e aumenta até

atingir os limites dos valores de atraso. Os atrasos limite são alcançados quando os praticantes não conseguem tocar devido ao desconforto perceptivo e/ou quando há impossibilidade de manutenção da sincronização de tempo. Um exemplo da disposição dos bancos de teste é mostrado na figura 1.

Figura 1 – Setup de testes (ROTTONDI *et al.*, 2015)



Fonte: (ROTTONDI *et al.*, 2015)

Para assegurar boas experiências, mantendo a sincronicidade de tempo, fidelidade natural de tempo e percepção de latência nula ou baixa, os valores de atraso devem estar na faixa de 8 a 25ms. De 25 ms a 55 ms, a latência é perceptível e os usuários experienciam perda de sincronicidade de tempo, bem como perda de fidelidade de tempo, devido à desaceleração do tempo individual. A desaceleração do tempo individual pode ser ocasionada como decorrência de estratégias conscientes ou inconscientes adotadas por um musicista para manter um tempo comum, a exemplo de seguir uma pessoa usando seu próprio tempo como referência, conforme sugerido por (FARNER *et al.*, 2009).

Um fenômeno interessante é notado quando os atrasos nas performances são menores que 8ms. Nesses casos, o tempo individual tende a acelerar. Foi sugerido que esse fenômeno se deve a uma tendência de antecipação. Isso pode estar relacionado ao fenômeno de assincronia média negativa (NMA, do inglês Negative Mean Asynchrony), onde as pessoas tendem a antecipar os toques do metrônomo ao tentar segui-lo. Infelizmente, ainda não há explicação científica para este fenômeno (REPP, 2005).

Os resultados são resumidos na tabela 4:

Tabela 1 – Sincronicidade no tempo, fidelidade e percepção de latência para diferentes intervalos de atraso

Latência (ms)	Sincronicidade no tempo	Fidelidade ao tempo	Perceptibilidade da latência
0-8	Impreciso	Aceleração	Nenhuma
8-25	Natural	Natural	Nenhuma
25-60	Impreciso	Desaceleração	Perceptível ou Tolerável
60 ou maior	Impossível	Impossível	Tolerável ou Intolerável

Fonte: elaborada pelo autor. (2021)

Outras pesquisas exploram a hipótese de que um metrônomo sincronizado em cada lado do usuário aumentaria a fidelidade e a sincronicidade do tempo. Os resultados mostram que isso não acarreta em um grande impacto sobre a sincronicidade do tempo entre os músicos, mas pode reduzir ou suprimir as perdas de fidelidade de tempo para uma ampla gama de latências (HUPKE *et al.*, 2019). Tal fenômeno é interessante para latências inferiores a 8ms, pois os problemas de aceleração são resolvidos. Ainda, a implementação de um metrônomo global pode ter um grande impacto sobre a qualidade e o conforto da performance em níveis mais altos de latência.

Nesse contexto, o objetivo da presente pesquisa consiste em implementar sistemas de Performances Musicais via Rede (NMP) com uma latência unidirecional de 25ms entre dois músicos, resultando em um tempo de ida e volta de 50ms.

2.3.2 Estudo Técnico da latência

Para tornar as Performances Musicais via Rede (NMP) viáveis, são necessários três elementos:

- Uma interface de áudio: responsável pela aquisição do áudio, sua transformação de analógico para digital e vice-versa, bem como a reprodução.
- Uma aplicação de NMP: responsável por conectar os usuários e enviar suas informações em pacotes pela rede. Também é responsável por receber informações de outros músicos.
- Conexão de rede: responsável por enviar informações de um usuário para os outros.

Quando os requisitos críticos de psicoacústica das NMPs são definidos, é possível analisar de forma mais crítica cada elemento técnico do sistema para entender suas contribuições para a latência geral e como eles funcionam.

Em uma performance musical em rede, à medida que um usuário toca ou canta, as ondas sonoras são capturadas por um microfone e são, em seguida, digitalizadas. Então,

as informações capturadas são empacotadas (*packetized*) e enviadas pela rede para os outros participantes. No lado do receptor, as informações são desempacotadas (*depacketized*) e transformadas em sinal analógico para ser transmitido através de fones de ouvido ou alto-falantes de monitoramento aos ouvidos dos usuários receptores. Esse processo acontece continuamente e em cada lado dos participantes, o que significa que eles recebem informações de todos os usuários e também transmitem seus sons.

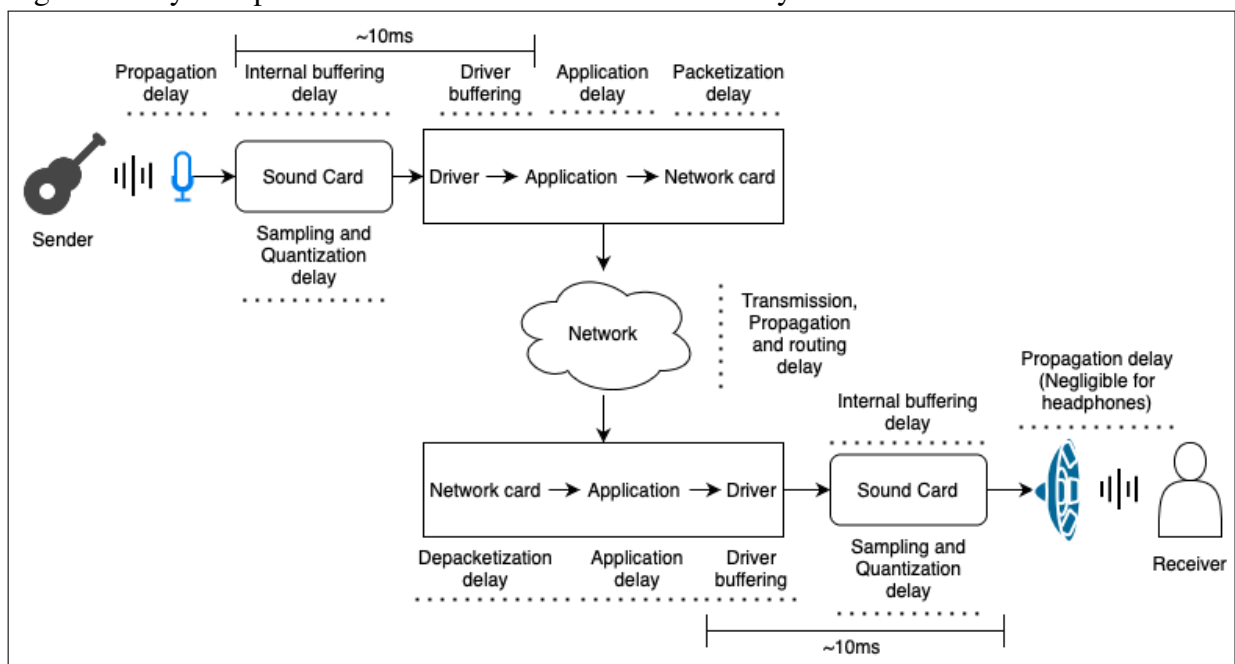
Cada processo contribui para a latência unidirecional geral, do ponto de origem ao ouvido, o que inclui:

1. Tempo de propagação do som da fonte para o microfone ou captador, que é um transdutor que captura ou detecta as vibrações mecânicas produzidas por instrumentos musicais. É adicionado um atraso de aproximadamente 3ms por metro.
2. Transdução da onda acústica para sinal elétrico e transmissões por cabos e conectores dos microfones para placas de som. Essas operações não têm influência considerável na latência.
3. Conversão analógico-digital e armazenamento interno em *buffer* da placa de som, dispositivo capaz de processar informações de entrada e saída de áudio. Isso pode adicionar de 2ms ou menos a 64ms ou mais de atraso, dependendo das configurações da placa de som. Entretanto, o atraso típico da placa de som é de cerca de 10ms.
4. Atraso de armazenamento em *buffer* do *driver* da placa de som, que é estimado como um múltiplo do atraso de armazenamento interno da placa de som, normalmente com duas ou três vezes o tamanho do *buffer* interno.
5. A aplicação pode aumentar os atrasos se forem utilizadas técnicas de codificação de áudio ou estratégias de aprimoramento de áudio. O atraso do processo depende da complexidade do algoritmo. Nesse contexto, a *aplicação* é a solução de NMP implementada para transportar informações de áudio pela rede.
6. Tempo de processamento para empacotar os dados de áudio antes da transmissão, o que não adiciona um atraso considerável, pois está na ordem de centenas de microssegundos.
7. Atraso de transmissão, propagação e roteamento na rede, que depende, respectivamente, dos limites de largura de banda e volume de fluxo de dados, distâncias físicas, condições de tráfego e algoritmos de roteamento. O atraso de transmissão de rede é o tempo necessário para enviar todos os bits do pacote pelo fio. Por exemplo, a transmissão de rede adiciona aproximadamente 210 ms/km. para fibras ópticas que transmitem dados a setenta por

- cento da velocidade da luz, se não forem considerados os atrasos dos equipamentos ativos.
8. Tempo de processamento para desempacotar os dados de áudio recebidos. Esse atraso depende principalmente do tempo dos algoritmos de decodificação.
 9. Atraso da aplicação, que depende do atraso de decodificação, de múltiplos algoritmos de mixagem de áudio e aprimoramento de áudio.
 10. Armazenamento em *buffer* do *driver* da placa de som e conversão digital-analógico, que dependem ainda das configurações da placa de som.
 11. Transmissão do sinal pelos fones de ouvido ou alto-falantes de monitoramento e transdução do sinal elétrico para onda acústica. Não adiciona atrasos consideráveis.
 12. Propagação do som dos alto-falantes para o ouvido, que depende da distância correspondente.

Esse processo pode ser visualizado na figura 2:

Figura 2 – System process that collaborate to over-all latency



Fonte: elaborada pelo autor. (2021)

2.4 Estudo da técnico de elementos de um sistema

As soluções ou aplicações das NPMs são definidas como todo o software desenvolvido para realizar o transporte de informações, interconectando dois ou mais usuários para apresentações musicais em rede. As soluções NMP podem ser desenvolvidas de diferentes formas, a depender dos seguintes fatores:

- Cadeia de captura de áudio: refere-se ao funcionamento e configuração da interface de áudio, bem como do *driver*.
- Técnicas de sincronização de fluxo de áudio e ocultação da perda de pacotes: que são usadas para garantir uma boa qualidade de áudio.
- Protocolos de rede: descreve o protocolo de transmissão usado.
- Arquitetura de rede: define como a aplicação irá interconectar os usuários.
- Gerenciamento da aplicação: como a aplicação gerencia todo o fluxo de dados.

2.4.1 Cadeia de Aquisição de áudio

Uma placa de som é um dispositivo capaz de processar informações de entrada e saída de áudio. Em apresentações musicais, geralmente é utilizada uma interface de áudio externa para processar áudio de maior qualidade. Esta interface opcionalmente pré-amplifica o sinal recebido, então seu conversor analógico-digital (ADC) provê amostras e quantifica o nível do sinal analógico, que é armazenado em um *buffer* interno como quadros de áudio. Por outro lado, as informações digitais são transmitidas como quadros de áudio em um *buffer* para a placa de som, na qual o conversor digital-analógico (DAC) associa as informações digitais aos níveis de tensão para reproduzir o áudio de forma analógica.

Figura 3 – Interface de Áudio Scarlett 2i2



Fonte: <https://focusrite.com/en/usb-audio-interface/scarlett/scarlett-2i2>.

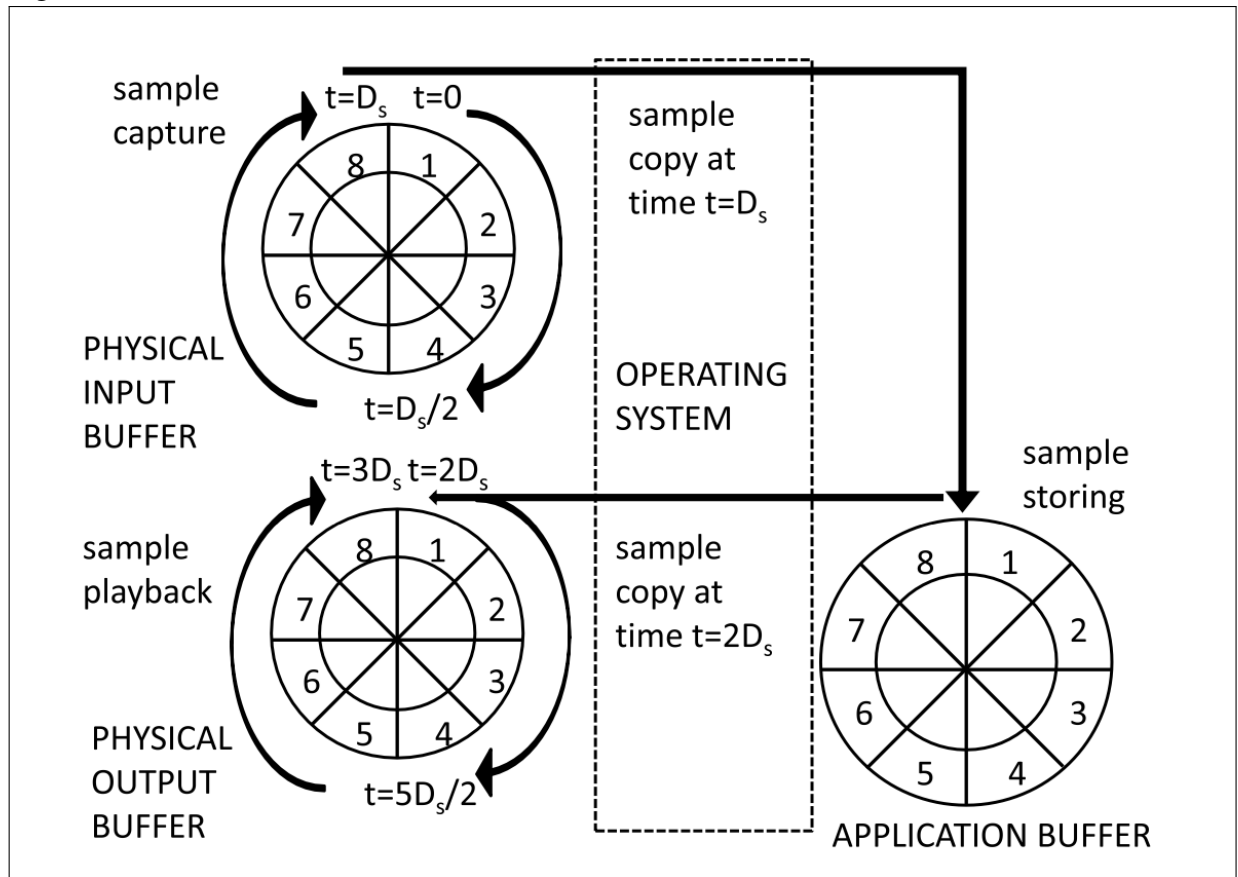
O uso de interfaces de áudio está geralmente associado a um computador que recebe e envia informações. As informações adquiridas armazenadas no *buffer* interno da placa de som são acessadas pelo computador por meio do uso de um *driver*. Um *driver* é um software usado para reconhecer e interagir adequadamente com um dispositivo, pois permite que as aplicações

acessem os *buffers* da placa de som.

Uma placa de som pode ser usada em três modos: captura, reprodução e monitoramento de som. No modo de captura, o sinal analógico amostrado convertido em sinal digital está disponível no buffer interno para ser capturado pela aplicação que controla a placa de som, essas informações podem ser gravadas, manipuladas ou redirecionadas através da rede para outros computadores. No modo de reprodução, o sinal digital enviado pelo computador é transmitido para a placa de som, essa informação pode ser o som capturado pela placa de som, que define o modo de monitoramento, o som capturado e manipulado com efeitos adicionais, bem como informações de áudio recebidas pela rede. Algumas interfaces de áudio oferecem um modo de monitoramento direto ou de latência zero no qual o sinal de entrada é direcionado diretamente para a saída.

O *driver* que controla a placa de som acessa seu buffer interno para extrair ou fornecer informações. O *driver* usa uma estratégia de fila para evitar situações de sub-execução e sobrecarga do *buffer*, onde as informações podem ser perdidas, mais especificamente, armazena um número predefinido de *buffers*, recebendo e transmitindo um por um.

Figura 4 – Funcionamento do buffer interno de um interface de áudio



Fonte: (ROTTONDI *et al.*, 2016)

A Figura 4 mostra como funcionam o buffer de entrada físico (*buffer* interno), o *buffer* da aplicação (*buffer* do *driver*) e o *buffer* de saída físico (*buffer* de reprodução). Uma interface de áudio possui alguns parâmetros que influenciarão a qualidade do áudio ou a fidelidade à onda sonora original e o tempo de processamento:

- Taxa de amostragem: Expressa em kHz, informa quantas amostras da informação de áudio serão extraídas a cada segundo.
- Resolução de amostragem: Expressa em bits/amostra. Informa o tamanho de cada amostra em bits.
- Tamanho do buffer interno: Expresso em número de amostras, informa quantas amostras estão no buffer interno com informações adquiridas ou recebidas.
- Número de *Buffers*: Inteiro que informa quantos *buffers* internos o *driver* acumulará antes de transmitir as informações adquiridas.

O ajuste desses parâmetros permite fazer um compromisso entre o tempo de processamento e a qualidade do áudio. Por exemplo, o uso de um tamanho de buffer interno menor e uma taxa de amostragem maior reduz a latência, mas aumenta o uso da CPU, dependendo das capacidades da CPU e das especificações superiores da interface de áudio para garantir a qualidade do áudio. A latência de processamento pode ser calculada da seguinte forma (ROTTONDI *et al.*, 2016):

$$\text{Latência}_{\text{interface de áudio}} = \frac{\text{Tamanho do buffer interno} * \text{Número de } \textit{Buffers}}{\text{Taxa de amostragem}} \quad (2.1)$$

Este cálculo não é observado na vida real devido a latências adicionais relacionadas ao funcionamento do *driver* e do sistema operacional.

2.4.2 Drivers

Sob a perspectiva de uma aplicação, é interessante ter ciência dos diferentes *drivers* presentes em cada sistema operacional. Normalmente, os *drivers* mais utilizados para aplicações musicais são:

- ASIO (Audio Stream Input/Output) no Windows
- CoreAudio no OSX
- ALSA (Advanced Linux Sound Architecture) para dispositivos Linux

Uma abordagem interessante em desenvolvimento é a RASPA, um *driver* desenvolvido para sistemas de áudio em tempo real. Foi desenvolvido pela Elk Audio OS (OS, 2021)

voltado para o seu sistema operacional Audio OS, que possui código aberto. Neste trabalho, o foco será no *driver* ALSA, pois será utilizado principalmente dispositivos Linux. O ALSA oferece suporte eficiente para todos os tipos de interfaces de áudio, desde placas de som de consumo até interfaces de áudio multicanal profissionais. Ele permite configurar a taxa de amostragem, o tamanho do *buffer* e o tamanho do *buffer* do *driver* para evitar sobrecargas e sub-execuções.

2.4.3 Processamento de Áudio

Um *codec* é um dispositivo ou programa de computador que codifica ou decodifica um fluxo de dados digital ou sinal. Para reduzir o tamanho dos dados utilizados, as aplicações podem optar por aplicá-lo. Codificadores de áudio geralmente usam codificação de sub-banda, uma vez que isso permite uma inclusão direta de modelos psicoacústicos. Além disso, para alcançar melhores taxas de compressão, o uso de sub-bandas geralmente é necessário, embora aumente o atraso algorítmico.

Nesse escopo, foi desenvolvido um *codec* de música de baixo atraso chamado *Opus Codec*, que pode oferecer grandes ganhos ao ser adaptado para aplicações interativas na internet. O software é gratuito e oferece suporte para voz e música, operando em uma ampla gama de taxas de bits variáveis, integração com o *Real-Time Protocol* (RTP), e boa ocultação de perda de pacotes, com um baixo atraso algorítmico (VALIN *et al.*, 2016). Ele escala para atrasos tão baixos quanto 5ms. Além disso, quatro diferentes estratégias psicoacústicas ocorrem para prevenir vazamentos, oferecer resolução variável de tempo-frequência, reduzir ruído tonal e prevenir colapsos em transientes em baixa gama de bits.

Algumas soluções optam pelo uso de áudio não comprimido para evitar o atraso adicionado pelos *codecs* de áudio. Entretanto, conforme os limites de largura de banda crescem, o uso de *codecs* de áudio não parece conveniente, uma vez que aumenta o atraso e reduz a qualidade do áudio.

2.4.4 Sincronização da stream de áudio

Na perspectiva de uma solução em execução, pode-se enfrentar alguns problemas maiores que podem influenciar a qualidade da experiência, como: sincronização de fluxo de dados e perda de pacotes. Para superar alguns desses problemas, várias estratégias podem ser implementadas.

A sincronização de fluxo de dados é um grande problema e aumenta conforme o número de participantes. Pequenas diferenças de inclinação de *clock* (*clock skew*) podem se acumular e aumentar a latência e, como é preciso lidar com interfaces de áudio externas, há exposição a possíveis variações. A inclinação do *clock* é a proporção entre a velocidade do *clock* de um dispositivo e a velocidade do *clock* de um relógio de referência. Para alcançar uma sincronização precisa, deve-se associar a cada fluxo de dados um carimbo de tempo e todos os *clocks* precisam ser sincronizados a um *clock* de referência. Como afirmado em (ROTTONDI *et al.*, 2016), algumas estratégias adaptadas para WAN podem ser utilizadas, como o uso de *Global Positioning Satellite* (GPS), usando *Global Navigation Satellite System* (GNSS) para gerar *clock words* para sincronizar interfaces de áudio com relógios externos configuráveis. De todo modo, segundo (CARÔT; WERNER, 2009), há uma proposta de solução por meio ajuste dinâmico do *clock* dos servos em referência a um mestre, baseado em função de *callback*, o que pode ser interessante para aplicações de tempo real.

2.4.5 Protocolos de Rede

Após a aquisição e o processamento do áudio, é necessário enviá-lo por meio da rede. Este processo inclui o empacotamento (*packaging*) do áudio processado, o envio dos pacotes por meio da rede e a desempacotamento (*depaketization*) dos pacotes (*packages*) recebidos. O empacotamento é o processo de agrupar dados em pacotes de acordo com um protocolo específico. Um pacote consiste em informações de controle e dados do usuário ou carga útil. As informações de controle fornecem todas as informações necessárias para transferir informações por meio da rede, como: rede de origem e destino, códigos de detecção de erro, etc. A carga útil tem as informações enviadas pelo usuário, como as informações de áudio. O desempacotamento é o processo de extração de informações de pacotes de rede. Para criar e transmitir um pacote, deve-se seguir um protocolo de transmissão específico.

O protocolo mais usado por aplicações de NMP é o *User Datagram Protocol* (UDP) (675, 1980). O protocolo UDP fornece um procedimento para programas de aplicação enviarem mensagens a outros programas com um mínimo de mecanismo de protocolo. Ele não garante a entrega nem a entrega ordenada de fluxos, pois depende da Qualidade de Serviço (*Quality of Service*, QoS) da rede. Outro protocolo amplamente usado, mas não comumente usado em aplicações de NMP, é o *Transmission Control Protocol* (TCP) (675, 1974). O TCP é o protocolo projetado para ser usado como um protocolo *host-to-host* altamente confiável, ele precisa de

uma boa quantidade de informações de controle em cada pacote para garantir que os pacotes são enviados, recebidos e retransmitidos quando perdidos, para controlar o congestionamento do tráfego de rede, e para habilitar outros controles. O UDP é um protocolo de rede mais adequado para *NMP*, já que introduz menos sobrecarga de transmissão devido ao menor tamanho do cabeçalho do pacote em comparação com o TCP, sendo, portanto, mais eficiente para aplicações em tempo real.

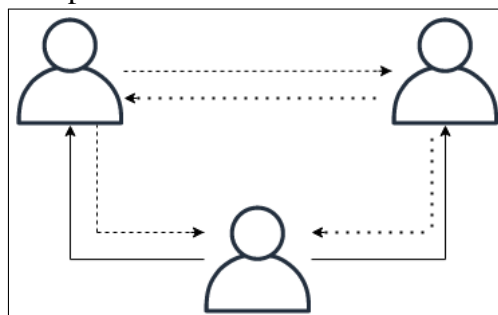
Para tornar as aplicações mais robustas, e como o UDP oferece pouco controle sobre a transmissão de pacotes, algumas estratégias podem ser implementadas para superar as desvantagens. Por exemplo, a estratégia de correção de erros para a frente, onde cada pacote enviado tem informações sobrepostas de pacotes anteriores para restaurar informações perdidas. Técnicas de ocultação de erro também podem ser implementadas para substituir pacotes perdidos, como o uso de silêncio para preenchimento ou ruído branco, repetição do último pacote recebido, e a correspondência de padrão usando pequenos segmentos de amostras antes e depois de pacotes perdidos, dentre outras técnicas (WAH *et al.*, 2000).

2.4.6 Arquitetura de Rede

Aplicações de performance musical em rede podem usar diferentes arquiteturas de rede para transmitir informações de um usuário para outros, como: ponto-a-ponto (*peer-to-peer*), cliente-servidor, ponto-a-ponto em *multicast* e cliente-servidor em *multicast*.

- Ponto-a-Ponto (P2P) (figura 5): Cada participante envia seus dados para todos os outros participantes. A arquitetura P2P não é escalável, o que significa que não é bem adaptada para um grande número de usuários conectados, principalmente porque precisa enviar uma cópia de seus dados para cada usuário conectado, o que faz com que a largura de banda necessária possa ultrapassar a largura de banda disponível.

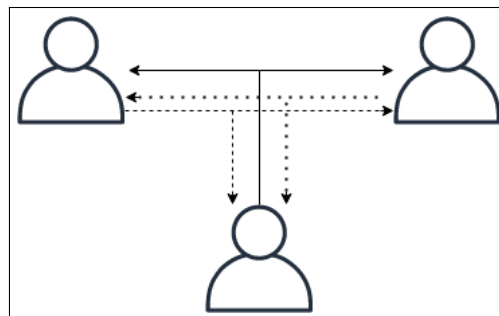
Figura 5 – Arquitetura ponto-a-ponto



Fonte: (ROTTONDI *et al.*, 2016)

- Cliente-Servidor (figura 6): Cada participante envia seus dados para um servidor e o servidor envia os dados recebidos para os outros participantes conectados. Em aplicações de *NMP*, um servidor pode misturar informações de áudio antes de transmiti-las. Esta arquitetura permite escalabilidade, mas força o desenvolvimento de um servidor com grande disponibilidade de largura de banda e de recursos computacional, o que pode ser uma oportunidade interessante para grandes empresas, com grandes recursos, que pode oferecê-lo como um serviço. Se a arquitetura cliente-servidor for implementada, os servidores devem ser instalados em regiões geográficas estratégicas, em aplicações de *NMP*, grandes distâncias entre usuários e servidores têm um impacto negativo no desempenho do sistema devido ao aumento da latência.

Figura 6 – Arquitetura cliente-servidor

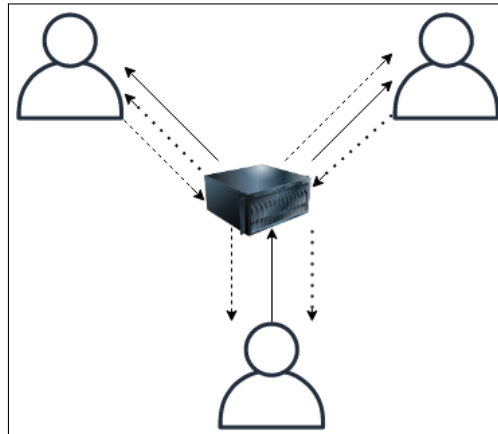


Fonte: (ROTTONDI *et al.*, 2016)

Enquanto as arquiteturas usuais de P2P e cliente-servidor usam uma abordagem de transmissão *unicast*, onde os dados são transmitidos de um ponto na rede para outro ponto, o *multicast* permite transmitir de um ponto na rede para múltiplos destinos. *Multicast* é uma comunicação em grupo onde a transmissão de dados é endereçada a um grupo de destinos simultaneamente (STAIS *et al.*, 2013),(SAVOLA, 2008).

- Ponto-a-Ponto Multicast (P2P Multicast) (figura 7): Cada participante envia uma vez seus dados e eles são enviados para todos os outros usuários simultaneamente. Para arquiteturas ponto-a-ponto, o uso de *multicast* resolve os problemas de escalabilidade, uma vez que os roteadores são responsáveis por duplicar os dados enviados quando necessário. Abordagens de *multicast* descentralizado (P2P *multicast*) mostram latências menores do que estratégias de *multicast* centralizado e *unicast* centralizado.

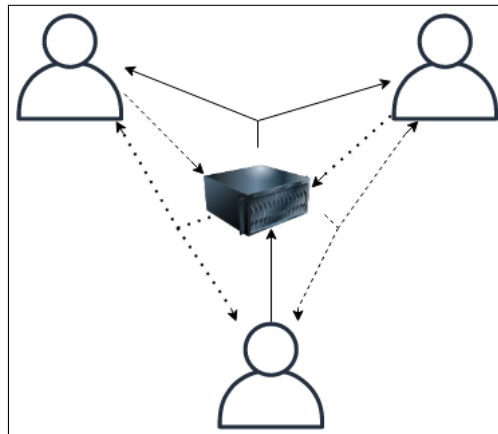
Figura 7 – Arquitetura ponto-a-ponto *multicast*



Fonte: (ROTTONDI *et al.*, 2016)

- cliente-servidor *Multicast* (figura 8): Cada participante envia seus dados para um servidor e o servidor envia uma vez a informação para todos os *outros* usuários simultaneamente. Para arquiteturas cliente-servidor, o uso de multicast pode ser útil para reduzir as exigências sobre poder computacional e largura de banda, já que os servidores só precisam enviar um fluxo de dados.

Figura 8 – Arquitetura cliente-servidor *multicast*



Fonte: (ROTTONDI *et al.*, 2016)

Uma desvantagem do *multicast* é que os roteadores em uma transmissão de pacotes devem ser capazes de executar transmissão *multicast* nos nós em seu caminho.

2.4.7 Jitter de Rede

A variação da rede, ou *jitter*, é a variação no atraso entre o momento em que os pacotes são transmitidos e o momento em que são recebidos em uma conexão de rede. O *jitter* de rede pode ser causado por várias razões, tais como:

- Congestionamento de rede: quando não há largura de banda suficiente disponível em um equipamento específico da rede.
- *Jitter* sem fio: equipamentos sem fio podem aumentar o *jitter* da rede em comparação com conexões com cabos.
- Desempenho de hardware: roteadores, cabos e *switches* de baixa qualidade podem aumentar o *jitter*.

Esta variação poderia aumentar o número de informações perdidas em aplicações em tempo real se estratégias para preveni-lo não forem implementadas. Como em aplicações em tempo real usam um valor fixo de latência, se o *jitter* for muito grande, um pacote pode chegar depois do tempo esperado. Isso faz com que ele não seja utilizado, resultando no mesmo impacto de um pacote perdido, ou seja, uma interrupção no som resultante.

2.4.8 Gerenciamento da aplicação

Para gerenciar o fluxo de informações de todos os usuários e superar o *jitter* de rede, as aplicações são projetadas com *buffers* de *jitter*. Cada sessão de Performance Musical em Rede (NMP) tem um tamanho de *buffer*, que é o tamanho do *buffer* usado por quem está fazendo a performance. O *buffer* de *jitter* tem o tamanho de um múltiplo do tamanho do *buffer*. Dependendo da conexão de rede, o *jitter* de rede pode ser grande ou pequeno, o *buffer* de *jitter* é usado para compensá-lo. O som só é reproduzido em cada lado do usuário quando o *buffer* de *jitter* está suficientemente preenchido.

Usar um *buffer* de *jitter* é interessante porque, quando um pacote chega mais tarde devido ao *jitter* de rede, ele possivelmente ainda será usado. Isso porque ele poderá chegar até o tempo limite do *buffer*.

2.4.9 Considerações Finais

No decorrer deste capítulo, exploramos os conceitos fundamentais da performance musical via rede e suas diversas aplicações. Compreendemos a relevância da latência na comunicação em rede e estabelecemos faixas de latência que permitem a realização de performances musicais remotas. Além disso, delimitamos o escopo técnico de um sistema de performance musical via rede e examinamos como cada componente contribui para a latência geral do sistema. Investimos tempo no estudo detalhado da cadeia de aquisição de áudio, abrangendo desde as interfaces de áudio até as configurações de rede mais adequadas para a comunicação entre os

usuários do sistema.

Baseando-nos nos fundamentos teóricos e nos conhecimentos adquiridos, avançamos para a próxima etapa crucial deste trabalho: a implementação e análise do sistema de performance musical via rede. Nesta seção, apresentaremos uma descrição minuciosa da metodologia adotada para alcançar nossos objetivos de implementação e avaliação do sistema.

3 METODOLOGIA

Este trabalho tem por objetivo a implementação e a análise objetiva e subjetiva de um sistema de performance musical via rede. Ou seja, há quatro linhas de estudo a serem seguidas:

- Análise das soluções NMP existentes e definição de uma solução a ser implementada: Iniciamos com uma análise detalhada das soluções NMP disponíveis, considerando seus recursos, vantagens e limitações. Com base nessa análise, selecionamos uma solução específica para implementação no nosso estudo.
- Implementação de um sistema NMP utilizando o software escolhido: Detalhamos o processo de implementação do sistema NMP selecionado, utilizando o software escolhido como base. Explicamos as etapas de configuração, integração e personalização necessárias para adaptar a solução às nossas necessidades específicas.
- Análise Objetiva do sistema NMP implementado: Apresentamos uma análise objetiva do desempenho do sistema NMP implementado. Isso inclui métricas quantitativas e resultados de testes realizados para avaliar a eficiência, precisão e confiabilidade do sistema em diferentes cenários.
- Análise Subjetiva do sistema NMP implementado: Complementando a análise objetiva, realizamos uma análise subjetiva do sistema NMP implementado. Isso envolve a coleta de opiniões e *feedback* de usuários e especialistas que interagiram com o sistema. Exploramos aspectos como usabilidade, experiência do usuário e satisfação geral com a solução.

A partir dessas linhas de estudo, os objetivos do trabalho poderão ser verificados e os resultados demonstrados.

3.1 Definição da solução NMP existente a ser utilizada

No decorrer dos últimos anos, várias aplicações de software foram desenvolvidas para permitir performances musicais em rede, além de outras que ainda estão em desenvolvimento.

A fim de melhor entender cada uma das soluções, será dada prioridade para os seguintes recursos:

- Tipo de arquitetura de rede
- Interface do usuário e design de experiência do usuário
- Uso de *codecs* de áudio
- Licença de software e plataformas suportadas

- Estratégias de *Jitter buffer*

O estudo dessas características permitirá entender como as aplicações funcionam e suas vantagens e desvantagens. Em seguida, serão exploradas algumas características especiais de cada aplicação apresentada.

3.2 Implementando um Sistema NMP

Após a análise das soluções existentes e escolha de uma solução para a implementação de um sistema NMP, executamos a implementação de um sistema de performance musical via rede. Esse sistema consiste não somente na escolha do software utilizado, mas também na escolha e configuração de todos os sub-itens presentes em um sistema:

- Computador utilizado pelo cliente
- Computador utilizado pelo servidor
- Interface de áudio
- Aplicação de performance via Rede
- Conexão de rede
- Configuração dos sistemas

A identificação de cada item é fundamental para o entendimento dos resultados obtidos.

3.3 Análise Objetiva

A análise objetiva de um sistema de performance musical via rede reside no estudo da possibilidade de haver uma colaboração musical síncrona utilizando esse sistema.

Como visto no capítulo da fundamentação teórica, a variável que nos informa se é possível ou não haver uma colaboração síncrona é a latência. Uma latência de *round-trip* de *50ms* é considerada o valor ótimo para uma performance confortável.

Além disso, a qualidade do áudio escutado na performance é outro elemento importante para nossa análise, como já visto, a queda de qualidade de áudios é devida a perda de pacotes de rede.

Para analisar um sistema então, é necessário analisar a latência global do sistema e a quantidade de pacotes de rede perdidos no decorrer da transmissão.

3.4 Análise Subjetiva

A análise subjetiva de um sistema de performance musical via rede reside na investigação de como os músicos se sentiram praticando com aquele sistema. Para analisar isso, podem ser feitas questões individuais sobre a percepção da latência do sistema e como ela influenciou a performance.

4 EXPERIMENTOS E RESULTADOS

Neste capítulo, apresentamos os resultados e experimentos realizados com base na metodologia previamente descrita. O objetivo principal é analisar e avaliar as soluções NMP existentes, além de implementar e investigar um sistema NMP específico. No desenvolvimento do capítulo serão abordados os seguintes tópicos:

- Análise das soluções NMP existentes e definição de uma solução a ser implementada
- Implementação de um sistema NMP utilizando o software escolhido
- Análise Objetiva do sistema NMP implementado
- Análise Subjetiva do sistema NMP implementado

Ao explorar esses tópicos, buscamos fornecer uma visão abrangente dos resultados obtidos e das conclusões alcançadas com relação à solução NMP implementada. Essas informações são essenciais para avaliar a eficácia do sistema e identificar possíveis melhorias ou ajustes necessários.

4.1 Avaliação de soluções NMP existentes

Dentre as diversas soluções desenvolvidas nos últimos anos, as seguintes soluções foram estudadas: JackTrip (LABS, 2023), Jamulus (FISCHER, 2023) e Sonobus (SONOSAURUS, 2023). Para cada um dos sistemas foram levantadas as principais características do sistema, funcionamento e detalhes sobre implementação.

4.1.1 *JackTrip*

O JackTrip é um software de performance de áudio de alta qualidade na internet. Seu desenvolvimento começou no início dos anos 2000 e continua sendo aprimorado. Foi desenvolvido pelo Centro de Pesquisa em Computação Musical da Universidade de Stanford (CCRMA).

4.1.1.1 *Como usar ?*

Para instalar o JackTrip, uma orientação de instalação é oferecida no site da (CCRMA, 2023). Um guia para instalar o Jack Audio Connection Kit, visto na figura 9 (também conhecido como JACK), que funciona como servidor de som e *driver* de placa de som, também está

disponível. A título de exemplo, um servidor de som é um software que gerencia o trânsito de áudio entre dispositivos e aplicações.

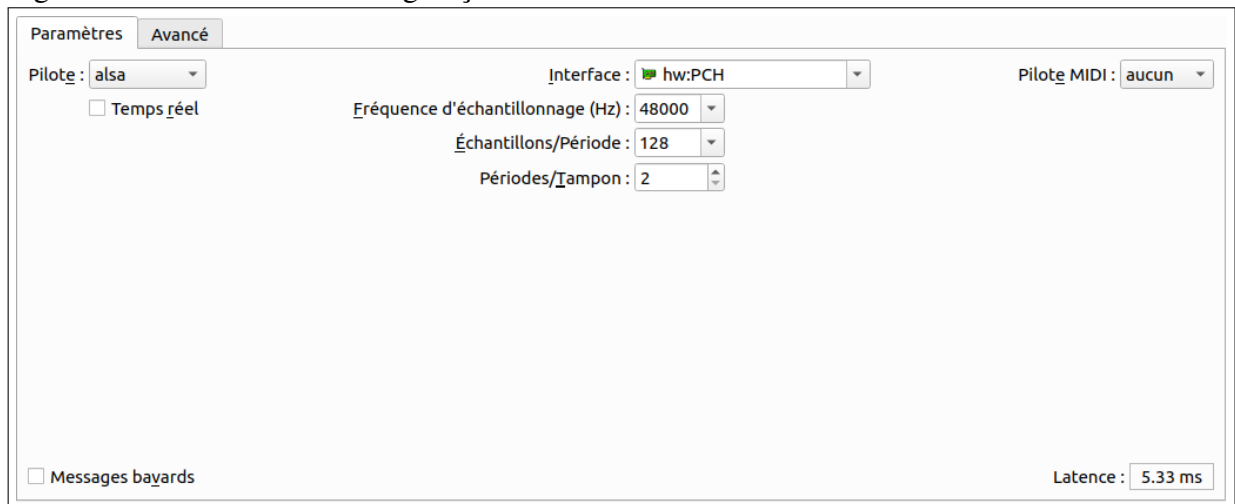
Antes de usar o JackTrip, o JACK deve ser configurado para acessar o áudio da placa de som. Na interface do usuário do JACK, pode-se configurar o tamanho do *buffer*, a taxa de amostragem, a placa de som e o número de *buffers* na reprodução (10).

Figura 9 – Interfaces principal do *Jack Audio Connection Kit*.



Fonte: elaborada pelo autor. (2021)

Figura 10 – Interfaces de configuração do *Jack Audio Connection Kit*.



Fonte: elaborada pelo autor. (2021)

Depois disso, o comando `jacktrip -C server_ip_address` é usado para se conectar a um servidor (figura 11). Para configurar um servidor, o mesmo processo de instalação deve ser seguido substituindo um `-S` em vez de `-C`. Como o JACK também funciona como um servidor de som (figura 12), ele oferece uma interface para gerenciar as informações de áudio sendo enviadas ou recebidas.

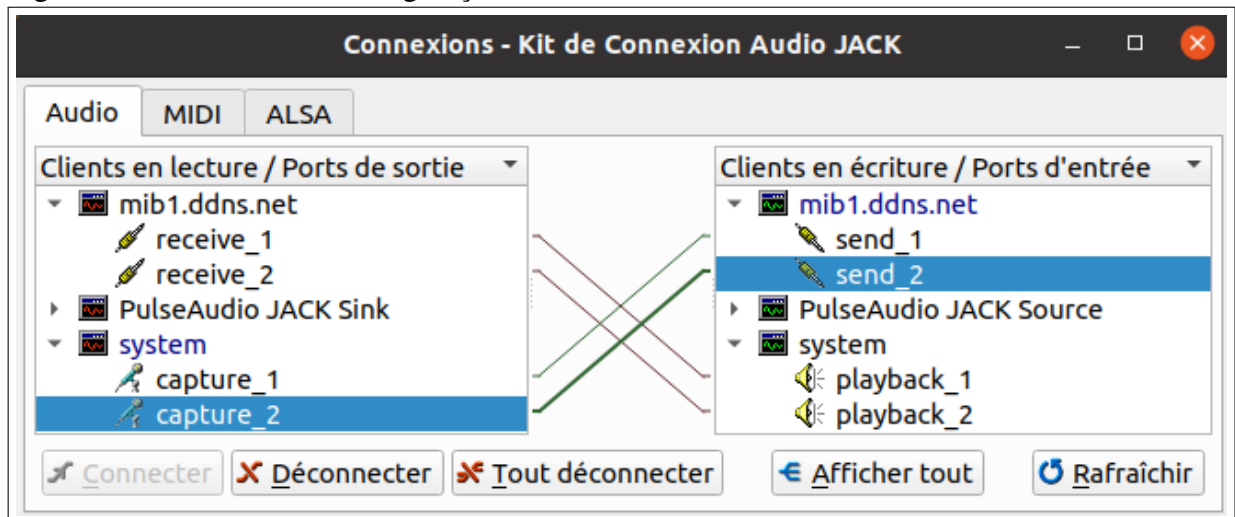
Figura 11 – Execução do *Jacktrip*.

```
gmdn0286@yd-5CG7462B57:~$ jacktrip -C mib1.ddns.net
Setting JACK Process Callback...
SUCCESS

-----
The Sampling Rate is: 48000
-----
The Audio Buffer Size is: 128 samples
                        or: 512 bytes
-----
The Number of Channels is: 2
-----
Using UDP Protocol
-----
Using JitterBuffer strategy 1
Server Address set to: mib1.ddns.net Port: 61002
-----
UDP Socket Receiving in Port: 4464
-----
Waiting for Peer...
WARNING: Peer Buffer Size is : 256
        Local Buffer Size is : 128
Received Connection from Peer!
```

Fonte: elaborada pelo autor. (2021)

Figura 12 – Interfaces de configuração do *Jack Audio Connection Kit*.



Fonte: elaborada pelo autor. (2021)

Para obter valores de atraso ótimos, alguns testes devem ser realizados, pois JACK e JackTrip têm uma configuração ideal para cada computador, placa de som e conexão de rede disponível.

4.1.1.2 *Arquitetura de rede*

O Jacktrip utiliza o UDP para enviar áudio não comprimido pela rede. Ele pode ser configurado tanto em arquiteturas de rede cliente-servidor quanto ponto a ponto (*peer-to-peer*). Utiliza-se da porta UDP número 4464 como padrão, mas os usuários podem ter que habilitar suas portas para usá-lo. No modo cliente-servidor, ele usa a porta número 4464 para receber um *handshake* de TCP de cada usuário e associá-los a uma porta entre 61002 a 62000, responsável por envio e recepção.

4.1.1.3 *Interfaces e Design*

O Jacktrip é executado na linha de comando do linux, não possuindo uma Interface de Usuário.

4.1.1.4 *Codec de Áudio*

O Jacktrip não usa nenhum Codec, pois envia áudio não comprimido.

4.1.1.5 *Licenças e plataformas suportadas*

O Jacktrip é um software de código aberto, tendo seu código-fonte disponível no GitHub. O grupo de pesquisa SoundWire do CCRMA e a fundação JackTrip ainda trabalham no desenvolvimento do JackTrip. Ele está disponível para Linux, OsX e Windows.

4.1.1.6 *Estratégias para lidar com Jitter e Perda de pacotes*

Para evitar falhas no áudio devido ao *jitter*, o Jacktrip propõe opcionalmente uma redundância de pacotes na extremidade de recepção. Esse processo consiste em enviar o mesmo pacote várias vezes para restaurar dados perdidos (CÁCERES; CHAFE, 2010).

4.1.1.7 *Características adicionais e limitações*

O Jacktrip possui diversos recursos, a exemplo de:

- Configuração do tamanho do *buffer* de *jitter*: Para lidar com o *jitter*, é necessária a implementação de um *buffer*, evitando falhas devido a pacotes que não chegam no tempo de atraso correto. Ele pode ser configurado na inicialização do *Jacktrip*, por exemplo:

`jacktrip -C server_ip_address -q 4`, o que inicia uma sessão usando 4 *buffers* de *jitter*.

- Ferramenta de estudo de atraso de áudio: Um áudio de teste é enviado para o servidor. O número de amostras até o retorno é medido. Este processo é repetido continuamente. A média cumulativa e o desvio padrão são calculados para os atrasos medidos. Uma mensagem é levantada quando ocorre perda de pacotes, embora nenhuma ferramenta seja fornecida para estudar as perdas de pacotes.

O JackTrip tem algumas limitações, tais como:

- É necessário saber como operar o Jack e o Jacktrip para usá-lo, ele não possui uma interface amigável ao usuário;
- Não oferece uma boa interface de instalação, além de requerer a instalação do Jack;
- Não lida com vídeo;
- A qualidade do áudio e a latência dependem da qualidade da internet;
- Sua operação necessita ser realizada por linha de comando;
- Todo o hardware e software conectado deve ter a mesma taxa de amostragem e tamanho de buffer;
- Não possui ferramentas para ajustar a deriva do *clock*;
- A configuração usada é estática. A fim de mudar os parâmetros de configuração, a conexão e as aplicações devem ser reiniciadas.

Estes são, portanto, alguns dos recursos do JackTrip, bem como algumas de suas limitações.

4.1.2 Jamulus

Jamulus é um software de código aberto que permite aos músicos se apresentar, ensaiar ou apenas tocar, tudo em tempo real, pela Internet.

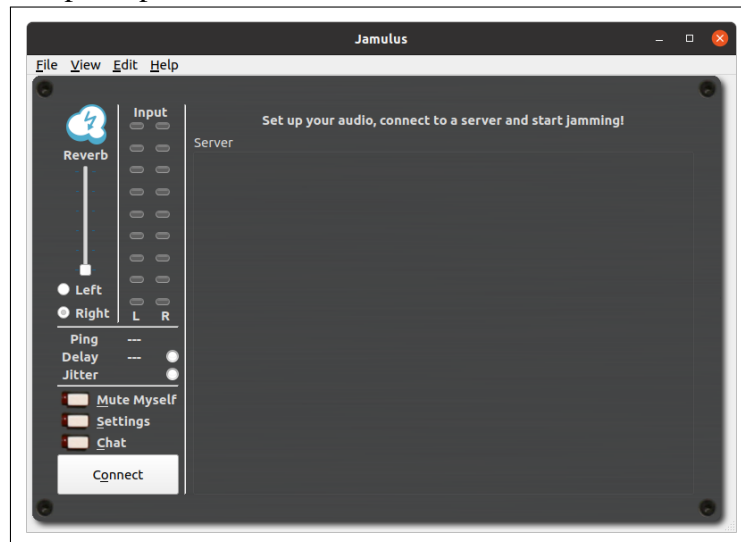
4.1.2.1 Como usar ?

Jamulus (13) precisa de um servidor de som. O Jamulus usa o Jack Audio Connection Kit (JACK) para Linux, Audio Stream Input/Output (ASIO) para Windows e Core Audio para OSX. Uma orientação sobre como instalar o aplicativo é fornecida no site do (JAMULUS, 2021).

Após configurar o servidor e *driver* de som, é possível iniciar o aplicativo Jamulus. Este permitirá a conexão a um servidor usando a janela (figura 14). Jamulus tem uma grande

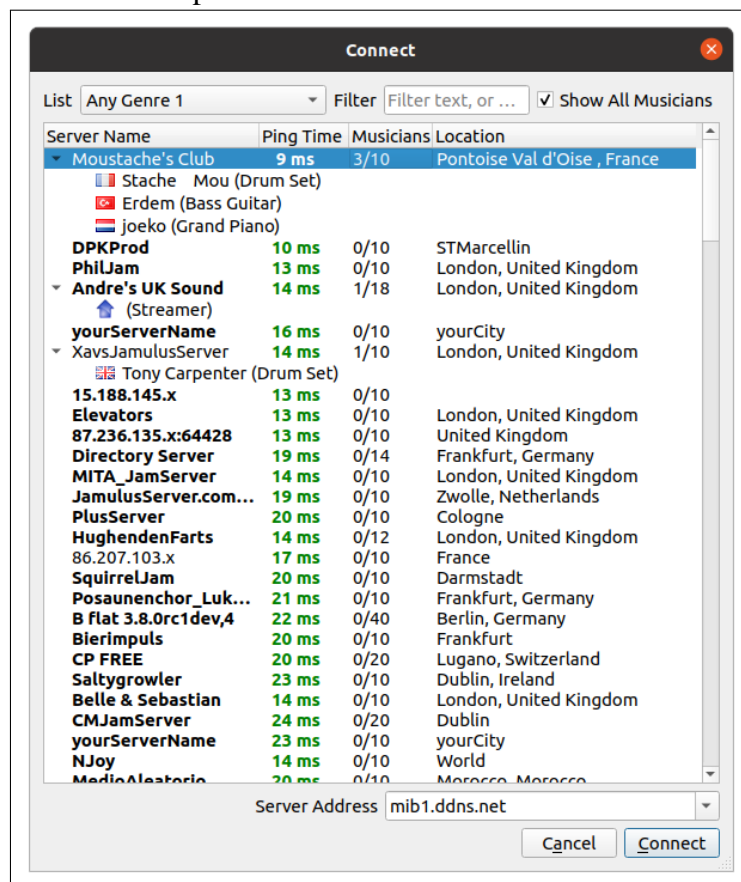
comunidade e há muitos servidores públicos disponíveis. Também é possível criar um servidor privado.

Figura 13 – Interface principal do *Jamulus*.



Fonte: elaborada pelo autor. (2021)

Figura 14 – Lista de servidores públicos.

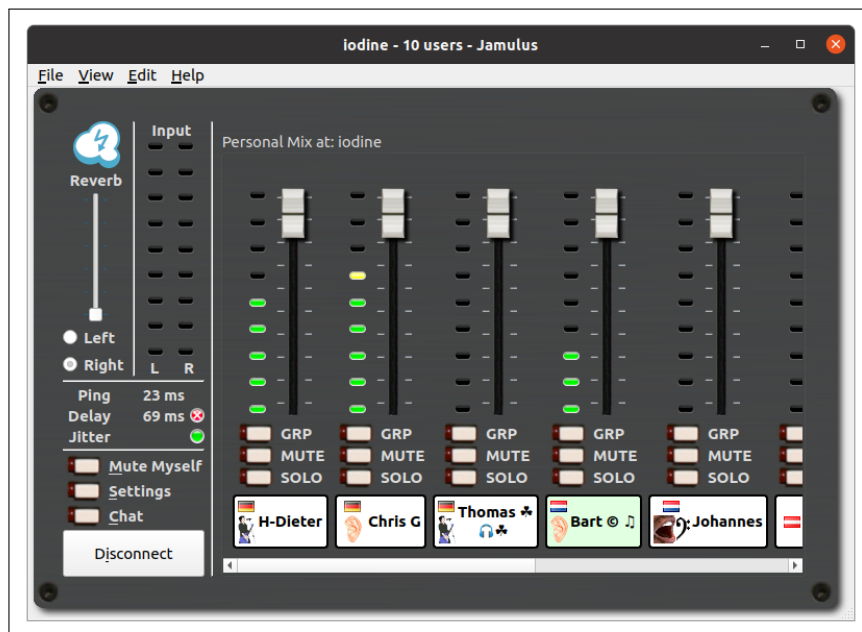


Fonte: elaborada pelo autor. (2021)

Quando conectado a um servidor, uma sessão é iniciada e é possível praticar junto com os outros participantes. Tanto os valores de latência quanto de *ping* são estimados e fornecidos pelo software. Em seguida, seus resultados são mostrados na janela da sessão. A interface do usuário oferece um *mixer* de volume para cada participante (figura 15). Um chat está disponível para se comunicar usando texto.

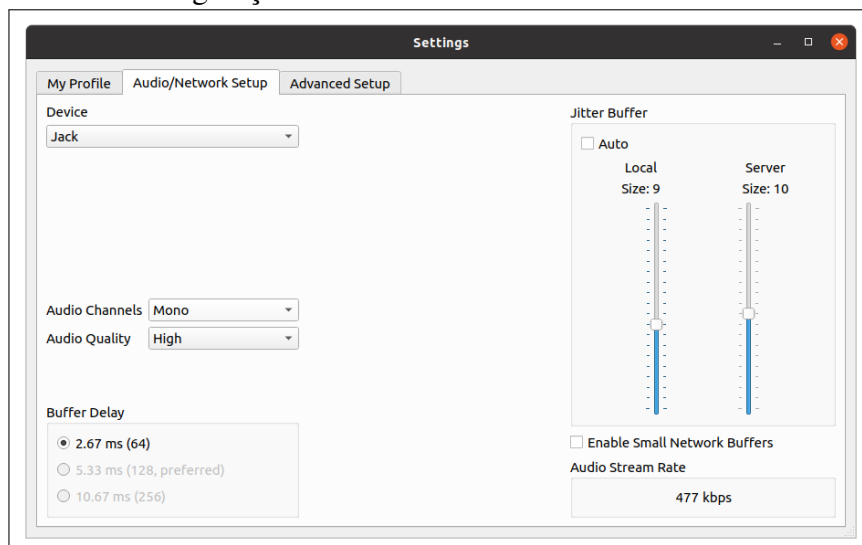
Nas configurações do Jamulus, é possível ajustar o tamanho do buffer de *jitter* e a qualidade do áudio (figura 16), já que o Jamulus usa um Codec.

Figura 15 – Janela de sessões do *Jamulus*.



Fonte: elaborada pelo autor. (2021)

Figura 16 – Janela de configuração do *Jamulus*.



Fonte: elaborada pelo autor. (2021)

4.1.2.2 *Arquitetura de rede*

O Jamulus usa uma arquitetura de rede cliente-servidor. O servidor usa a porta UDP 22124 para enviar e receber dados.

4.1.2.3 *Interfaces e Design*

O Jamulus é um aplicativo e tem uma interface de usuário que permite aos usuários configurar suas sessões, identificar-se usando seu nome, o seu país e o instrumento tocado por ele.

4.1.2.4 *Codec de áudio*

O Jamulus usa o Codec Opus, um codec de música de baixo atraso.

4.1.2.5 *Licenças e plataformas suportadas*

O Jamulus é um software de código aberto (licença GPL-2.0) e seu código está disponível no GitHub. Ele tem suporte para Linux, OSX e Windows.

4.1.2.6 *Estratégias para lidar com Jitter e Perda de pacotes*

O Jamulus possui um *buffer* de *jitter* para compensar os *jitters* de tempo da placa de som e da rede. Oferece uma opção para calibrar automaticamente o tamanho do buffer de *jitter*. Ele não implementa nenhum método para recuperar ou compensar pacotes perdidos.

4.1.2.7 *Características adicionais e limitações*

Algumas limitações do Jamulus são:

- O uso de um Codec, o que reduz a qualidade do som e adiciona atraso ao sistema, mesmo sendo um Codec de baixo atraso.
- Sessões privadas estão disponíveis apenas se um servidor for configurado.

4.1.3 *Sonobus*

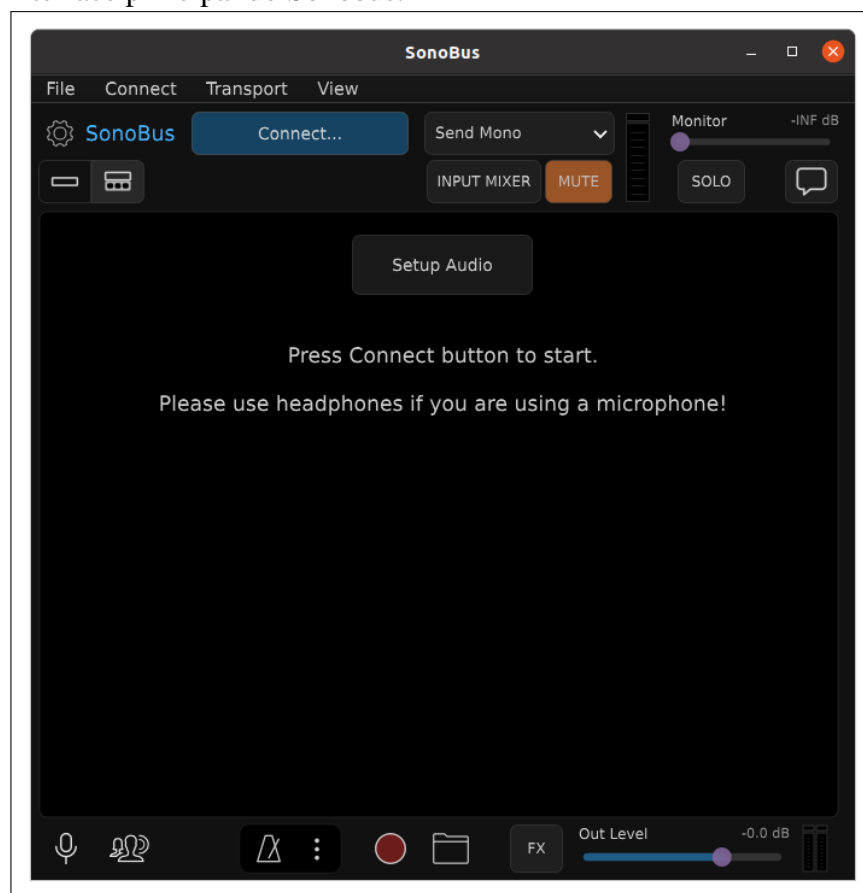
SonoBus é uma solução peer-to-peer elegante e de código aberto para NMPs que é compatível com várias plataformas.

4.1.3.1 Como usar ?

Para usar o SonoBus, é necessário instalar o aplicativo disponível no site. Como *driver* de placa de som, SonoBus usa JACK ou ALSA no Linux, ASIO no Windows e CoreAudio no OSX.

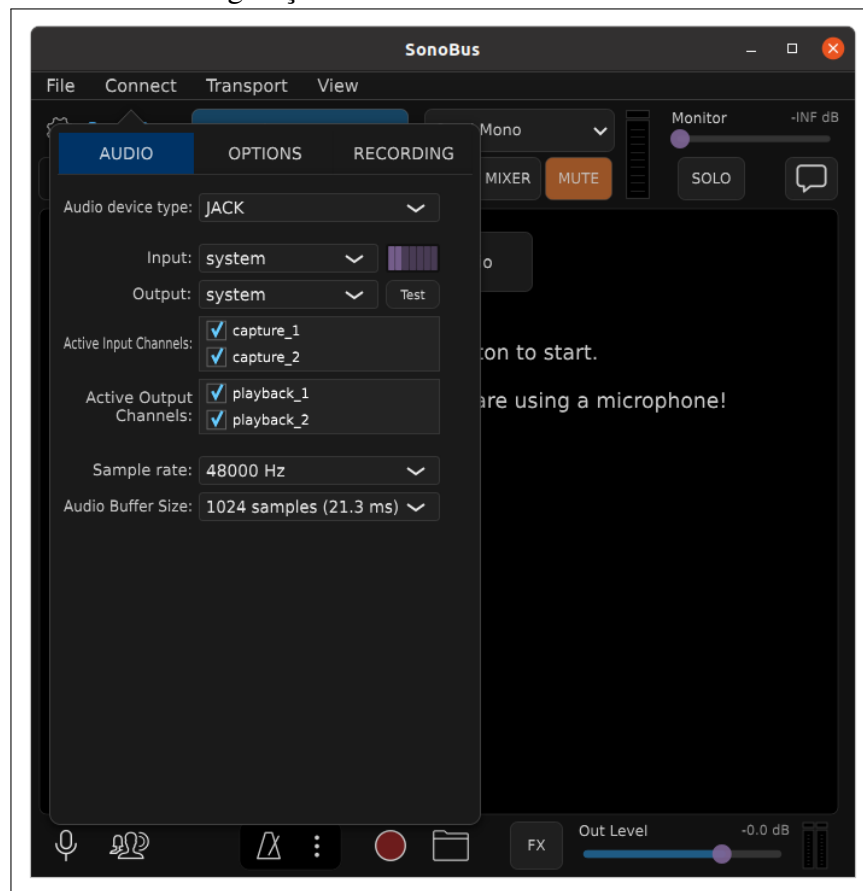
Após lançar o aplicativo, é possível configurar o *driver* de áudio e criar ou se conectar a uma sessão, conforme visto na figura 17. No menu do *driver* de áudio, existe uma interface para configurar os parâmetros da placa de som e escolher seu *driver*, como visto na figura 18.

Figura 17 – Interface principal do Sonobus.



Fonte: elaborada pelo autor. (2021)

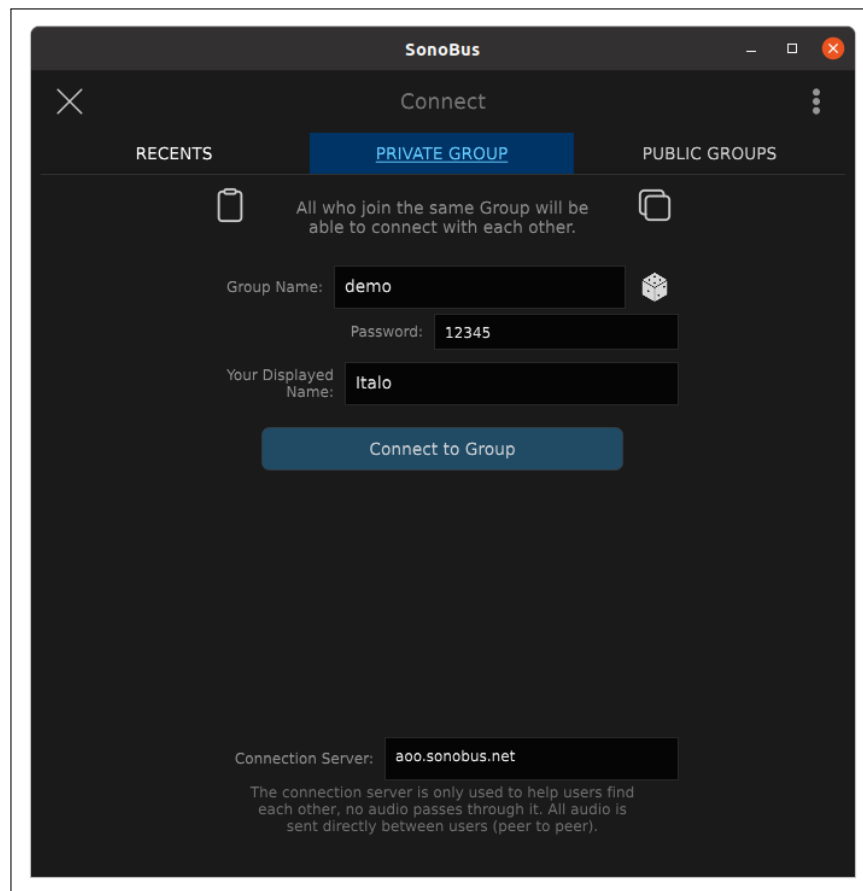
Figura 18 – Interface de configuração do Sonobus.



Fonte: elaborada pelo autor. (2021)

No menu de conexão, o usuário pode criar uma sessão privada ou pública e definir uma senha para a sessão. A informação é transmitida de maneira *peer-to-peer* entre os participantes (figura 19).

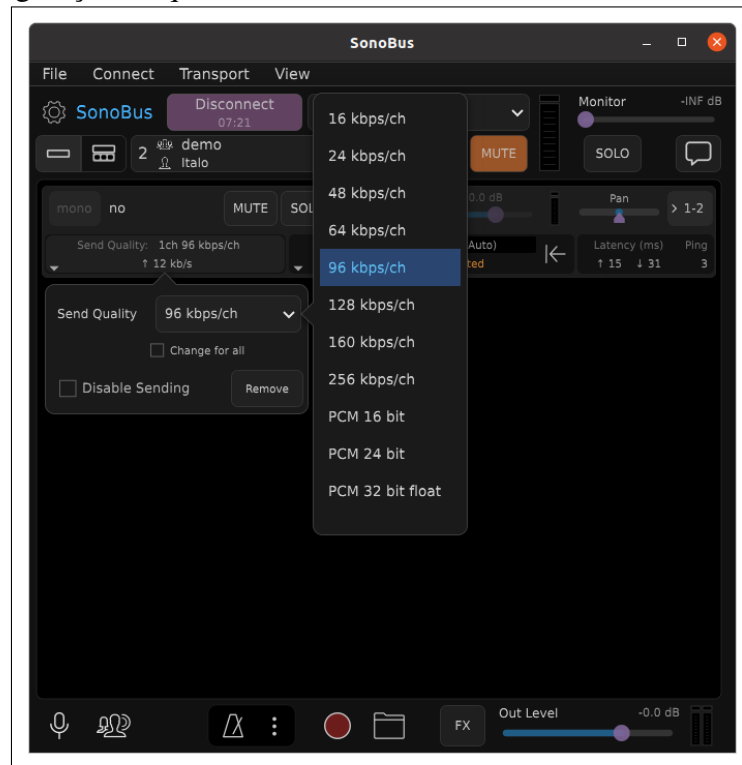
Figura 19 – Menu de conexão do SonoBus.



Fonte: elaborada pelo autor. (2021)

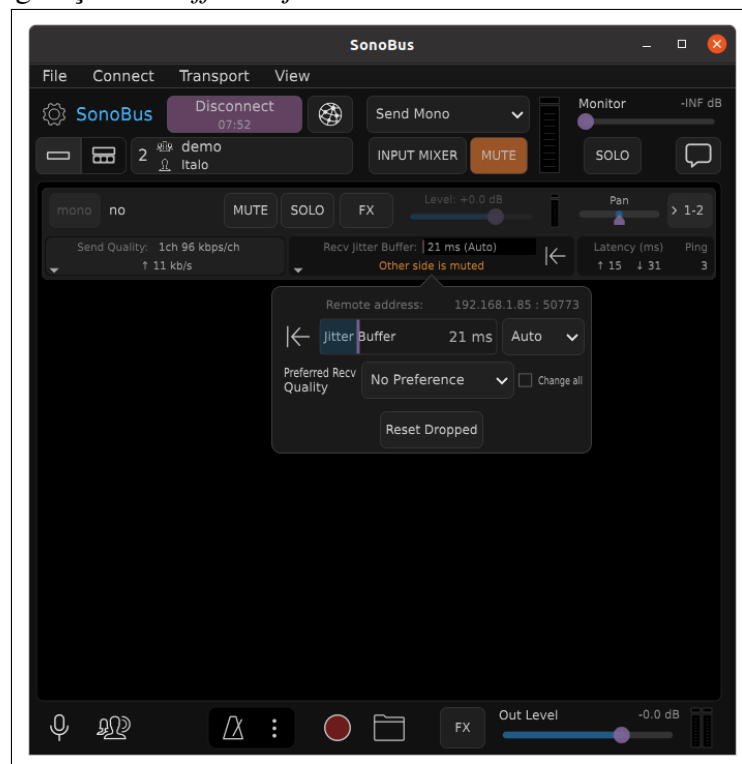
Na sessão, o usuário pode configurar a qualidade do áudio e o buffer de *jitter* (figuras 20 e 21). O SonoBus permite que o usuário escolha a qualidade do áudio enviada aos outros usuários, pois usa o *codec* Opus para comprimir o áudio. Há também a possibilidade de enviar áudio não comprimido. O *software* também oferece uma ferramenta automática para calibrar o tamanho do *buffer* de *jitter* local para cada participante de acordo com as capacidades da rede, que também pode ser calibrado manualmente.

Figura 20 – Configuração da qualidade de áudio do Sonobus.



Fonte: elaborada pelo autor. (2021)

Figura 21 – Configuração do *buffer de jitter* do Sonobus.



Fonte: elaborada pelo autor. (2021)

4.1.3.2 *Arquitetura de rede*

O SonoBus usa uma arquitetura de rede *peer-to-peer*. O SonoBus usa a porta UDP 12000 para enviar e receber valores.

4.1.3.3 *Interfaces e Design*

O SonoBus tem uma interface elegante e simples de usar. É fácil de usar e intuitivo.

4.1.3.4 *Codec de áudio*

O SonoBus permite escolher entre áudio comprimido e não comprimido, bem como a taxa de compressão. Ele usa o Codec Opus.

4.1.3.5 *Licenças e plataformas suportadas*

O SonoBus é um software de código aberto (Licença GPL-3.0) e seu código está disponível no GitHub. Ele tem suporte para Linux, OSX e Windows.

4.1.3.6 *Estratégias para lidar com Jitter e Perda de pacotes*

O SonoBus possui um buffer de jitter para compensar os *jitters* de tempo da placa de som e da rede. Oferece uma opção para calibrar automaticamente o tamanho do *buffer* de *jitter*. Entretanto, não implementa nenhum método para recuperar ou compensar pacotes perdidos.

4.1.3.7 *Características adicionais e limitações*

Dentre os recursos do SonoBus, destacam-se:

- Controle completo da qualidade do áudio transmitido e recebido em uma base por usuário
- Inclusão de efeitos de áudio embutidos opcionais (compressão, cancelamento de ruído, equalização) para otimizar tanto a qualidade quanto o nível do áudio que você transmite.
- Controle de latência em uma base por usuário, possível através do controle manual ou automático do "*buffer* de *jitter*"

Algumas limitações do SonoBus são:

- O SonoBus atualmente não usa nenhuma criptografia para a comunicação de dados.
- O *software* não lida com vídeo.

4.1.4 Comparação das soluções NMP existentes

JackTrip (LABS, 2023), Jamulus (FISCHER, 2023) e Sonobus (SONOSAURUS, 2023) são soluções bem desenvolvidas e que funcionam bem para performances via rede. A tabela 2 sumariza as principais características dos sistemas estudados.

Tabela 2 – Comparação entre *softwares* de performance musical remota

	Jacktrip	Jamulus	Sonobus
Licença	Licença GPL-3.0	Licença GPL-2.0	Licença GPL-3.0
Arquitetura de rede	Cliente-Servidor ou <i>Peer-To-Peer</i>	Cliente-Servidor	Cliente-Servidor ou <i>Peer-To-Peer</i>
Interface e Design	<i>Command Line</i> (CLI)	Interface fácil de utilizar	Interface fácil de utilizar
<i>Codec</i> de áudio	Sem <i>coded</i>	Opus <i>Codec</i>	Sem <i>coded</i> ou Opus <i>Codec</i>
Gestão de Jitter	Redundância de pacotes	<i>Buffer</i> de <i>jitter</i> automático	<i>Buffer</i> de <i>jitter</i> automático

Fonte: elaborada pelo autor. (2021)

Dentre as várias soluções analisadas, optamos pela implementação do sistema utilizando a solução Jacktrip. Essa escolha foi baseada em sua facilidade de uso por meio da linha de comando e sua simplicidade de configuração. Esses aspectos tornaram o Jacktrip uma opção ideal para o nosso estudo, permitindo uma implementação mais ágil e eficiente do sistema de performance musical via rede.

4.2 Implementando um NMP

Para implementar o sistema de performance musical via rede, foi necessária a utilização de computadores, interfaces de áudio, conexão de internet e uma aplicação.

Tendo em vista que o objetivo dessa implementação foi endereçado ao estudo objetivo e subjetivo das performances por meio do sistema, optou-se pelo software com maior possibilidade de personalização possível. Por isso foi escolhida a aplicação JackTrip.

Visto que Jacktrip é um sistema que pode ser utilizado em sistemas operacionais baseados em Linux, utilizando a distribuição Ubuntu com sistema operacional. A aplicação foi hospedada em computadores portáteis, *raspberry pi's* e mini-computadores.

Outro elemento muito importante para levar em consideração são as interfaces de áudio utilizadas na implementação foram utilizadas três interfaces: Scarlett 2i2 3rd generation, Scarlett 2nd generation, and Presouns Audioobox 44 VSL 4614. Essas são interfaces de áudio profissionais, o que significa que elas permitem menores latências na captação e transmissão de

áudio. Para fins avaliativos básicos, uma interface de áudio de baixo custo foi também utilizada: Behringer UCA202.

Partindo desses equipamentos, foi implementado o sistema. Foi implementado um cliente e um servidor executando a aplicação Jacktrip conectados a uma interface de áudio, utilizando-se uma conexão via internet para conectar cliente e servidor.

4.3 Análise Objetiva

Em um sistema de performance musical via rede, três elementos influenciam a latência global drasticamente:

- Tamanho do *Buffer* da Interface de áudio: dada uma taxa de amostragem fixa, o tamanho do buffer dita a disponibilidade dos pacotes.
- Tamanho do *Buffer* de Jitter da aplicação: áudio é enviado via redes em pacotes, o tamanho do buffer dita a disponibilidade de pacotes para envio.
- Conexão de rede: Influência a latência de transmissão de um ponto a outro.

4.3.1 Influência do tamanho do buffer utilizado em uma interface de áudio

O estudo da influência do tamanho do *buffer* de uma interface de áudio na latência de disponibilização desse áudio pode ser medido de forma isolada, visto que essa propriedade é referente somente a interface de áudio e não ao sistema completo.

Menores tamanhos de *buffer* implicam em uma menor latência para disponibilização de áudio. Entretanto, interfaces de áudio não são capazes de lidar com qualquer tamanho de *buffer*, devido à capacidade de processamento.

Foi executado um experimento para avaliar o menor tamanho de *buffer* dentre interfaces de áudio utilizadas no experimento, mantendo uma frequência de amostragem de 48000 Hz. Para isso, foram utilizados dois métodos:

- Função do *driver* Alsa:

O *driver* Alsa tem uma função interna para testar interfaces de áudio chamada "*latency.c*", permitindo avaliar as latências. Para executar esse teste, é necessário conectar a saída de áudio à entrada de áudio da interface. Ao ser executada, a função emite um sinal sonoro e recupera o mesmo sinal. É definido um valor mínimo inicial, e o método irá avaliar se a interface de áudio é capaz de processar áudio com aquele tamanho de *buffer*. Caso ela não

seja capaz de lidar com aquele tamanho de *buffer*, um XRUN Error é levantado, indicando a falta de pacotes (*underrun*) ou excesso de pacotes (*overrun*). Em seguida, o tamanho de *buffer* é acrescido e testado novamente, até que um valor funcional seja encontrado.

Executando esse teste nas interfaces de áudio disponíveis, foram obtidos os seguintes resultados:

- Scarlett 2i2 3rd generation - tamanho de *buffer* mínimo = 16 amostras
- Behringer UCA202 - tamanho de *buffer* mínimo = 96 amostras
- Método utilizando captação de áudio: O segundo método foi desenvolvido para encontrar o menor tamanho funcional de *buffer*, dado uma interface de áudio utilizando os sinais de áudio captados.

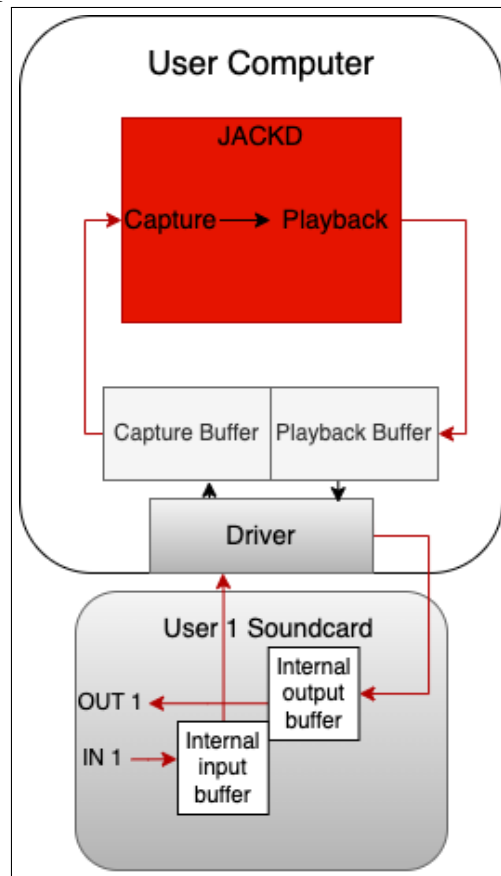
O protocolo utilizado foi reduzir manualmente o tamanho de *buffer* e avaliar o momento em que o áudio captado passa a ter distorções advindas de perdas de pacotes de áudio ou adiantamento de pacotes.

Além disso, foi calculada a latência para o menor tamanho de *buffer* que a interface de áudio consegue gerir.

O cálculo dessa latência baseou-se no atraso entre o áudio referência, captado por um sistema de controle que foi responsável por gerar o sinal para avaliação, e o áudio captado pelo sistema avaliado. Cada um dos sistemas é formado por um computador e uma interface de áudio. O princípio desse protocolo é a captação de áudio de forma a zerar a latência advinda de elementos além do sistema de avaliação.

Para realizar este estudo, inicialmente, um computador foi conectado a uma interface de áudio, como visto na figura 22. Nesta figura, vemos o canal de entrada chamado "IN1" e o canal de saída "OUT1". O áudio captado pelo canal de entrada é colocado no *buffer* interno da interface de áudio, em seguida, o computador capta esses dados através do *driver* JACKD. Os dados captados são mantidos no *buffer* interno do *driver*. Redirecionamos as informações capturadas para a saída da mesma interface de áudio, passando pelos *buffers* internos do *driver* e da interface de áudio.

Figura 22 – Protocolo de avaliação de latência de uma interface de áudio - Configuração da interface de áudio no computador do usuário

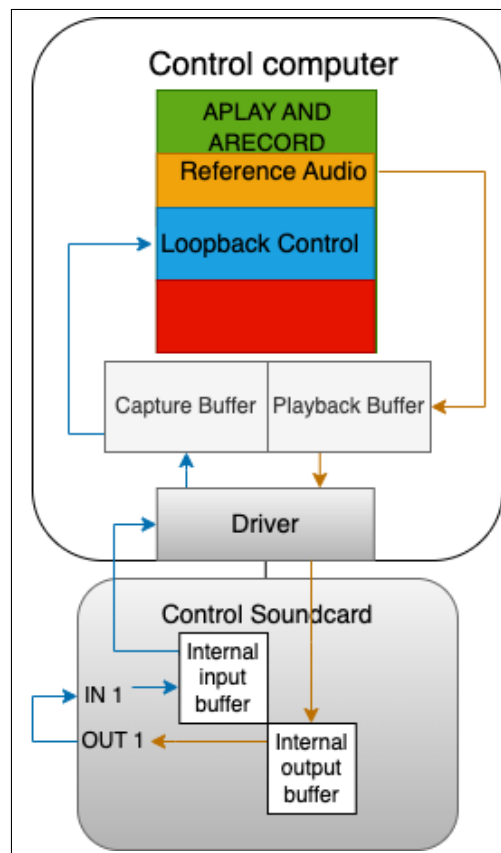


Fonte: elaborada pelo autor. (2021)

Em seguida, um computador de controle foi configurado, como visto na 23. O objetivo do computador de controle é gerar sinais de áudio que serão enviados para os sistemas cuja latência deseja-se medir e, em seguida, capturar o áudio enviado após ser recebido e reenviado pelo sistema.

Visto que a interface de áudio conectada ao computador de controle incide latência sob o sistema, foi feito um *loopback* para captar o sinal que é gerado pelo computador de controle e estudar a sua latência. A distância no tempo entre o sinal do *loopback* e o sinal referência é a latência que o computador de controle e sua interface de áudio trazem ao sistema completo. Para capturar e reproduzir sinais de áudio, utilizou-se os comandos *aplay* e *arecord*, duas funções integradas na biblioteca de *driver* ALSA.

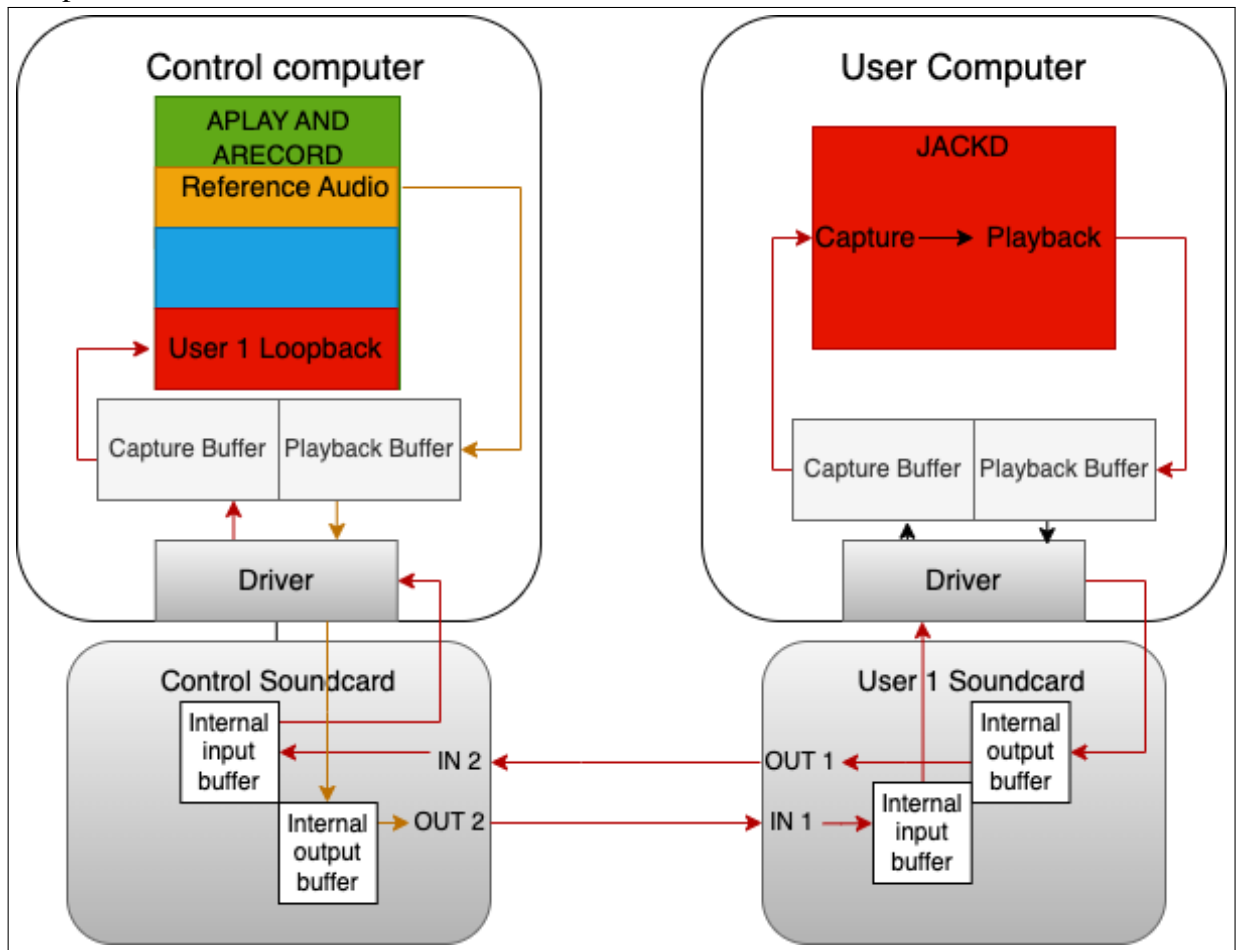
Figura 23 – Protocolo de avaliação da latência de uma interface de áudio - Configuração do computador de controle



Fonte: elaborada pelo autor. (2021)

Em seguida, o canal de saída de áudio do computador de controle foi conectado à entrada do computador do usuário. Um *loopback* foi feito e o sinal reenviado pelo computador do usuário é gravado pelo computador de controle.

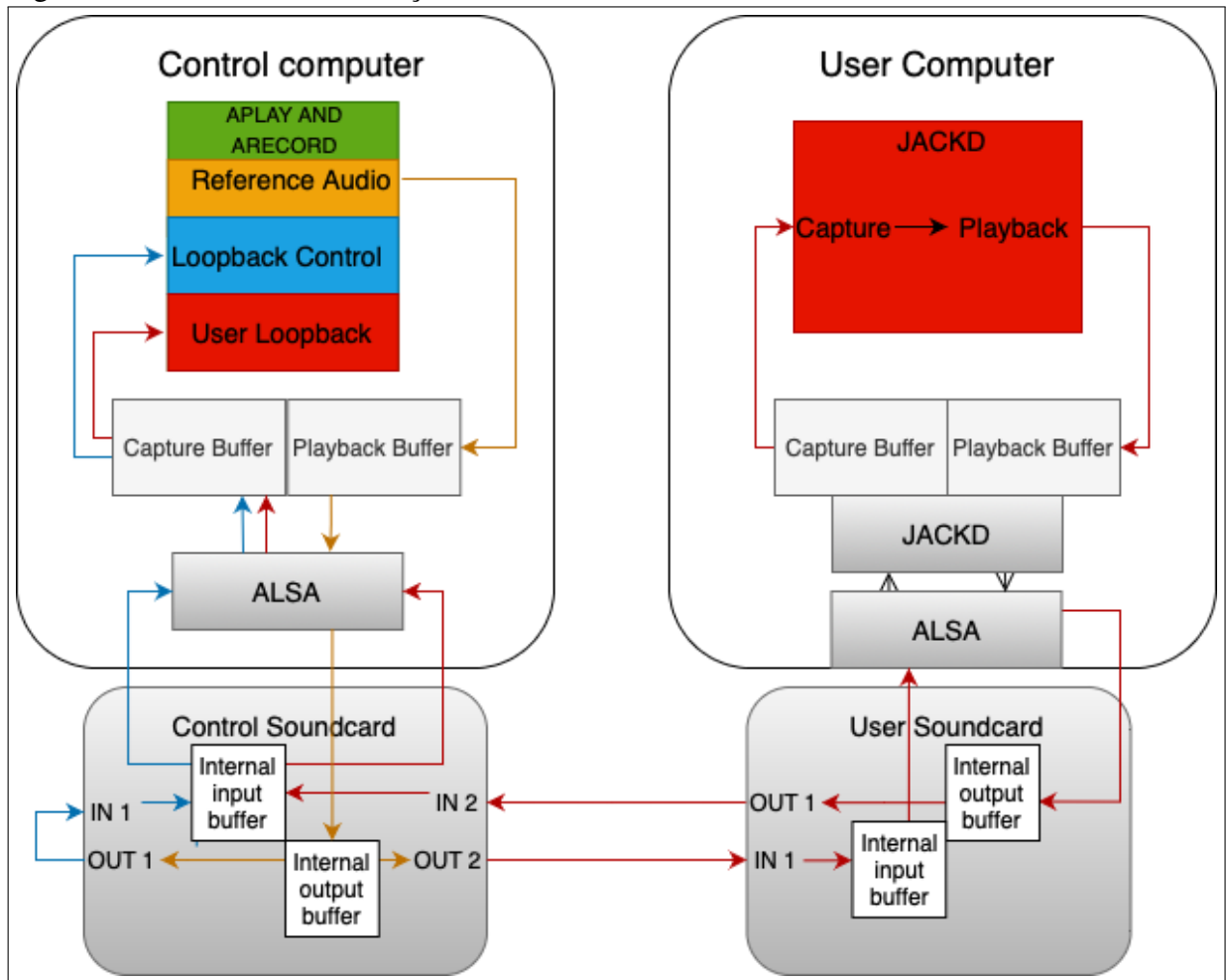
Figura 24 – Protocolo de avaliação da latência de uma interface de áudio - Envio de áudio para o computador do usuário



Fonte: elaborada pelo autor. (2021)

Na Figura 25, pode-se observar todas as conexões feitas para cada medida. Cada execução grava os áudios "*Loopback Control*" e "*User Loopback*" simultaneamente. A distância temporal entre esses dois áudios é a latência do sistema do usuário (interface de áudio e computador).

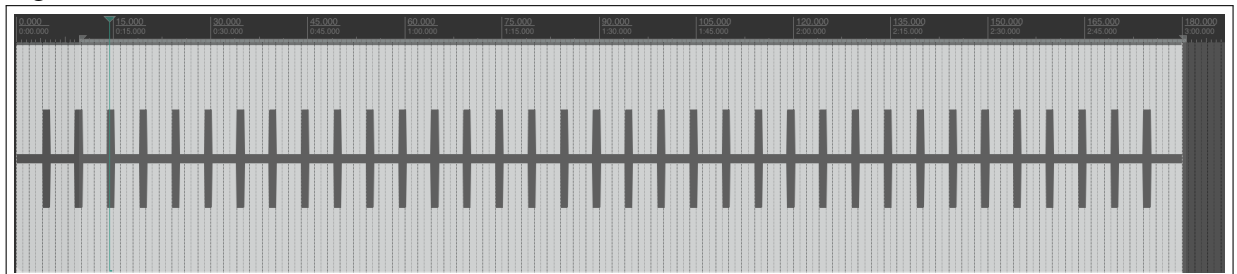
Figura 25 – Protocolo de avaliação da latência de uma interface de áudio



Fonte: elaborada pelo autor. (2021)

Durante os experimentos, o seguinte áudio sintético, uma varredura, foi utilizado, como mostrado na figura 26:

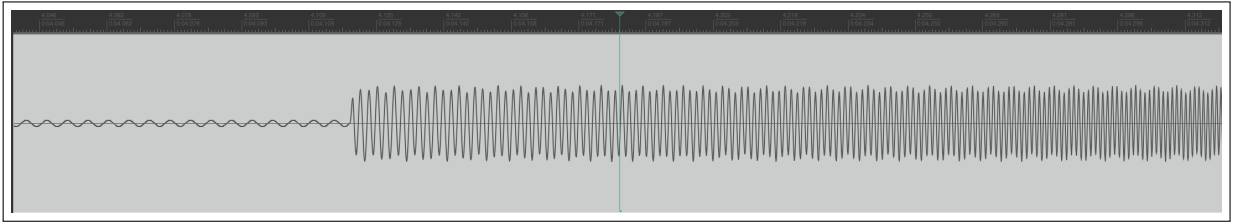
Figura 26 – Forma de onda de áudio sintético



Fonte: elaborada pelo autor. (2021)

Uma varredura é resultado de um sinal no qual as frequências variam com o tempo. Na varredura gerada as frequências variam de 500Hz à 1500Hz em 1 segundo. Conforme visto na Figura 27, a frequência aumenta com o tempo.

Figura 27 – Forma de onda de áudio sintético com zoom



Fonte: elaborada pelo autor. (2021)

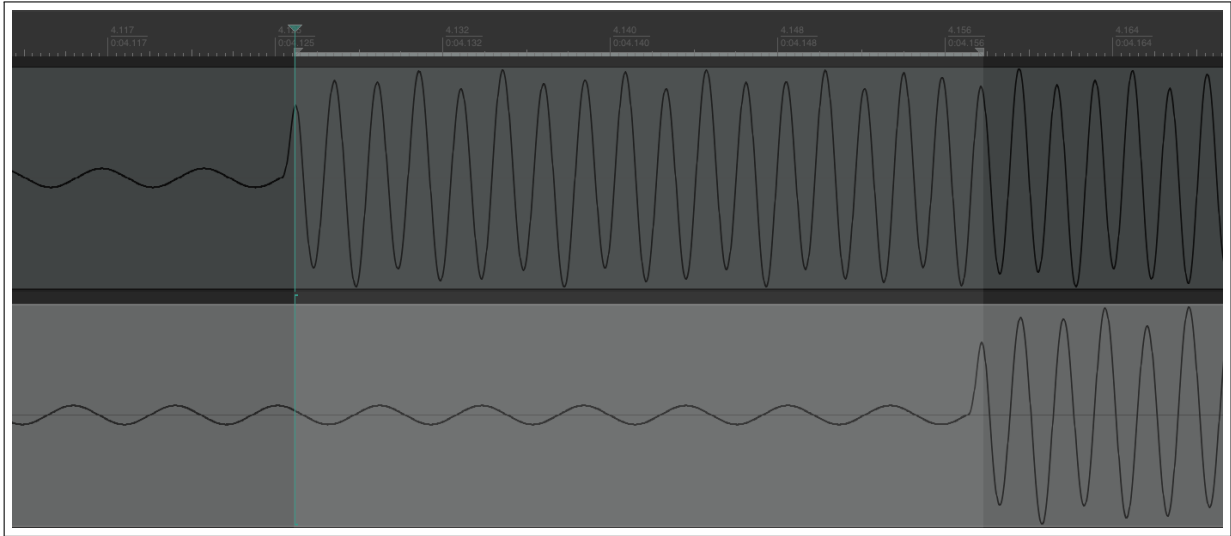
Cada experimento possui diferentes testes e cada teste contém múltiplas tentativas. Uma tentativa é equivalente ao envio do áudio sintético uma vez, assim como sua captura no retorno dos *loopbacks*.

O "computador de controle" reproduziu um sinal de áudio sintético. No caso, este estava conectado a uma placa de som, chamada de "placa de som de controle". Esse áudio foi direcionado para a placa de som conectada a um segundo computador, chamado respectivamente de "placa de som do usuário" e "computador do usuário". A captura de áudio foi direcionada para a reprodução da placa de som do usuário e capturada na placa de som do computador de controle, gerando um arquivo de áudio chamado "*loopback* do usuário". Ao mesmo tempo, o áudio reproduzido pela placa de som foi capturado diretamente após sua saída, gerando um arquivo de áudio chamado "*loopback* de controle".

Para capturar e reproduzir o áudio simultaneamente, utilizou-se os comandos *aplay* e *arecord*, duas funções integradas na biblioteca de driver ALSA. O atraso entre o *loopback* de áudio do usuário e do controle é a latência presente devido à aquisição e reprodução de áudio. Usando este protocolo, as placas de som foram configuradas, usando diferentes tamanhos de *buffer*, e calculando a latência associada. Para tanto, 'tentativa' é definida como uma execução do áudio sintético, 'testes', como todas as tentativas feitas usando a mesma configuração, e 'experimento' como o conjunto de todos os testes.

Após a aquisição do áudio, é possível comparar os áudios de *loopback* do controle e do usuário e avaliar a latência entre as duas varreduras. Na figura 28, é possível ver como a latência pode ser calculada manualmente usando um DAW ou software de edição de áudio.

Figura 28 – Intuição sobre como calcular o atraso entre dois sinais



Fonte: elaborada pelo autor. (2021)

Entretanto o cálculo manual é pouco preciso, então foram utilizados os estudos encontrados em (JOUBAUD; PALLONE, 2021), seguimos o método do *Exponential Sweep Sine* (equação 4.1), onde um sinal vai em T segundos da frequência f_1 à frequência f_2 é enviado:

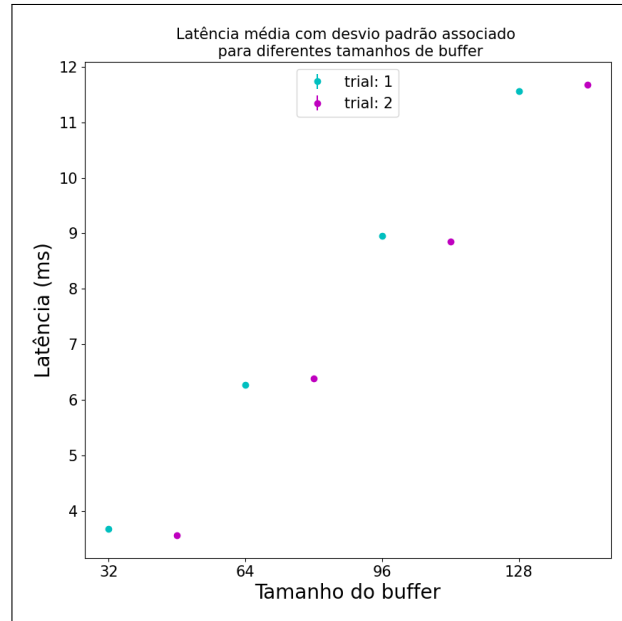
$$ESS(t) = \sin \left(\frac{2\pi f_1 T}{\ln \left(\frac{f_1}{f_2} \right)} \left(\exp \left(\frac{t}{T} \ln \left(\frac{f_1}{f_2} \right) \right) - 1 \right) \right) \quad (4.1)$$

O sinal inverso é convoluído com o sinal deslocado no tempo para obter uma função delta (*dirac*) centrada em $T + lat$, onde lat é a latência do sistema. Utilizou-se para todos os testes os seguintes valores:

- $f_1=500$ Hz
- $f_2=15000$ Hz
- $T= 1$ s

Esse protocolo foi utilizado para testar duas placas de som: a Scarlett 2i2 e a Behringer UCA202. Para a Scarlett 2i2, pode-se configurar um tamanho de buffer de 32 quadros e obter menos de 4ms de latência entre a captura e a reprodução. Os resultados mostrados têm qualidade de áudio perfeita (o desvio padrão é nulo). (Figura 29)

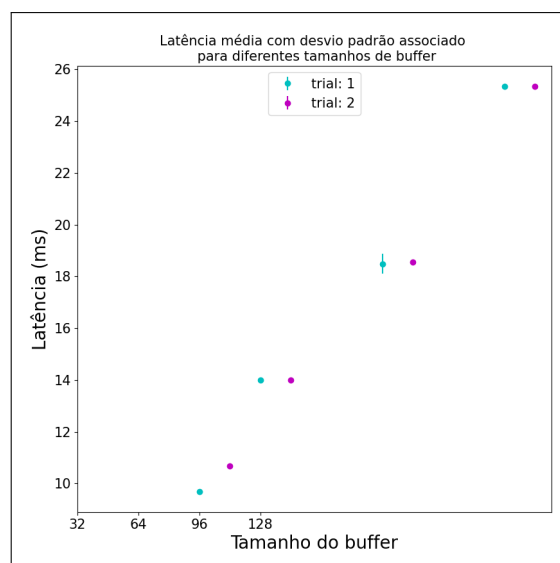
Figura 29 – Valores de latência (ms) para diferentes tamanhos de *buffer* para a placa de som Scarlett 2i2 de 3ª geração



Fonte: elaborada pelo autor. (2021)

Para a Behringer, pode-se ver que há necessidade usar um tamanho de *buffer* de 96 quadros, a fim de aumentar o valor da latência para até 19ms. Os resultados para os tamanhos de *buffer* mostrados têm qualidade de áudio perfeita. Há necessidade de um tamanho de *buffer* maior para prevenir áudio danificado. Para valores iguais de tamanho de *buffer*, as duas placas de som têm diferentes valores de latência, podendo-se inferir que a qualidade do hardware influencia os valores de latência. (Figura 30)

Figura 30 – Valores de latência (ms) para diferentes tamanhos de *buffer* para a placa de som Behringer UCA202



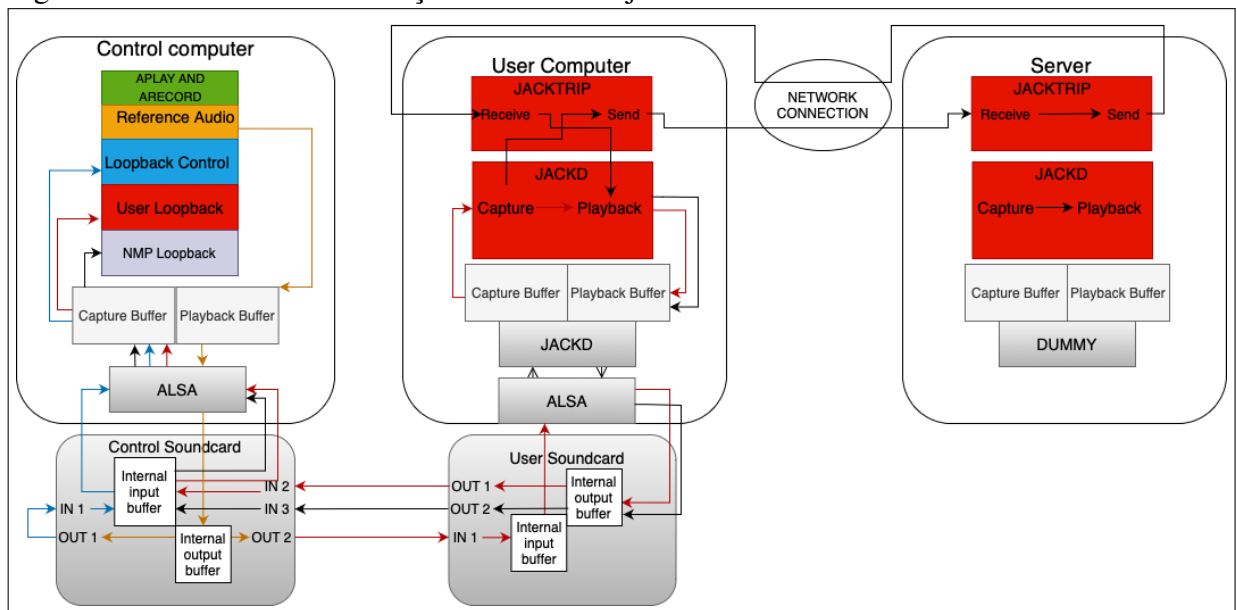
Fonte: elaborada pelo autor. (2021)

4.3.2 Influência do tamanho do buffer de jitter da aplicação utilizado

Os *buffers* de *jitter* são usados para compensar o *jitter* da rede, que pode danificar a qualidade do áudio devido ao fato de os pacotes chegarem atrasados. No entanto, os *buffers* de *jitter* aumentam o atraso, pois aumentam o tempo entre o recebimento das informações de áudio em pacotes e o seu processamento. Isso permite garantir a qualidade do áudio apesar do *jitter* da rede. Cada *buffer* de *jitter* tem o mesmo tamanho que o buffer da placa de som configurada, e são usados múltiplos *buffers* de *jitter* para prevenir o *buffer* da rede. Um bom número de *buffers* de *jitter* depende da qualidade de serviço da rede, que pode ser influenciada por múltiplos elementos na arquitetura da rede. Como o *buffer* de *jitter* depende da conexão de rede e de sua arquitetura, deve-se levar em consideração a conexão de rede ao analisar o buffer de *jitter*.

Para estudar tanto o *buffer* de *jitter* quanto a conexão de rede ao mesmo tempo, o seguinte protocolo foi elaborado (figura 31):

Figura 31 – Protocolo de avaliação do buffer de jitter e conexão de rede



Fonte: elaborada pelo autor. (2021)

Usando o Jacktrip como aplicativo de NMP, um computador de controle foi utilizado para configurar uma sessão em um computador do usuário e em um servidor. Foi conectado o cliente ao servidor. Nesse intuito, foi padronizado um tamanho de *buffer* de 64 quadros e uma taxa de amostragem de 48000 Hz nas sessões do usuário e do servidor para comparar resultados e analisar apenas a influência do *buffer* de *jitter* e da conexão de rede na latência. Vale ressaltar que o Jacktrip permite escolher o número de *buffers* de *jitter* utilizados.

Para diferentes números de *buffers* de *jitter*, foi executada uma sessão do Jacktrip enviando áudio do computador do usuário para o servidor e de volta. Para cada valor, foram feitas várias tentativas a fim de avaliar a replicabilidade dos resultados. O áudio (*sweep*) enviado através do Jacktrip provém do computador de controle. O computador de controle também é responsável pela captura do *loopback* de sua placa de som, da placa de som do usuário, e do áudio enviado pelo servidor. Cada tentativa tem a duração de 5 minutos.

Para automatizar o processo de configuração do servidor e do cliente Jacktrip, o computador de controle acessou o outro computador usando acesso remoto via *Secure shell* (SSH). Ao final de um experimento, tem-se por resultado três arquivos de áudio:

- *Loopback* de áudio do controle: *loopback* do áudio reproduzido pelo computador de controle, que é a referência para latência
- *Loopback* de áudio do usuário: *loopback* do áudio capturado pelo computador do usuário (ou seja, latência devido ao computador do usuário, excluindo a latência devido ao Jacktrip)
- *Loopback* de áudio do NMP: *loopback* do áudio enviado pelo servidor para o computador do usuário

Os arquivos de áudio adquiridos permitem calcular as seguintes latências:

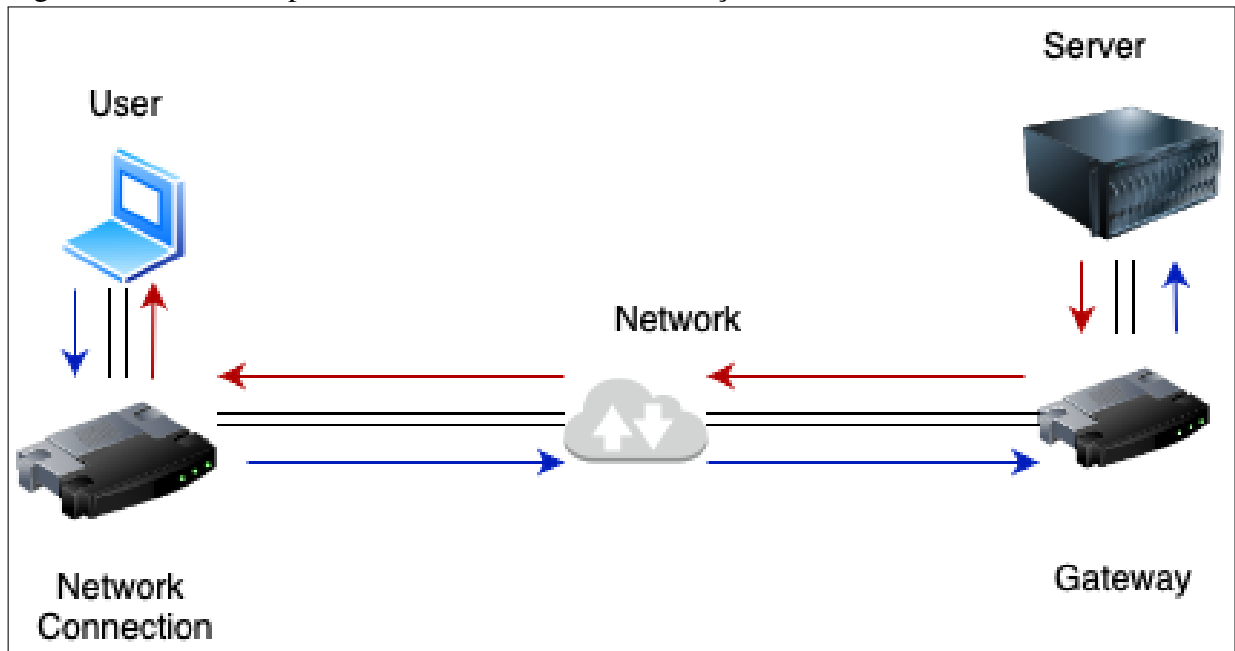
- Latência do sistema NMP: o atraso entre o *loopback* do NMP e o *loopback* do áudio do controle. Isso representa a latência total do sistema NMP.
- Latência de conexão de rede: o atraso entre o *loopback* do usuário e o *loopback* do áudio do controle.

Isso representa a latência do computador do usuário, ao se excluir todos os aspectos de rede.

A diferença entre a latência do sistema NMP e a latência da conexão de rede representa a latência da rede (incluindo o processamento do servidor).

A Figura 32 mostra o fluxo de pacotes durante os experimentos.

Figura 32 – Fluxo de pacotes de rede durante as simulações



Fonte: elaborada pelo autor. (2021)

4.3.3 Influência da conexão de rede na latência de um sistema NMP

Para cada experimento, a latência medida pela função integrada do Jacktrip foi mantida, esta que se baseia no atraso de transmissão dos pacotes de rede em vez de se basear no áudio. Este valor permite comparar os resultados com a latência calculada da conexão de rede. As funções do Jacktrip permitem medir o número de pacotes perdidos devido ao *jitter* de rede, o que mostra como o *buffer* de *jitter* reduz a perda de pacotes. Durante os experimentos, para cada conexão de rede, foram testados vários números de *buffers* de *jitter*. O servidor foi mantido sempre conectado à internet por fibra óptica e variamos as conexões de rede no computador do usuário. Os testes foram realizados nas seguintes configurações de rede:

- Computador do usuário e servidor em conexão de rede local
- Computador do usuário e servidor conectados à fibra óptica, mas em diferentes *gateways*, posições geograficamente separadas (diferentes localidades em Rennes)
- Computador do usuário conectado à 4G e servidor conectado à fibra óptica
- Computador do usuário conectado à 5G comercial e servidor conectado à fibra óptica
- Computador do usuário conectado à 5G experimental (provedor1) e servidor conectado à fibra óptica
- Computador do usuário conectado à 5G experimental (provedor2) e servidor conectado à fibra óptica

Para o experimento, também foram utilizadas plataformas experimentais de 5G de diferentes provedores. Foi uma oportunidade para testar o NMP com redes móveis. Durante os testes, utilizaram-se os seguintes equipamentos:

- Computador de controle: Computador HP
- Placa de som de controle: Scarlett 6i6 de 2ª geração
- Computador do usuário: Intel NUC
- Placa de som do usuário: Scarlett 2i2 de 3ª geração
- Servidor: Intel NUC

Para processar o som recebido da rede, o servidor foi configurado usando um driver chamado *Dummy*. Ele permite redirecionar as informações de áudio recebidas sem uma placa de som real.

Para cada experimento, são apresentados os seguintes resultados por meio de gráficos:

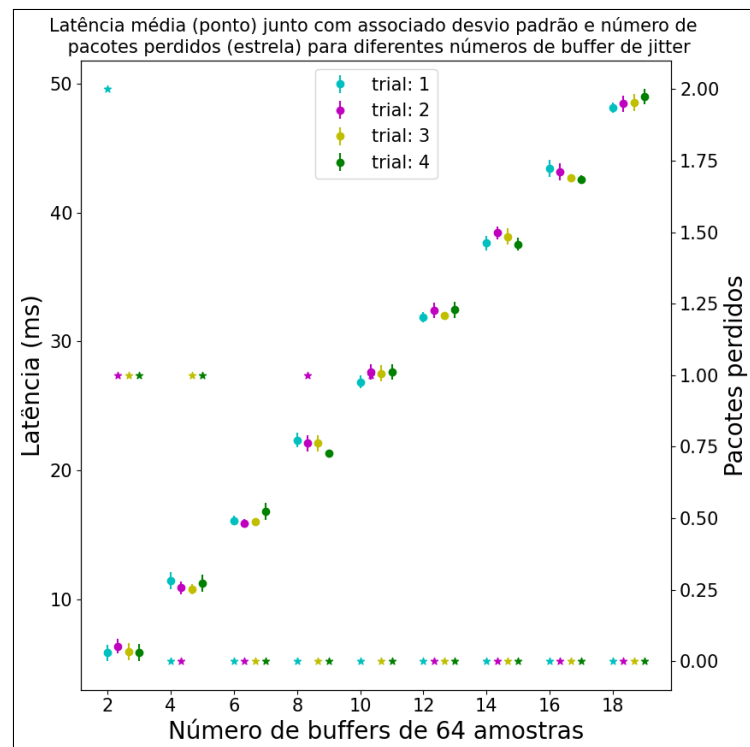
- Latência de rede calculada usando o áudio capturado pelo computador de controle
- Latência de rede medida usando a função integrada do Jacktrip
- Latência global do sistema: Latência de aquisição de áudio + latência de rede

Para este trabalho, foram tomados como norte os casos de uso não profissionais, como ensaios e aulas de música. Destarte, será considerado normal se houver alguns pacotes perdidos durante a performance (máximo de 1 por minuto).

4.3.3.1 *Conexão Local*

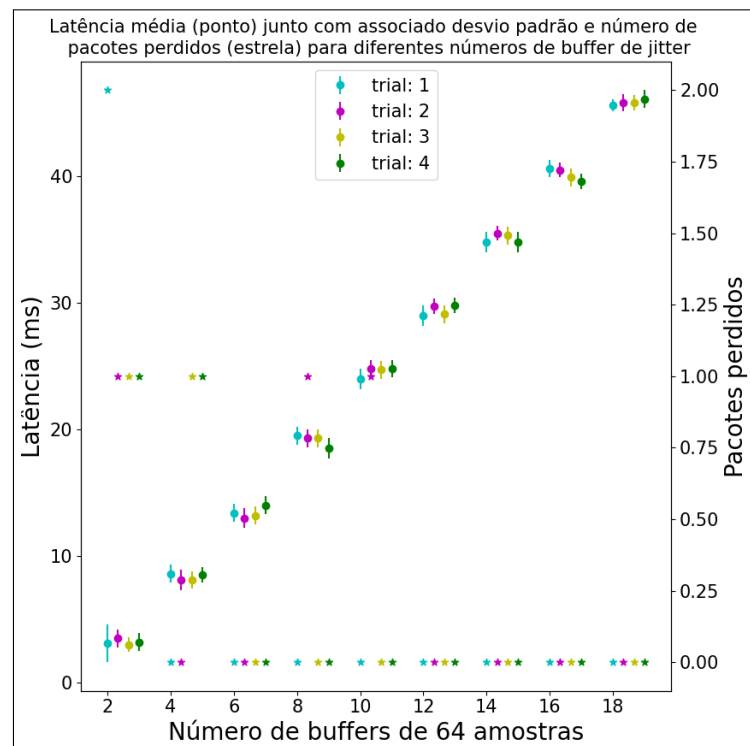
Primeiramente, foi estudado o cenário de melhor caso de conexões de rede. Este caso ocorre quando o computador do usuário e o servidor estão conectados em *ethernet* à mesma rede. Os resultados deste caso de uso são a referência para as outras conexões de rede testadas. Nas figuras 33, 34 e 35, vê-se que 2, 4 ou 6 *buffers* de *jitter* de 64 amostras são suficientes para um número aceitável de pacotes perdidos devido ao *jitter*, já que há um máximo de 1 pacote perdido em 5 minutos (considera-se neste trabalho que 1 pacote perdido por minuto era aceitável). A configuração ideal permanece abaixo do limite de 50 ms de latência de ida e volta.

Figura 33 – Latência de rede usando conexão de rede local calculada usando o áudio capturado pelo computador de controle



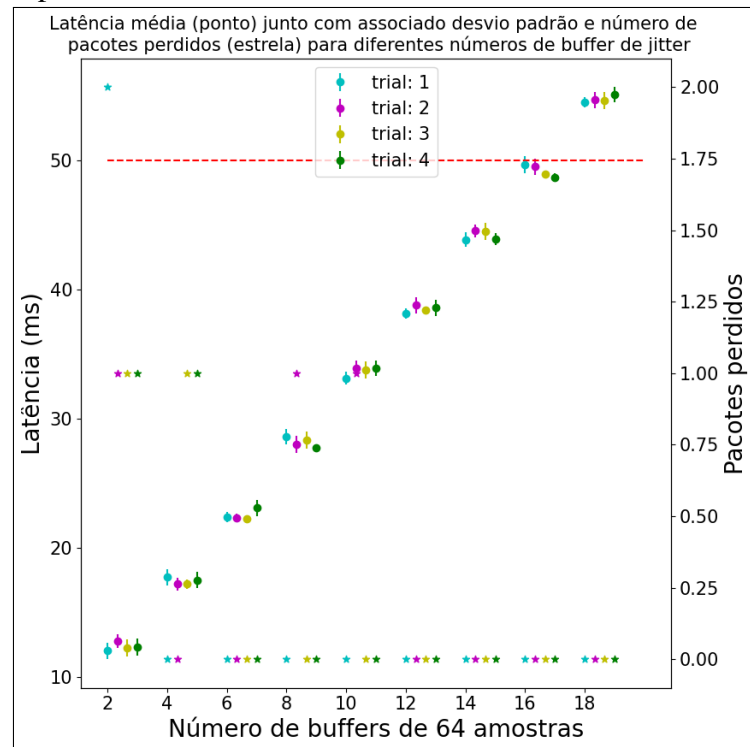
Fonte: elaborada pelo autor. (2021)

Figura 34 – Latência de rede usando conexão de rede local calculada usando funções integradas do Jacktrip



Fonte: elaborada pelo autor. (2021)

Figura 35 – Latência global do sistema usando conexão de rede local calculada usando o áudio capturado pelo computador de controle

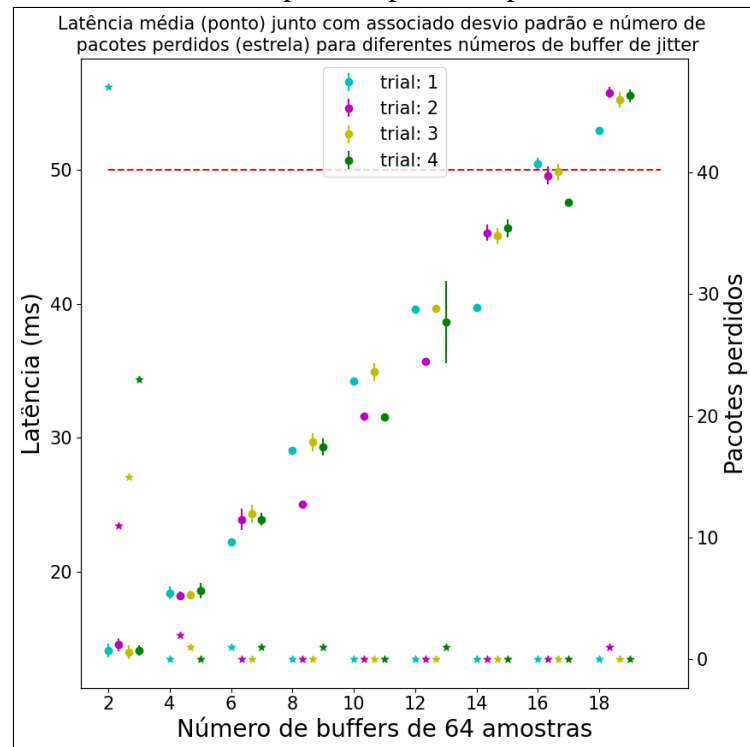


Fonte: elaborada pelo autor. (2021)

4.3.3.2 Fibra óptica

A internet de fibra óptica pode enviar dados a setenta por cento da velocidade da luz, resultando em transmissões de baixa latência. Vale ressaltar, no entanto, que equipamentos entre *links* de fibra adicionam latência. Usando conexão de fibra óptica no lado do usuário, foram feitos dois experimentos de locais diferentes. No primeiro (figura 37), o usuário e o servidor estavam conectados a diferentes roteadores, mas geograficamente próximos. No segundo, o usuário e o servidor estavam a 5 km de distância, conforme visto na figura 36.

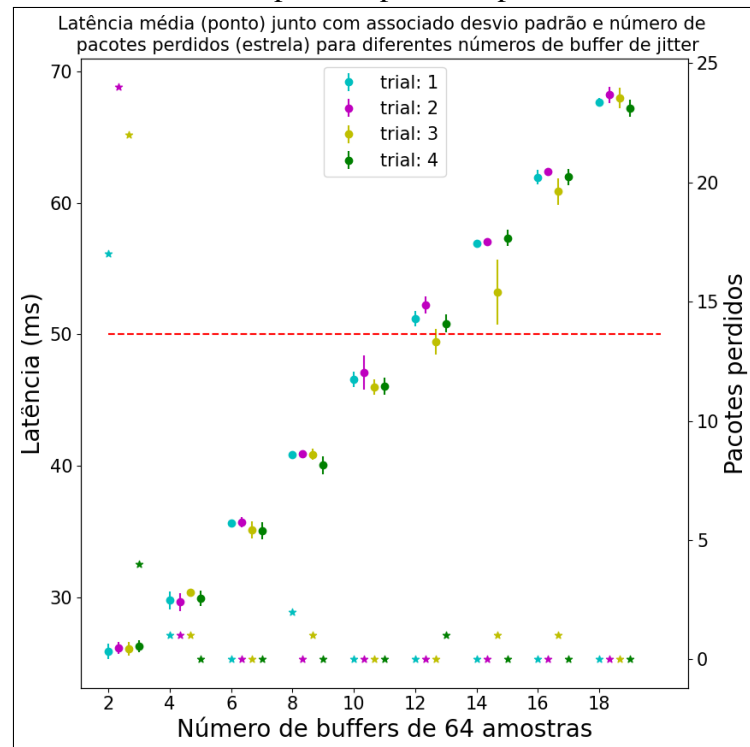
Figura 37 – Latência global do sistema usando conexão de rede de fibra óptica (cliente e servidor próximos) calculada usando o áudio capturado pelo computador de controle



Fonte: elaborada pelo autor. (2021)

O segundo experimento revela como a localização geográfica e, portanto, os equipamentos encontrados, podem alterar a latência da rede. Usando o mesmo número de *buffers* de *jitter*, há latências maiores, como visto nas figura 38.

Figura 38 – Latência global do sistema usando conexão de rede de fibra óptica (cliente e servidor distantes) calculada usando o áudio capturado pelo computador de controle



Fonte: elaborada pelo autor. (2021)

Devido à arquitetura de rede, não é possível prever qual caminho um pacote vai tomar entre dois computadores. Mesmo se dois roteadores estão no mesmo prédio, os pacotes não vão diretamente de um roteador para o outro. Por exemplo, mesmo se o usuário e o servidor estão na mesma cidade, os pacotes de rede transmitidos podem viajar até Paris antes de chegar ao outro extremo (isso é comum quando os usuários têm diferentes provedores de internet).

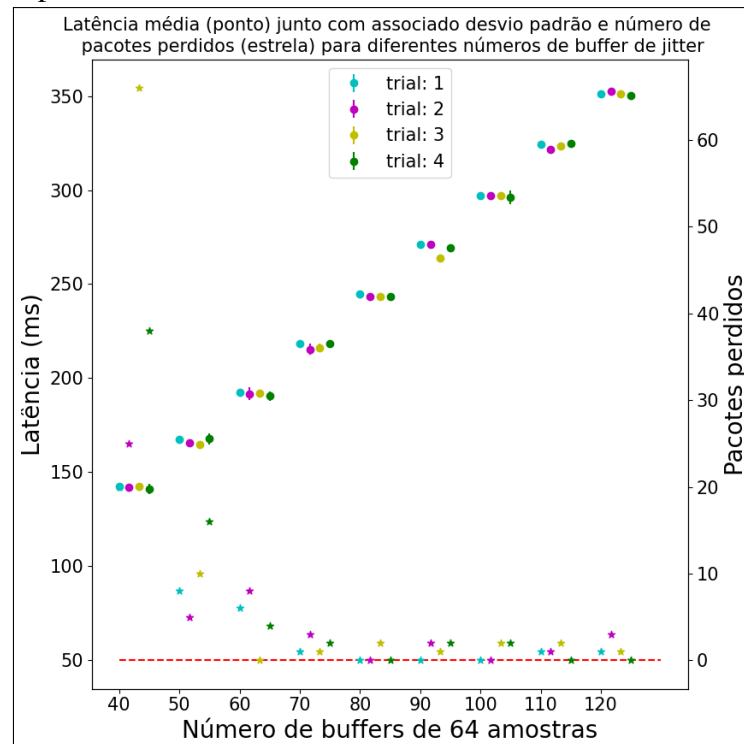
A imprevisibilidade do caminho do pacote pode explicar por que os valores de latência no segundo experimento são 10 ms maiores que no primeiro, visto que 5 km não é uma distância considerável para um aumento de latência dessa ordem de grandeza, uma vez que a informação é transmitida quase à velocidade da luz. Em condições específicas, a conexão de fibra óptica permite ter resultados tão bons quanto em conexão local, mas isso depende da arquitetura da rede

4.3.3.3 4G Comercial

Nos resultados usando a conexão 4G, os testes específicos indicam que esta tecnologia não é adaptada para requisitos de baixa latência. Ao menos 80 *buffers* de *jitter* devem ser usados para absorver o *jitter* da rede e latências próximas a 250 ms são alcançadas. Este valor

está bem acima do limite de latência de 50ms. De acordo com os testes, a conexão 4G não parece ser uma opção viável para o NMP. (Figura 39)

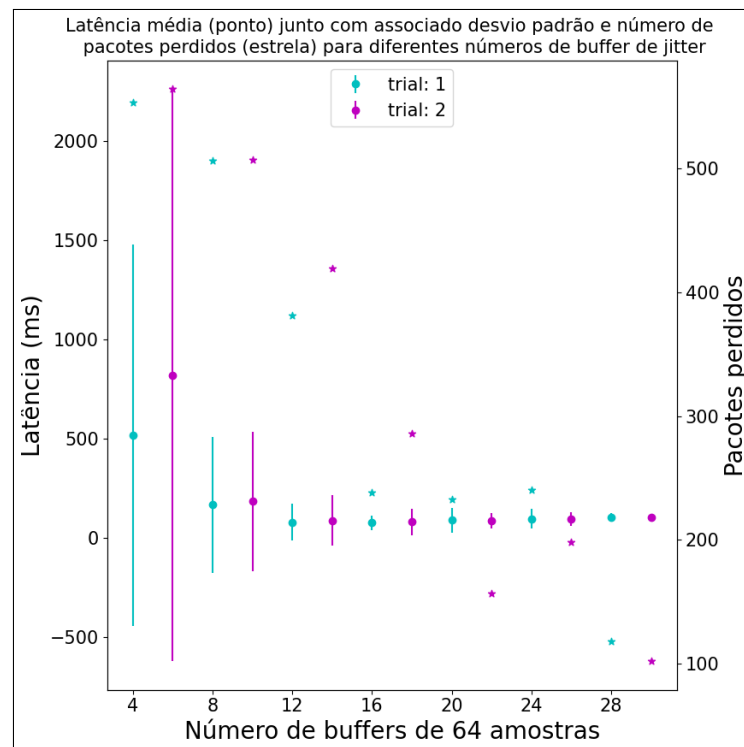
Figura 39 – Latência global do sistema usando conexão de rede 4G calculada usando o áudio capturado pelo computador de controle



Fonte: elaborada pelo autor. (2021)

Este teste permite concluir que a função integrada do Jacktrip que mede a latência de rede não é precisa em altos *jitters* de rede. Como visto na figura 40.

Figura 40 – Latência de rede usando conexão de rede 4G medida usando funções integradas do Jacktrip.



Fonte: elaborada pelo autor. (2021)

4.3.3.4 5G Comercial

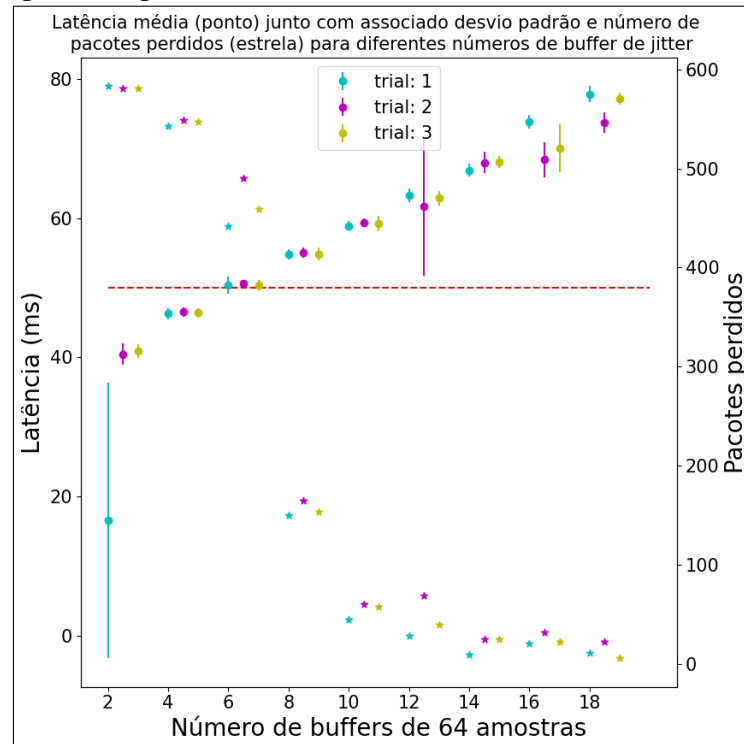
O 5G é a quinta geração da tecnologia de redes móveis. As tecnologias sem fio 5G prometem oferecer maior largura de banda, latência ultra-baixa e mais confiabilidade do que o 4G. Foram testadas 3 tecnologias:

- 5G Comercial: 5G já disponível comercialmente
- 5G do Fornecedor1: 5G experimental e privado, SA (*stand-alone*) a 2.6 GHz. Valores disponíveis: *bitrate* >260Mbps para baixo, e 65 Mbps para cima, latência em torno de 7ms e jitter em torno de 3 ms
- 5G do Fornecedor2: 5G experimental e privado, NSA (*não stand-alone*) a 26 GHz. Valores disponíveis: *bitrate* >1Gbps para baixo, e 130 Mbps para cima, latência em torno de 7ms e jitter em torno de 3 ms

Durante os estudos, um iPhone 12 Pro foi usado para acessar o 5G comercial disponível. Infelizmente, os resultados mostram que não é possível usar o 5G comercial atual, como visto na figura 41. De fato, a arquitetura atual da rede móvel não é adequada para o caso de uso do NMP, porque todos os dados transmitidos são enviados para Paris para alcançar a rede fixa e voltar para Rennes, locais em que os experimentos foram realizados. A figura 41 mostra

que um grande número de *buffers* de *jitter* é necessário para garantir a qualidade do áudio, e aumentando o valor da latência acima dos limites toleráveis. Por outro lado, usar um número suficientemente pequeno de *buffers* de *jitter* leva a um valor de latência aceitável (abaixo de 50ms), mas o *jitter* leva a um número inaceitável de pacotes perdidos.

Figura 41 – Latência global do sistema usando conexão de rede 5G comercial calculada usando o áudio capturado pelo computador de controle

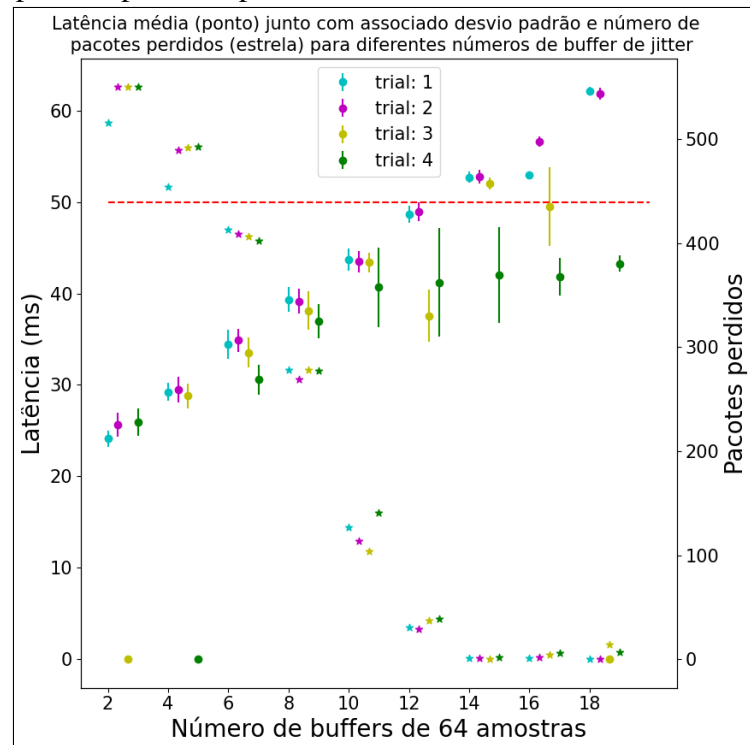


Fonte: elaborada pelo autor. (2021)

Para usar o 5G de Fornecedor1 e Fornecedor2, foi utilizado o acesso aos laboratórios experimentais 5G. Estudos sobre a arquitetura de rede e a configuração dessas duas tecnologias não foram conduzidos durante os testes e avaliações. Os experimentos foram principalmente utilizados como ferramentas para os engenheiros que trabalham nessas tecnologias para avaliar a qualidade da tecnologia.

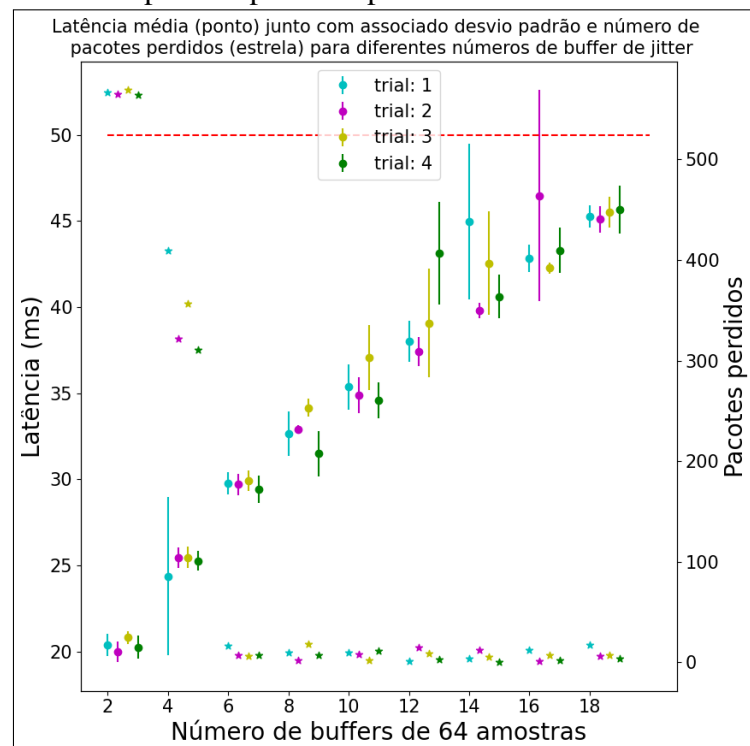
O 5G de Fornecedor1 e Fornecedor2 mostra resultados promissores e se aproxima da fibra óptica em termos de latência. Não foi possível responder especificamente a razão pela qual o 5G do Fornecedor2 (figura 43) tem melhores resultados do que o 5G do Fornecedor1 (figura 42).

Figura 42 – Latência de rede usando conexão de rede 5G experimental do Fornecedor1 calculada usando o áudio capturado pelo computador de controle



Fonte: elaborada pelo autor. (2021)

Figura 43 – Latência global do sistema usando conexão de rede 5G experimental do Fornecedor2 calculada usando o áudio capturado pelo computador de controle

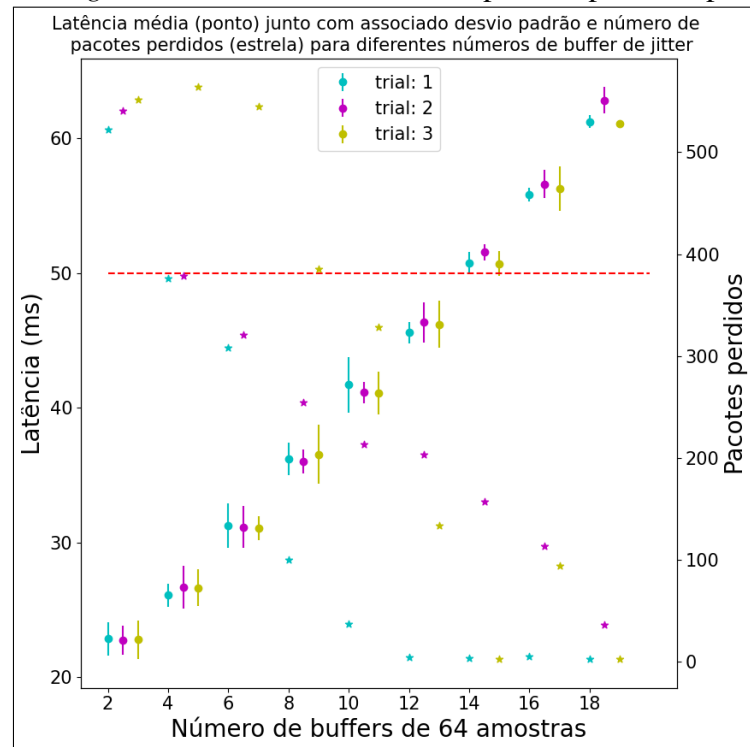


Fonte: elaborada pelo autor. (2021)

4.3.3.5 5G e Fortinet

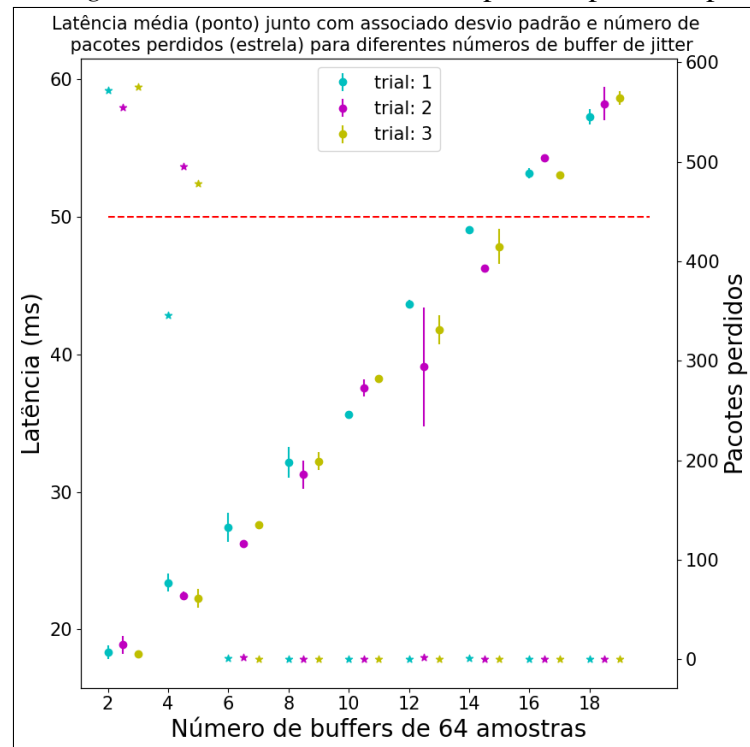
Por último, foi realizado um teste com um servidor Edge. Um servidor Edge é um servidor localizado na borda da rede (integrado à rede móvel 5G) para reduzir a latência. Foi instalado um servidor nos Laboratórios Orange atrás de um equipamento de *firewall* chamado "Fortinet", para depois realizar os experimentos. Os resultados mostram que isso pode ser uma estratégia interessante para reduzir a latência da rede tanto para o 5G do provedor1 (figura 44) quanto do provedor2 (figura 45).

Figura 44 – Latência global do sistema usando a conexão de rede 5G experimental do provedor1 com um servidor em *edge* calculada usando o áudio capturado pelo computador de controle



Fonte: elaborada pelo autor. (2021)

Figura 45 – Latência global do sistema usando a conexão de rede 5G experimental do provedor2 com um servidor em *edge* calculada usando o áudio capturado pelo computador de controle



Fonte: elaborada pelo autor. (2021)

4.3.3.6 Avaliação da influência da conexão de rede em um sistema NMP

A Tabela 3 apresenta um resumo dos estudos realizados sobre a influência da conexão de rede em sistemas NMP. Nossas análises revelaram que é possível realizar performances musicais remotas utilizando diferentes tipos de conexão, como conexões locais (entre cômodos diferentes), fibra óptica (entre localizações geográficas distintas) e até mesmo o 5G experimental. No entanto, observamos que as conexões de 5G comercial e 4G não são capazes de oferecer condições ideais para esse tipo de performance até o momento.

Tabela 3 – Resultados dos experimentos sobre a influência da conexão de rede em um sistema NMP

Conexão de rede	Latência média (ms)
Local	12
Fibra óptica	18
4G	243
5G Comercial	62
5G Experimental	29

Fonte: elaborada pelo autor. (2021)

Esses resultados são baseados em medições de latência média obtidas durante nossos

experimentos. Verificamos que conexões locais apresentam a menor latência, proporcionando condições favoráveis para performances musicais em tempo real, com uma média de 12 ms. A fibra óptica também demonstrou ser uma opção viável, com uma média de 18 ms de latência. No entanto, é importante ressaltar que as conexões de 5G comercial apresentaram uma média de 62 ms de latência, enquanto as conexões de 4G apresentaram uma média de 243 ms de latência, tornando-as menos adequadas para performances musicais em tempo real. Por outro lado, a tecnologia 5G experimental mostrou resultados promissores, com uma média de 29 ms de latência, sendo uma opção mais viável em comparação com as demais opções de conexão.

Esses resultados destacam a importância de uma conexão de rede estável e de baixa latência para a realização de performances musicais via rede. Além disso, eles indicam o potencial do 5G experimental como uma opção futura para melhorar ainda mais as condições de desempenho em sistemas NMP.

No entanto, é importante mencionar que esses resultados foram obtidos com base em nossos experimentos específicos e podem variar dependendo do ambiente, configurações e infraestrutura de rede utilizadas. Portanto, é fundamental realizar análises específicas para cada caso e considerar a evolução das tecnologias de rede para obter resultados ainda mais precisos e atualizados.

4.4 Análise Subjetiva

Como visto anteriormente, diferentes valores de latência unidirecional implicam em diferentes experiências de performance. Para estudar como a latência impacta a experiência de tocar, foram testados vários intervalos de valores de latência.

Durante o experimento, buscou-se avaliar os valores na tabela 4, já apresentados.

Tabela 4 – Sincronicidade no tempo, fidelidade e percepção de latência para diferentes intervalos de atraso

Latência (ms)	Sincronicidade no tempo	Fidelidade ao tempo	Perceptibilidade da latência
0-8	Impreciso	Aceleração	Nenhuma
8-25	Natural	Natural	Nenhuma
25-60	Impreciso	Desaceleração	Perceptível ou Tolerável
60 ou maior	Impossível	Impossível	Tolerável ou Intolerável

Fonte: elaborada pelo autor. (2021)

Em um teste, entre dois músicos em Rennes, foi configurado o sistema com uma latência no intervalo de 0 ms a 8 ms de latência unidirecional. Não houve percepção de nenhuma

latência e não foi sentida aceleração do tempo. Ressalta-se, sobretudo, que nenhum protocolo de avaliação preciso foi usado. Em seguida, configurou-se para o intervalo de 8 ms a 25 ms. Não houve percepção a latência para esses valores, pois foi percebida estabilidade de sincronização de tempo. Para latências maiores que 25 ms, começou-se a perceber a latência, e o tempo comum diminuiu progressivamente, o que acarretou em percalços para a performance musical.

Em outro teste, três músicos tocando em conjunto avaliaram o sistema. Um músico cantando e tocando violão em Rennes, o segundo tocando baixo em Nantes e o terceiro tocando saxofone em Paris. Em geral, definiram a experiência como satisfatória e relataram que a performance era possível. Mesmo com a percepção da latência, ainda pode-se encontrar maneiras de lidar com o atraso.

Houve também a produção de um vídeo tocando "Garota de Ipanema", um dos clássicos da Bossa Nova. Não houve percepção de latência durante a performance. Nesta performance, o baixo estava na rede 5G conectado à internet através de um *gateway* de fibra, e as guitarras atrás de 2 outros *gateways* de fibra, usando as seguintes placas de som: Scarlett 2i2 3ª geração, Scarlett 2ª geração, e Presouns Audiobox 44 VSL 4614.

Por razões inerentes ao tempo de realização do estudo, não foi possível, formalmente, encontrar uma metodologia adequada para estudar a experiência subjetiva.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, nosso objetivo foi explorar o estado da arte, implementar e analisar um sistema de performance musical via rede. Iniciamos definindo o conceito de performance musical via rede e seus diversos casos de uso. Em seguida, compreendemos a importância da latência e seus efeitos na comunicação em rede, estabelecendo faixas de latência que viabilizam as performances musicais remotas. Delimitamos o escopo técnico de um sistema de performance musical via rede e investigamos como cada componente contribui para a latência geral do sistema. Estudamos a cadeia de aquisição de áudio, desde as interfaces de áudio até as configurações de rede mais adequadas para comunicação entre os usuários do sistema. Analisamos as aplicações existentes no mercado e selecionamos o Jacktrip como base para o desenvolvimento do nosso sistema. Em seguida, desenvolvemos protocolos de teste para investigar as latências decorrentes da aquisição de áudio, da aplicação e da rede. Realizamos testes com múltiplas conexões de rede para avaliar em quais cenários é possível realizar performances musicais remotas. Por fim, conduzimos testes subjetivos, nos quais músicos tocaram em tempo real e de forma sincronizada em diferentes localizações geográficas.

Ao longo do estudo, adquirimos um entendimento sofisticado sobre o funcionamento dos *drivers* das placas de som e fornecemos detalhes aprofundados sobre as aplicações NMP. Nossos experimentos foram focados no desenvolvimento de protocolos de avaliação meticulosamente projetados, que facilitaram a análise de diversos cenários de uso de forma prática. Integrando conhecimentos de processamento de sinal, percepção musical e ciência da computação, conseguimos realizar uma fusão harmoniosa dessas disciplinas.

As performances musicais via rede abrangem múltiplos domínios e representam um campo vasto. Com a pandemia de SARS-CoV-2, observamos um aumento do interesse e investimento nesse campo por parte de novas empresas. Identificamos diversas oportunidades para inovação, especialmente no sentido de tornar as configurações necessárias mais acessíveis e fáceis de implementar para usuários comuns.

Como resultado geral, conseguimos implementar com sucesso um sistema de performance musical via rede. Nossas implementações e testes confirmaram que as performances musicais remotas são viáveis, desde que sejam utilizadas conexões de rede robustas e placas de som que garantam latências abaixo de 25ms, com 50ms de ida e volta. No entanto, é importante ressaltar que esses sistemas ainda são limitados por restrições físicas, mesmo com o uso de fibra óptica. Os resultados obtidos com tecnologias como o 5G são promissores, e a evolução

das arquiteturas de rede e o emprego de servidores *Edge* podem viabilizar aplicações de baixa latência em NMP, tornando-as mais populares e acessíveis em escala global.

REFERÊNCIAS

- 675, R. *Specification of Internet Transmission Control Program*. RFC Editor, 1974. RFC 675. (Request for Comments, 675). Disponível em: <<https://www.rfc-editor.org/info/rfc675>>.
- 675, R. *User Datagram Protocol*. RFC Editor, 1980. RFC 768. (Request for Comments, 768). Disponível em: <<https://www.rfc-editor.org/info/rfc768>>.
- CÁCERES, J.-P.; CHAFE, C. Jacktrip: Under the hood of an engine for network audio. **Journal of New Music Research**, Taylor & Francis, v. 39, n. 3, p. 183–187, 2010.
- CARÔT, A.; WERNER, C. External latency-optimized soundcard synchronization for applications in wide-area networks. In: **Proceedings of the AES Regional Convention, Tokyo, Japan**. [S. l.: s. n.], 2009.
- CARÔT, A. Livemusic on the internet. **Diploma thesis, Fachhochschule Lübeck**, Citeseer, Germany, 2004.
- CCRMA. **Instructions for installing Jack and Jacktrip**. 2023. Disponível em: <https://ccrma.stanford.edu/software/jacktrip/osx/index.html>. Acesso em: [Online; accessed 2023].
- ELKAUDIO. **Aloha Website**. 2023. Disponível em: <https://www.elk.audio>. Acesso em: 28 jun. 2023.
- FARNER, S.; SOLVANG, A.; SÆBO, A.; SVENSSON, U. P. Ensemble hand-clapping experiments under the influence of delay and various acoustic environments. **Journal of the Audio Engineering Society**, v. 57, n. 12, p. 1028–1041, December 2009.
- FISCHER, V. **Jamulus Website**. 2023. Disponível em: https://jamulus.io/pt_BR/. Acesso em: 28 jun. 2023.
- HUPKE, R.; BEYER, L.; NOPHUT, M.; PREIHS, S.; PEISSIG, J. Effect of a global metronome on ensemble accuracy in networked music performance. In: AUDIO ENGINEERING SOCIETY. **Audio Engineering Society Convention 147**. [S. l.], 2019.
- JAMULUS. **GitHub Jamulus repository**. 2021. Disponível em: <https://github.com/jamulussoftware/jamulus>.
- JOUBAUD, T.; PALLONE, G. Electroacoustic method for the calibration of a heterogeneous distributed speaker system. In: **2020 28th European Signal Processing Conference (EUSIPCO)**. [S. l.: s. n.], 2021. p. 476–480.
- LABS, J. **Jacktrip Website**. 2023. Disponível em: <https://www.jacktrip.com>. Acesso em: 28 jun. 2023.
- OS, E. A. **GitHub Repository of ELK Audio OS and RASPA**. 2021. Disponível em: <https://github.com/elk-audio>.
- REPP, B. H. Sensorimotor synchronization: a review of the tapping literature. **Psychonomic bulletin & review**, Springer, v. 12, n. 6, p. 969–992, 2005.
- ROTTONDI, C.; CHAFE, C.; ALLOCCHIO, C.; SARTI, A. An overview on networked music performance technologies. **IEEE Access**, IEEE, v. 4, p. 8823–8843, 2016.

ROTTONDI, C. E. M.; BUCCOLI, M.; ZANONI, M.; GARAO, D. G.; VERTICALE, G.; SARTI, A. Feature-based analysis of the effects of packet delay on networked musical interactions. Università degli Studi di Milano, 2015.

SAVOLA, P. **Overview of the Internet Multicast Routing Architecture**. RFC Editor, 2008. RFC 5110. (Request for Comments, 5110). Disponível em: <https://www.rfc-editor.org/info/rfc5110>.

SCHUETT, N. The effects of latency on ensemble performance. **Bachelor Thesis, CCRMA Department of Music, Stanford University**, Citeseer, 2002.

SONOSAURUS. **Sonobus Website**. 2023. Disponível em: <https://sonobus.net>. Acesso em: 28 jun. 2023.

STAIS, C.; THOMAS, Y.; XYLOMENOS, G.; TSILOPOULOS, C. Networked music performance over information-centric networks. In: IEEE. **2013 IEEE International Conference on Communications Workshops (ICC)**. [S. l.], 2013. p. 647–651.

VALIN, J.-M.; MAXWELL, G.; TERRIBERRY, T. B.; VOS, K. High-quality, low-delay music coding in the opus codec. **arXiv preprint arXiv:1602.04845**, 2016.

WAH, B. W.; SU, X.; LIN, D. A survey of error-concealment schemes for real-time audio and video transmissions over the internet. In: **Proceedings International Symposium on Multimedia Software Engineering**. [S. l.: s. n.], 2000. p. 17–24.