

Failover da UPF em Núcleo 5G *Cloud-Native*: Arquitetura de *Testbed* e Avaliação Experimental

Fausto da Silva Moraes*, Gustavo Dias de Oliveira*, Jonas Augusto Kunzler*, Jonathas dos Santos*, Lucas Borges de Oliveira†, Murilo Cruz da Silva†, Diego Medeiros de Abreu†, Gustavo Hermínio de Araújo† e Flávio Geraldo Coelho Rocha*

Resumo— A UPF representa um elemento crítico do núcleo 5G, especialmente em cenários de computação de borda, nos quais a indisponibilidade dessa função pode comprometer sessões ativas e degradar a experiência do usuário. Este trabalho apresenta um *testbed* experimental para avaliação de mecanismos de *failover* da UPF em um núcleo 5G *cloud-native*, utilizando Open5GS, srsRAN, rádio X310 e UE comercial. O ambiente implementado permite comparar diferentes estratégias de recuperação, analisando métricas como tempo de *downtime*, continuidade da sessão PDU, uso de CPU e RAM, além de indicadores associados à QoE em aplicações multimídia reais. Os resultados experimentais mostram que o ajuste de parâmetros internos do mecanismo de *heartbeat* do Open5GS impacta diretamente o tempo de recuperação percebido pelo UE, reduzindo significativamente o *downtime*. A comparação entre experimentos conduzidos com UE real e ambientes emulados evidencia comportamentos semelhantes, indicando a viabilidade de extrapolação dos resultados para cenários escaláveis com múltiplos UEs. Por fim, avaliações com transmissão adaptativa MPEG-DASH demonstram o impacto do tamanho do *buffer* da aplicação na percepção de interrupções durante a mudança de âncoras de sessões entre UPFs.

Palavras-Chave— 5G, 5G Core, UPF, *failover*, *testbed*

Abstract— The UPF represents a critical component of the 5G Core, especially in edge computing scenarios, where the unavailability of this function may compromise active sessions and degrade the user experience. This work presents an experimental *testbed* for evaluating UPF *failover* mechanisms in a *cloud-native* 5G Core, using Open5GS, srsRAN, X310 radio, and commercial UE. The deployed scenario enables the comparison of different recovery strategies by analyzing metrics such as *downtime*, PDU session continuity, CPU and RAM, as well as QoE-related indicators in real multimedia applications. Experimental results show that tuning internal parameters of the Open5GS *heartbeat* mechanism impacts the recovery time perceived by the UE, significantly reducing *downtime*. The comparison between experiments conducted with real UE and emulated UEs demonstrates similar behaviors, indicating the feasibility of extrapolating the results to scalable scenarios with multiple UEs. Finally, evaluations using MPEG-DASH adaptive streaming demonstrate the impact of the application *buffer* size on the perception of interruptions during session anchor switching between UPFs.

Keywords— 5G, 5G Core, UPF, *failover*, *testbed*

*Autores vinculados ao Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE) da Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) da Universidade Federal de Goiás (UFG). E-mails: faustomoraes@discente.ufg.br, gustavodias@ufg.br, k_jonasaugusto@ufg.br, jonathas_santos@discente.ufg.br, flaviogcr@ufg.br.

†Autores vinculados à Rede Nacional de Ensino e Pesquisa (RNP). E-mails: lucas.oliveira@rnp.br, murilo.silva@rnp.br, diego.abreu@rnp.br, gustavo.araujo@rnp.br.

Este trabalho foi parcialmente financiado pela Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG) por meio do Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE), e no âmbito do Programa OpenRAN@Brasil, apoiado pelo Ministério da Ciência, Tecnologia e Inovação, com recursos da Lei nº 8.248, de 23 de outubro de 1991, de acordo com as diretrizes da Secretaria de Empreendedorismo e Inovação.

I. INTRODUÇÃO

As redes móveis de quinta geração (5G) consolidaram uma arquitetura de núcleo (5G Core) baseada em funções de rede desagregadas, virtualizadas e orquestradas de forma dinâmica [1], [2]. Entre essas funções, a Função do Plano de Usuário (User Plane Function – UPF) executa o encaminhamento do plano de usuário, aplica políticas de tráfego e interliga a Rede de Acesso por Rádio (Radio Access Network – RAN) à Rede de Dados (Data Network – DN) por meio de Protocolo de Tunelamento GPRS para o Plano de Usuário (GPRS Tunneling Protocol for the user plane – GTP-U), sob controle da Função de Gerência de Sessão (Session Management Function – SMF) via interface Protocolo de Controle de Encaminhamento de Pacotes (Packet Forwarding Control Protocol – PFCP) [1], [3]. Em cenários de computação na borda e Indústria 4.0, a colocação da UPF próxima ao Equipamento de Usuário (User Equipment – UE) reduz latência fim a fim, mas concentra o risco operacional: a indisponibilidade de um único ponto de ancoragem pode comprometer sessões ativas e acarretar violações de requisitos de Qualidade de Serviço (Quality of Service – QoS) e confiabilidade [4], [5].

Em paralelo, o Open Radio Access Network (Open RAN) reforça os princípios de desagregação introduzidos pelo 5G [6]. O Open RAN promove interfaces padronizadas e interoperabilidade entre componentes de diferentes fornecedores, fomentando ecossistemas mais flexíveis, modulares e inovadores. Nesse ambiente, torna-se possível integrar uma miríade de implementações de UPF no plano de usuário, explorando novas oportunidades para otimização e controle de tráfego. Consequentemente, a adoção dessas arquiteturas abertas, impulsionadas pela O-RAN Alliance, aliada a núcleos *cloud-native*, amplia a necessidade de mecanismos de Alta Disponibilidade (High Availability – HA) que combinem padronização, flexibilidade de *software* aberto e implantação em plataformas de clusterização, como o Kubernetes [7]–[9].

Trabalhos Relacionados. Estudos recentes sobre resiliência do núcleo 5G em ambientes *cloud-native* abordam diferentes estratégias para aumento da disponibilidade e tolerância a falhas. Entre elas, destacam-se mecanismos de orquestração e escalabilidade em Kubernetes [10], [11], técnicas de migração e *failover* de contêineres agnósticas à aplicação [12], além de soluções voltadas especificamente à recuperação da UPF, incluindo detecção acelerada de falhas, redirecionamento de tráfego entre instâncias ativa-passiva e mecanismos de recuperação com menor custo computacional [13], [14]. De maneira complementar, trabalhos relacionados ao desempenho da UPF e à observabilidade baseada em *extended Berkeley Packet*

Filter (eBPF) reforçam a importância do plano de usuário na avaliação de métricas de QoS em redes móveis de próxima geração [3], [15], [16].

Apesar dos avanços recentes e da disponibilidade de implementações comerciais e acadêmicas, ainda persiste uma lacuna na caracterização experimental reproduzível de estratégias de *failover* da UPF em ambientes de borda utilizando pilhas abertas. Em particular, faltam estudos sistemáticos que avaliem, sob métricas homogêneas, os impactos de diferentes estratégias de redundância e recuperação de *pods* na experiência do usuário e em aplicações reais. Nesse contexto, o programa OpenRAN@Brasil [9], [17] fornece uma infraestrutura distribuída relevante para validação experimental de soluções abertas. Motivado por esse cenário, este trabalho apresenta um *testbed* multi-UPF com UE real e avaliação multimídia baseada em métricas de Qualidade de Experiência (*Quality of Experience* – QoE).

Contribuições. Neste contexto, este artigo apresenta trabalho desenvolvido no Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE)/Universidade Federal de Goiás (UFG) em parceria com a Rede Nacional de Pesquisa (RNP), com as seguintes contribuições: (i) definição de uma arquitetura de *testbed* multi-UPF para avaliação de resiliência no plano de usuário; (ii) avaliação experimental com métricas de conectividade, tempo de recuperação e continuidade de sessão; e (iii) resultados de cenários de *failover* em núcleo 5G *cloud-native* em ambiente base para avaliações futuras nas ilhas do OpenRAN@Brasil.

Organização do Artigo. O restante do artigo está organizado da seguinte forma: a Seção II descreve o *testbed* e a metodologia; a Seção III apresenta resultados experimentais do ambiente; a Subseção III-D apresenta a avaliação específica de uma aplicação multimídia real baseada em MPEG-DASH durante eventos de falha de UPF; e a Seção IV conclui o trabalho e indica perspectivas para trabalhos futuros.

II. ARQUITETURA DE TESTBED E METODOLOGIA

A. Visão Geral

O *testbed* multi-UPF foi concebido para isolar interfaces 3GPP do núcleo 5G (N3, N4, N6, SBI) e permitir a inserção controlada de falhas na UPF ativa. A Fig. 1 resume o arranjo lógico multi-UPF: a SMF mantém associações PFCP com mais de uma UPF e as monitora através de mensagens periódicas de *heartbeat*; em caso de falha da instância primária, a sessão Unidade de Dados de Protocolo (*Protocol Data Unit* – PDU) deve ser reancorada na UPF secundária sem perda permanente de conectividade do UE. A pilha do núcleo 5G do projeto Open5GS foi escolhida para a avaliação feita neste artigo pois, atualmente, é a principal pilha de código aberto que permite a configuração de redundância ativa-passiva (1:1) sem a necessidade de modificações no código ou do uso de tecnologias adicionais, como comutadores programáveis. A versão adotada para o Open5GS é a 2.7.5 com cada função virtual separada em um contêiner Docker exclusivo. Na SMF, as duas UPFs foram configuradas com a mesma DN e o mesmo espaço de endereçamento para a sessão dos UEs.

As métricas utilizadas são: (i) tempo de recuperação da sessão, ou *downtime* (interrupção até retomada de tráfego);

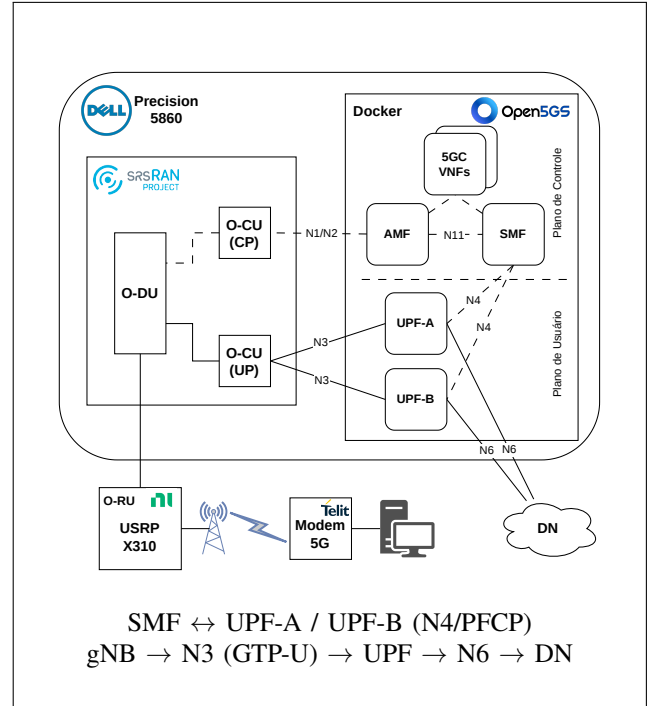


Fig. 1: Visão lógica do *testbed* multi-UPF para experimentação de *failover* (elementos e interfaces simplificados).

(ii) continuidade de conectividade IP do UE; e (iii) utilização de recursos (CPU/memória) nas UPFs ativa e passiva.

B. Equipamentos

A infraestrutura do *testbed* é composta por:

Workstation Dell Precision 5860: equipada com um processador Intel(R) Xeon(R) w3-2425 de 6 núcleos e 12 *threads*, com frequência de *clock* de aproximadamente 3,9 GHz; 64 GB de memória RAM DDR5; e armazenamento de 1 TB NVMe. Esse equipamento é responsável por executar cada função do núcleo 5G Open5GS em contêineres Docker separados. Além disso, é utilizado para execução da gNB utilizando o *software* srsRAN compilado na versão 25.0.4.0 com *driver* UHD na versão 4.8.0, necessário para comunicação com o rádio USRP por meio de uma interface Ethernet de 10 Gigabit (10GBASE-T). Uma estação de trabalho com as mesmas características de *hardware* foi utilizada como UE no *testbed*, equipada com um módulo 5G que permite a sua conectividade à rede móvel.

Rádio Ettus Research USRP X310: equipamento de Rádio Definido por *Software* (*Software Defined Radio* – SDR) que opera em faixa de frequência sub-6 GHz, com largura de banda máxima de 160 MHz. Possui capacidade de conectividade dupla com interfaces Ethernet de 10 Gigabit. Configurado para operar como a unidade de rádio da gNB com a largura de banda de 100 MHz.

Módulo 5G Telit FN990A40: utilizado como modem 5G conectado a uma estação de trabalho, permitindo a realização dos testes de conectividade e de *failover*.

Smartphone Samsung Galaxy S26 plus: aparelho celular comercial utilizado para avaliação de QoE durante transmissão multimídia.

III. AVALIAÇÃO EXPERIMENTAL

Os experimentos foram conduzidos utilizando o laboratório descrito na Fig. 1. Nesse cenário multi-UPF baseado no Open5GS, foram implementados roteiros automatizados para interrupção da UPF ativa e verificação da retomada do tráfego após a recuperação da sessão PDU.

Foi realizada uma comparação entre o cenário de redundância ativa-passiva, identificado nas legendas dos resultados como *failover*, e o cenário de recuperação dinâmica, no qual existe apenas uma UPF disponível para associação com a SMF, identificado como *restart*. Nesse segundo caso, após uma falha simulada, mecanismos nativos de orquestração de contêineres são responsáveis pela restauração da UPF. Cada configuração testada foi submetida a 10 repetições, das quais se extraiu a distribuição do *downtime*, medido como o tempo entre o primeiro pacote perdido pelo UE e o retorno da sua conectividade com a DN.

Para cada cenário de teste foi usado um *script*¹ que: automatiza o início do tráfego do UE; provoca a falha da UPF âncora da sessão PDU após o decorrer de 5 segundos; coleta dados do tempo de resposta dos pacotes enviados. Nesse *script* elaborado para os experimentos, os dados trafegados pelo UE são pacotes ICMP subsequentes com destino ao primeiro salto da DN, com o *timeout* definido em 50 ms. Um sinal do tipo SIGKILL provoca a falha da UPF ativa. O processo de *failover* ou *restart* é iniciado de forma automática pelo Open5GS após as mensagens de *heartbeat* enviadas pela SMF não serem respondidas pela UPF.

A. Avaliação de Downtime e Recuperação da Sessão

O Open5GS, por padrão, envia mensagens de *heartbeat* em intervalos de 10 segundos e, em caso de não haver resposta da UPF, faz 3 novas tentativas (*retries*) antes de considerar a UPF inacessível e fazer a sua desassociação. O valor de *timeout* de cada mensagem de *heartbeat* é dado pela razão entre a periodicidade do *heartbeat* e o número de novas tentativas somado a um. Ou seja, a diminuição do número de *retries* aumenta o tempo de *timeout* das mensagens de *heartbeat*.

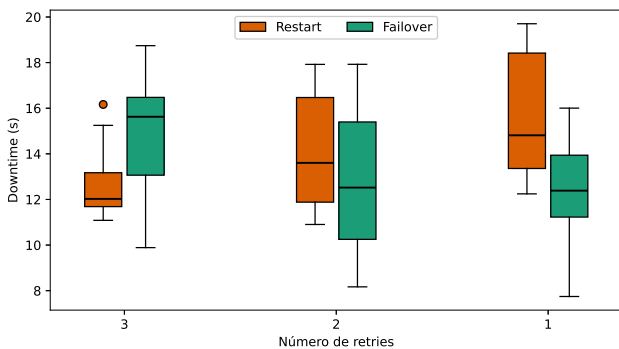


Fig. 2: Downtime percebido pelo UE após falha da UPF.

A Fig. 2 apresenta, na forma de diagrama de caixa (*boxplot*), a distribuição do *downtime* em relação à variação do número de *retries*, e mostra que o valor padrão do número de novas

tentativas adotado pelo Open5GS desfavorece cenários configurados com *failover* da UPF, uma vez que, mesmo que a UPF secundária já esteja disponível para assumir como âncora da sessão dos UEs afetados pela falha da UPF principal, a SMF deve aguardar o transcorrer do tempo de *timeout* das 3 novas tentativas da mensagem de *heartbeat*. Já no cenário de *restart*, em que é possível restaurar a operação da UPF por meio da sua reinicialização, novas tentativas sucessivas de obter uma resposta das mensagens de *heartbeat*, com um tempo de *timeout* menor, aumentam a chance de uma destas mensagens ser recebida pela UPF assim que ela esteja novamente apta a restabelecer a associação com a SMF.

Avaliando a redução do número de *retries* da mensagem de *heartbeat*, nota-se, Fig. 2, que o *downtime* percebido pelo UE diminui no cenário de *failover* e aumenta no cenário com *restart* da UPF. O parâmetro do número de *retries* no Open5GS é um valor *hardcoded* no código-fonte da aplicação e a sua modificação demanda a recompilação do *software*. Portanto, no cenário em que o operador da rede 5G tem a possibilidade de utilizar múltiplas UPFs com *failover* configurado, a adoção do Open5GS pré-compilado com um valor do número de *retries* menor que o valor padrão reduziria o *downtime* percebido pelos UEs em caso de falha em uma das UPFs.

B. Impacto no Uso de Recursos Computacionais

Em seguida, foram avaliados os impactos no uso de recursos computacionais decorrentes de cada uma das abordagens de recuperação da UPF ativa. Em especial, buscou-se mensurar qual seria a sobrecarga de memória e CPU gerada tanto pela inclusão de uma UPF adicional quanto pela migração dos usuários entre as UPFs. A Fig. 3a mostra que a inclusão de uma segunda UPF representa um aumento aproximado de 32 MB no uso de memória RAM. Esse valor equivale ao dobro do uso de memória no cenário em que há apenas uma UPF, mas se comparado ao uso de memória total do núcleo 5G o valor é de apenas aproximadamente 10%. Contudo, ao se avaliar o uso de CPU no cenário com uma UPF, mostrado na Fig. 3b, nota-se que o processo de recuperação por meio do *restart* da UPF atinge um pico de 4% de uso de CPU. Esse valor corresponde a 20 vezes o uso de CPU no cenário com uma UPF secundária, e equivale ao mesmo uso de CPU de todas as funções do núcleo 5G.

C. Comparação entre UE Real e Ambiente Emulado

Os resultados apresentados na Fig. 2 foram obtidos utilizando um UE real. Com o objetivo de avaliar se o comportamento do *failover* observado com UEs reais seria comparável ao obtido em ambientes com UEs e gNB emulados, ou seja, cenário no qual é possível escalar os experimentos para um número elevado de dispositivos, utilizou-se o *software* UERANSIM, versão 3.2.6.

A Fig. 4 apresenta o tempo médio de *downtime*, com intervalo de confiança de 95%, obtido para o teste de *failover* com o valor de *retry* definido em 1. Na comparação entre o cenário com infraestrutura real e o cenário contendo UEs emulados, a sobreposição dos intervalos de confiança sugere que a diferença entre os tempos de *downtime* não é estatisticamente significativa.

¹Os artefatos estão disponíveis em repositório aberto: <https://github.com/faustomoraes/upf-failover>

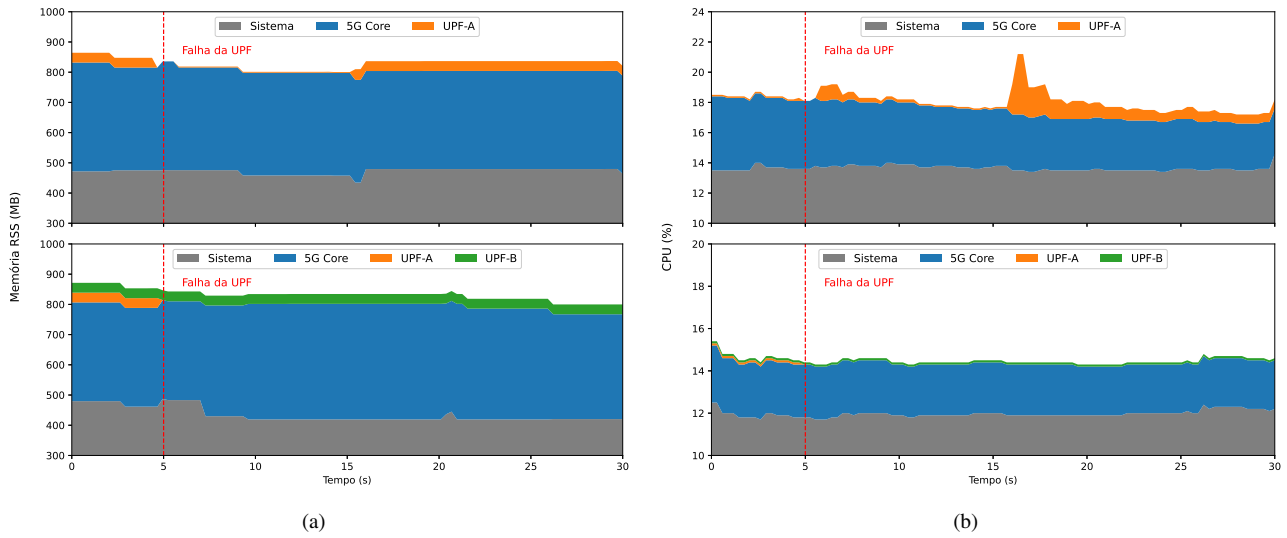


Fig. 3: (a) Uso de memória durante a falha da UPF ativa no cenário de *Restart* (acima) e *Failover* (abaixo); (b) Uso de CPU durante a falha da UPF ativa no cenário de *Restart* (acima) e *Failover* (abaixo).

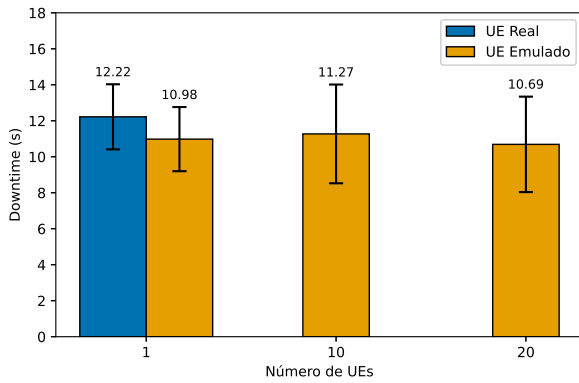


Fig. 4: *Downtime* percebido pelo UE real e pelos UEs emulados.

Dessa forma, conclui-se que os resultados obtidos em ambientes emulados podem ser utilizados como aproximação válida para os cenários reais, permitindo a extrapolação dos experimentos para contextos com múltiplos UEs, nos quais há maior facilidade de escalabilidade e automação das avaliações.

D. Avaliação de QoE em Aplicação Multimídia

O objetivo principal desta seção foi analisar o impacto da falha de uma UPF ativa durante uma sessão contínua de transmissão multimídia, verificando a manutenção da sessão PDU e a continuidade da reprodução do vídeo após a migração do fluxo de dados.

Para representar um cenário de tráfego multimídia mais próximo de aplicações reais, foi utilizada transmissão adaptativa baseada em MPEG-DASH (*Dynamic Adaptive Streaming over HTTP*) em uma aplicação web acessível pelo navegador do UE. A Fig. 5 é uma captura de tela do *smartphone* utilizado nos experimentos. Essa abordagem permite o controle granular do *buffer* do cliente e a avaliação de métricas de QoE, como

tempo total de interrupção (*stall time*), quantidade de eventos de congelamento e tempo de recuperação após o *failover*.

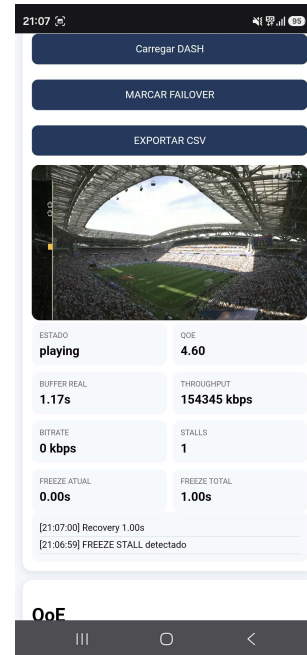


Fig. 5: Captura de tela da aplicação multimídia executando no UE real durante os experimentos.

Os vídeos utilizados nos experimentos foram segmentados em blocos de 1 segundo utilizando *FFmpeg*, com geração de manifesto *.mpd* e segmentos *.m4s*. A escolha de segmentos curtos foi realizada para aumentar a sensibilidade do sistema a interrupções na rede, reduzindo o efeito de mascaramento causado por *buffers* extensos no navegador. O cliente DASH foi implementado utilizando a biblioteca *dash.js*, permitindo configurar dinamicamente o tamanho do *buffer* de reprodução. Durante os testes, foram avaliados diferentes tamanhos de *buffer* local do cliente (2s, 5s, 10s, 15s, 20s e 25s), possibilitando

analisar como o armazenamento antecipado de segmentos influencia a percepção de falhas.

Os resultados experimentais demonstraram que o tamanho do *buffer* exerce impacto direto na percepção de continuidade do serviço durante o *failover* da UPF. *Buffers* maiores (15s, 20s e 25s) tendem a mascarar parcialmente a falha da rede, permitindo que a reprodução continue por alguns segundos mesmo após a interrupção da UPF primária. Entretanto, *buffers* reduzidos (2s, 5s e 10s) tornam o sistema significativamente mais sensível às falhas, evidenciando imediatamente a degradação do *throughput*, o esgotamento do *buffer* e o surgimento de eventos de *stall*. Esse comportamento permitiu observar com maior precisão o tempo necessário para recuperação da sessão multimídia após o redirecionamento do tráfego para a nova UPF, possibilitando avaliar a robustez do mecanismo de *failover* para a aplicação de vídeo.

A Fig. 6 demonstra o comportamento do *buffer* local do cliente DASH durante a troca de UPF. *Buffers* reduzidos sofrem esgotamento rapidamente após a interrupção da conectividade, ocasionando eventos de *stall*. Já *buffers* maiores mantêm conteúdo previamente armazenado, permitindo continuidade temporária da reprodução mesmo durante a falha da UPF.

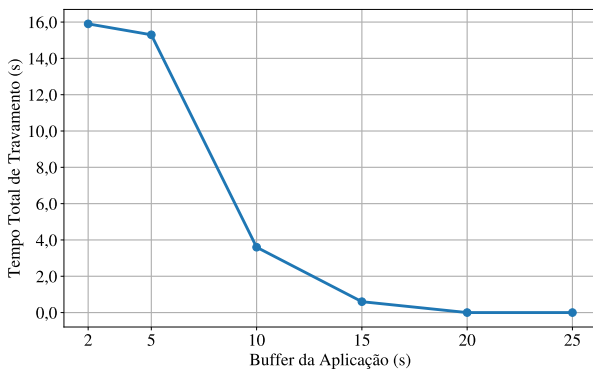


Fig. 6: Comportamento do *buffer* local do cliente DASH durante o *failover* da UPF.

Os resultados indicam que *buffers* superiores a aproximadamente 20 segundos passam a apresentar ganhos marginais na redução do *stall time* durante o *failover*.

IV. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo apresentou um *testbed* para investigação de *failover* da UPF em núcleo 5G *cloud-native*, fundamentado em trabalhos recentes sobre resiliência, orquestração e desempenho do plano de usuário [3], [10], [13], [15].

A avaliação comparativa dos cenários de redundância evidenciou a necessidade de ajustes em parâmetros *hardcoded* do Open5GS para otimização do tempo de recuperação da sessão PDU, dependendo da estratégia de recuperação adotada pelo operador da rede 5G. Em relação ao uso de recursos computacionais, demonstrou-se que o processo de *restart* da UPF pode gerar uma sobrecarga de CPU até 20 vezes superior à observada no cenário com redundância ativa-passiva.

A análise comparativa entre o *testbed* composto por gNB e UE reais e os ambientes emulados confirmou a possibilidade de equiparação dos resultados de *downtime*, indicando a viabilidade de extrapolação das métricas obtidas no ambiente

experimental para cenários escaláveis com múltiplos UEs. Adicionalmente, os experimentos com transmissão de vídeo demonstraram que, para aplicações multimídia, responsáveis por parcela significativa do tráfego global da Internet, a estratégia de reancoragem de sessões pode se tornar imperceptível ao usuário, dependendo do tamanho do *buffer* utilizado.

Como trabalhos futuros, planeja-se conduzir experimentos nas ilhas do projeto OpenRAN@Brasil, investigar estratégias preditivas para detecção antecipada de falhas em UPFs utilizando eBPF/XDP, permitindo ações antes da indisponibilidade efetiva da função, e incorporar mecanismos avançados de migração de sessões em ambientes multicluster, reproduzindo cenários heterogêneos de múltiplas bordas com compartilhamento de recursos computacionais.

REFERÊNCIAS

- [1] 3GPP, "System architecture for the 5G System (5GS)," Technical Specification TS 23.501 v18.4.0, 2024.
- [2] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.
- [3] F. Siegmund, R. Kundel, T. Meuser, and R. Steinmetz, "User plane performance in beyond 5g networks: Comprehensive analysis and evaluation," *Computer Communications*, vol. 247, p. 108397, 2026.
- [4] J. García-Morales, M. C. Lucas-Estañ, and J. Gozalvez, "Latency-Sensitive 5G RAN Slicing for Industry 4.0," *IEEE Access*, vol. 7, pp. 143 139–143 159, 2019.
- [5] E. Goshi, H. Harkous, S. Ahvar, R. Pries, F. Mehmeti, and W. Kellerer, "Joint upf and edge applications placement and routing in 5g & beyond," in *2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, 2024, pp. 446–452.
- [6] M. Wani, M. Kretschmer, B. Schröder, A. Grebe, and M. Rademacher, "Open RAN: A Concise Overview," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 13–28, 2024.
- [7] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, "Toward Next Generation Open Radio Access Networks: What O-RAN Can and Cannot Do!" *IEEE Network*, vol. 36, no. 6, pp. 206–213, 2022.
- [8] B. Agarwal, R. Irmer, D. Lister, and G.-M. Muntean, "Open ran for 6g networks: Architecture, use cases and open issues," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.
- [9] W. Mauricio, F. H. Costa Neto, M. Silva, R. K. Y. Aoki, E. Melão, S. Barros, F. Farias, L. Bondan, and F. M. Abinader, "Assessing nationwide distributed open ran deployments with the openran@brasil testbed," *IEEE Communications Magazine*, vol. 63, no. 2, pp. 46–52, 2025.
- [10] S. Paul, T. Keluskar, and M. Vutukuru, "A scalable and fault-tolerant 5g core on kubernetes," in *17th International Conference on Communication Systems and Networks (COMSNETS)*, 2025, pp. 658–666.
- [11] A. E. Ferguson, J. Larrea, and M. K. Marina, "CoreKube: An efficient, autoscaling and resilient mobile core system," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. Association for Computing Machinery*, 2023.
- [12] M. Terneborg, J. K. Rönnberg, and O. Schelén, "Application agnostic container migration and failover," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021, pp. 565–572.
- [13] K. Nguyen Trung and Y. Kim, "Design and implementation of a cost-effective failover mechanism for containerized UPF," *Electronics*, vol. 14, no. 15, p. 2991, 2025.
- [14] L. Wernet, L.-M. Spang, F. Siegmund, and T. Meuser, "Resilient user plane traffic redirection in cellular networks," in *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2024, pp. 1–6.
- [15] D. Soldani, P. Nahi, H. Bour, S. Jafarizadeh, M. F. Soliman, L. Di Giovanna, F. Monaco, G. Ognibene, and F. Risso, "ebpf: A new approach to cloud-native observability, networking and security for current (5g) and future mobile networks (6g and beyond)," *IEEE Access*, vol. 11, pp. 69 743–69 757, 2023.
- [16] G. Nunziati, C. Fiandrino, L. Foschini, and P. Bellavista, "Monitoring 5g core networks vulnerabilities with ebpf," *IEEE Networking Letters*, vol. 7, no. 3, pp. 220–224, 2025.
- [17] OpenRAN@Brasil, "OpenRAN@Brasil Program," 05 2025. [Online]. Available: <https://openranbrasil.org.br/>