UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Thayná Emilly Cavalcante Santos

Simulando a arquitetura L4S em redes 5G

Recife

2025

Thayná Emilly Cavalcante Santos

Simulando a arquitetura L4S em redes 5G

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Ciência da Computação.

Área de Concentração: Redes de Computadores

Orientador: Djamel H. Sadok

Recife

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Santos, Thayná Emilly Cavalcante.
Simulando a arquitetura L4S em redes 5G / Thayná Emilly Cavalcante Santos. - Recife, 2025.
36p. : il., tab.

Orientador(a): Djamel H. Sadok
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2025.
Inclui referências, anexos.

1. Redes de Computadores. 2. Redes Móveis. 3. Arquitetura de Rede Low Latency, Low Loss, Scalable Throughput (L4S). I. Sadok, Djamel H.. (Orientação). II. Título.

000 CDD (22.ed.)

Thayná Emilly Cavalcante Santos

Simulando a arquitetura L4S em redes 5G

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Ciência da Computação.

Aprovado em: 09 / 04 / 2025

BANCA EXAMINADORA

Prof. Dr. Djamel F. H. Sadok (Orientador)

Universidade Federal de Pernambuco

Prof. Dr. Renato M. Moraes (Examinador Interno)

Universidade Federal de Pernambuco

Simulando a arquitetura L4S em redes 5G

Thayná E. C. Santos[1], **Djamel F. H. Sadok**[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)

tecs@cin.ufpe.br, jamel@cin.ufpe.br

Abstract. *The evolution of 5G networks has enabled the development of latency-sensitive applications such as augmented reality (AR), virtual reality (VR), autonomous vehicles, and industrial remote control systems. However, the dynamic nature of the wireless medium and the unpredictability of congestion pose significant challenges to ensuring low latency and high transmission efficiency. In this context, the Low Latency, Low Loss, and Scalable Throughput (L4S) architecture emerges as a promising solution by providing real-time congestion notifications and allowing congestion control mechanisms at the end nodes to dynamically adjust the transmission rate. This study implements L4S in the downlink buffer of the Radio Link Control (RLC) layer in 5G networks, using the ns-3 simulator. The results demonstrate that L4S significantly reduces queueing delay in the RLC layer, particularly in high-demand scenarios with intense competition for radio resources, making it a viable alternative for mitigating queueing delays observed in real networks.*

Resumo. *A evolução das redes 5G tem viabilizado o desenvolvimento de aplicações altamente sensíveis à latência, como realidade aumentada (AR), realidade virtual (VR), veículos autônomos e sistemas de controle remoto industrial. No entanto, a natureza dinâmica do meio sem fio e a imprevisibilidade do congestionamento impõem desafios à garantia de baixa latência e alta eficiência na transmissão de dados. Nesse contexto, a arquitetura Low Latency, Low Loss, and Scalable Throughput (L4S) surge como uma solução promissora ao fornecer notificações em tempo real sobre o congestionamento e permitir que os mecanismos de controle de congestionamento nos nós finais ajustem dinamicamente a taxa de transmissão. Este estudo implementa a L4S na fila de saída (downlink) da camada RLC (Radio Link Control) em redes 5G, utilizando o simulador ns-3. Os resultados demonstram que a L4S reduz significativamente o atraso de fila dos fluxos de baixa latência na fila de saída da camada RLC, especialmente em cenários de alta demanda e competição por recursos de rádio, tornando-se uma alternativa viável para mitigar os atrasos de fila observados em redes reais.*

1. Introdução

A evolução das redes de telecomunicações tem sido impulsionada pela crescente demanda por conectividade de alta velocidade, baixa latência e maior eficiência no uso dos recursos de comunicação. Nesse cenário, o 5G surge como um marco transformador, prometendo não apenas taxas de transmissão mais altas, mas também a capacidade de suportar um ecossistema diversificado de dispositivos e aplicações com restrições severas de qualidade de serviço, como Internet das Coisas (IoT), redes veiculares, realidade aumentada

(AR) e realidade virtual (VR) (Ericsson, 2021). No entanto, à medida que essas aplicações, especialmente as que dependem de comunicação quase que instantânea, ganham popularidade, a latência torna-se um desafio crítico. Atrasos causados pelo congestionamento da rede podem comprometer a experiência do usuário e inviabilizar aplicações que exigem respostas imediatas. Assim, garantir um desempenho consistente e previsível é essencial para a adoção em larga escala dessas novas tecnologias.

Dentre as diversas fontes de atraso em redes, destaca-se o atraso por tempo de fila, no qual os pacotes, em cada nó, aguardam nas filas de entrada e saída antes de serem transmitidos ao próximo salto. Com o crescimento da internet, tornou-se evidente que o mecanismo de controle de congestionamento do TCP, embora poderoso, não era suficiente para garantir a qualidade de serviço em todas as circunstâncias. Isso revelou um limite no quanto os nós finais podem contribuir para o desempenho da rede. Assim, tornou-se necessário implementar mecanismos nos roteadores para complementar as estratégias de prevenção de congestionamento dos sistemas finais (Zhang et al., 1998).

Inicialmente, foi adotado nos roteadores um gerenciamento de filas conhecido como *Tail Drop*. Nesse esquema, cada fila possuía um tamanho máximo predefinido: enquanto houvesse espaço disponível, os pacotes eram aceitos e enfileirados. No entanto, quando a fila atingia sua capacidade máxima, novos pacotes eram descartados (dropados) até que houvesse espaço disponível, liberado pela transmissão de pacotes já enfileirados. Embora esse mecanismo tenha sido útil por muitos anos, ele apresentava falhas significativas, como a monopolização da fila por um ou poucos fluxos e a manutenção prolongada de filas lotadas. Isso ocorria porque a sinalização de congestionamento só acontecia quando os pacotes eram descartados por falta de espaço, retardando a resposta do sistema à sobrecarga (Zhang et al., 1998).

Na internet atual, o descarte de pacotes é um mecanismo essencial para notificar os sistemas finais sobre a ocorrência de congestionamento. Para resolver o problema das filas cheias, também conhecido como *bufferbloat*, é necessário que os sistemas finais sejam alertados antes mesmo da ocorrência da congestão, permitindo a redução proativa das taxas de transmissão. Nesse contexto, surgem os métodos de gerenciamento ativo de filas (*Active Queue Management* – AQM), que desempenham um papel fundamental não apenas na arquitetura L4S, mas em toda a infraestrutura de redes cabeadas modernas. Os AQMs atuam de forma proativa, regulando o tamanho das filas e limitando o tempo médio de permanência dos pacotes, evitando a formação de filas longas e os consequentes atrasos excessivos na entrega dos dados (Baker; Fairhurst, 2015).

Em redes cabeadas, para mitigar o problema do *bufferbloat* — que causa atrasos e degradação do desempenho, impactando especialmente aplicações sensíveis à latência (Pan et al., 2013) — foram desenvolvidos algoritmos como o *Dual Queue Coupled Active Queue Management* (DualPI2) e o *Controlled Delay* (CoDel). Esses mecanismos proporcionam um controle mais eficiente sobre o tempo de permanência dos pacotes na fila, reduzindo os impactos do congestionamento. No entanto, a crescente demanda por conectividade em redes sem fio revelou a necessidade de adaptar esses métodos para a borda da rede, onde o tráfego é mais dinâmico e sujeito a variações significativas, como no caso das redes 5G. Na borda das redes 5G, a importância dos AQMs se intensifica não somente por conta da coexistência de aplicações com requisitos distintos de latência e largura de banda, como também por conta das rápidas flutuações nas condições do canal.

Nesse cenário, a arquitetura L4S surge como uma solução promissora ao combinar AQMs com controles de congestionamento escaláveis. Essa abordagem permite que os nós finais realizem ajustes dinâmicos na taxa de transmissão, reduzindo os atrasos de fila e melhorando a qualidade de serviço (*Quality of Service* – QoS) para aplicações críticas que exigem latências ultrabaixas. A implementação da L4S no contexto das redes 5G representa, portanto, um avanço significativo na gestão do congestionamento e na otimização do desempenho da rede.

Dessa forma, este trabalho tem como objetivo avaliar o impacto da arquitetura L4S na redução da latência em redes 5G, com foco na fila de saída da camada de controle de acesso ao rádio (RLC) da RAN (Rede de Acesso ao Rádio). Para isso, o simulador ns-3 será utilizado como ferramenta de experimentação, possibilitando a análise de diferentes cenários de carga e congestionamento. Os resultados obtidos contribuirão para um entendimento mais aprofundado dos benefícios e limitações da arquitetura L4S, além de fornecer subsídios para a otimização de parâmetros e sua futura implementação em redes reais.

A relevância deste estudo reside na crescente necessidade de soluções que garantam latência ultrabaixa em ambientes dinâmicos e de alta demanda. Com a expansão das aplicações de tempo real, como realidade aumentada, cirurgia remota e veículos autônomos, a avaliação da viabilidade da L4S em redes 5G não apenas fortalece a fundamentação teórica da área, mas também pode influenciar futuras decisões na padronização e implementação de soluções de gerenciamento de congestionamento, promovendo redes mais eficientes e adaptáveis aos desafios das próximas décadas.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica, abordando conceitos sobre redes 5G, a arquitetura L4S, as camadas da RAN e o simulador de redes ns-3; a Seção 3 detalha o desenvolvimento do projeto e seus desafios, abordando as mudanças de código realizadas, o ambiente de testes e as métricas de avaliação utilizadas; a Seção 4 apresenta os resultados e discute as implicações dos dados obtidos; por fim, as seções de trabalhos relacionados, conclusão e trabalhos futuros encerram o trabalho, sintetizando as contribuições e apontando direções para pesquisas subsequentes.

2. Fundamentação Teórica

Nessa seção serão abordados os conceitos técnicos necessários para o bom entendimento do projeto, como redes 5G, arquitetura L4S e as camadas da RAN.

2.1. Redes de Telecomunicações e o 5G

As redes de quinta geração (5G) representam uma evolução significativa em relação às gerações anteriores, não apenas em termos de velocidade, mas também na capacidade de suportar uma ampla variedade de aplicações e serviços com requisitos distintos. A visão do 5G é baseada em três pilares principais (Verizon, 2023):

(i) **eMBB (*Enhanced Mobile Broadband*)**: Projetado para oferecer altas taxas de transmissão de dados, o eMBB é ideal para aplicações como *streaming* de vídeo em alta definição, realidade aumentada (AR) e realidade virtual (VR). Essas aplicações exigem grande largura de banda e latência moderada.

(ii) **mMTC (*Massive Machine-Type Communications*)**: Voltado para a conectividade massiva de dispositivos IoT, o mMTC suporta um grande número de dispositivos conectados simultaneamente, com requisitos de baixa taxa de transmissão de dados e longa duração da bateria. Aplicações típicas incluem smart cities, agricultura inteligente e monitoramento industrial.

(iii) **URLLC (*Ultra-Reliable Low-Latency Communications*)**: Focado em aplicações críticas que exigem latência ultrabaixa e altíssima confiabilidade, como cirurgias remotas, controle de veículos autônomos e automação industrial. O URLLC é essencial para garantir respostas em tempo real e operações seguras.

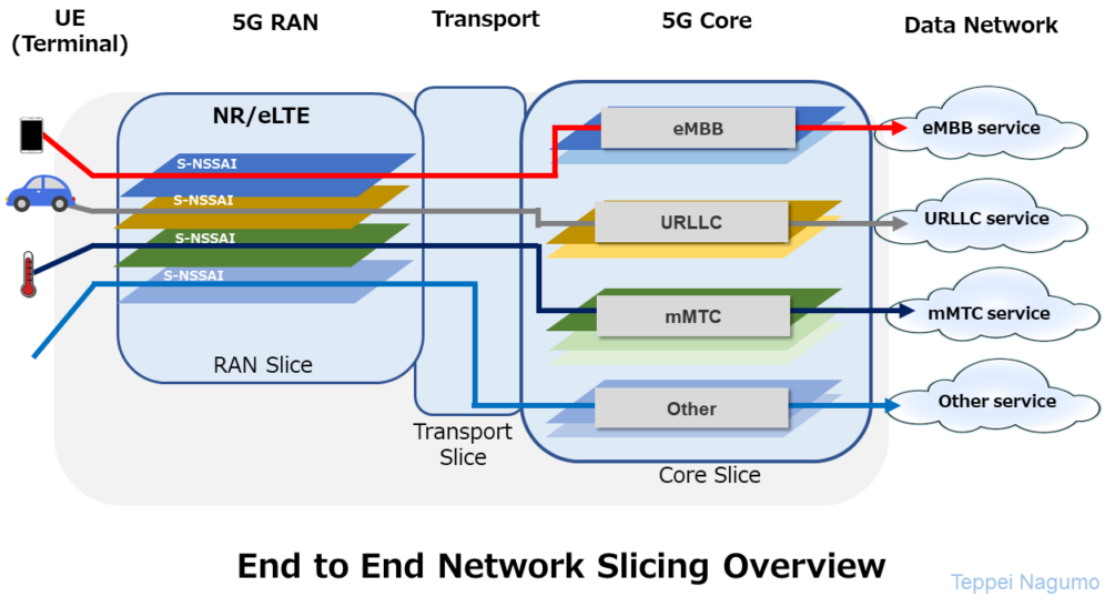

Figura 1. Visão geral do "fatiamento"das redes 5G.

Fonte: Blog Teppei Log - **Matéria:** *What is 5G? Differences between 4G and 5G characteristic (eMBB, URLLC, mMTC).*

Como é possível visualizar na Figura 1, para que esses pilares sejam possíveis na prática, o 5G emprega a técnica de *network slicing*. Em mais detalhes, o *network slicing* é uma das tecnologias fundamentais do 5G, permitindo a criação de redes virtuais independentes sobre uma mesma infraestrutura física. Cada *fatia* pode ser configurada com parâmetros específicos de QoS, segurança e desempenho, atendendo às necessidades de diferentes aplicações (3GPP, 2023). Por exemplo, uma fatia pode ser dedicada ao serviço eMBB para streaming de vídeo, enquanto outra é configurada para URLLC, garantindo latência ultrabaixa para aplicações industriais críticas.

Além disso, o 5G introduz um modelo avançado de QoS, onde cada fluxo de dados é associado a um *QoS Flow Identifier* (QFI). Esses fluxos são diferenciados através de vários parâmetros, dentre os quais podemos citar a taxa de transmissão garantida (GBR - *Guaranteed Bit Rate*), sua prioridade e a latência máxima permitida, que especifica o tempo máximo que um pacote pode levar para ser entregue.

Com isso, podemos perceber que as redes 5G foram projetadas para suportar uma ampla gama de aplicações especializadas, oferecendo garantias rigorosas de qualidade

de serviço. De acordo com (GSMA, 2022), espera-se que, até o final de 2025, o 5G represente 25% de todas as conexões móveis globalmente, um aumento significativo em relação aos 8% registrados em 2021. Esse cenário ressalta a importância de mecanismos eficientes de gerenciamento de congestionamento e acesso à rede, como o AQM, para lidar com a crescente demanda por conectividade e garantir a qualidade de serviço em um ambiente de tráfego cada vez mais diversificado e intenso.

2.2. Arquitetura L4S: Conceitos e Funcionamento

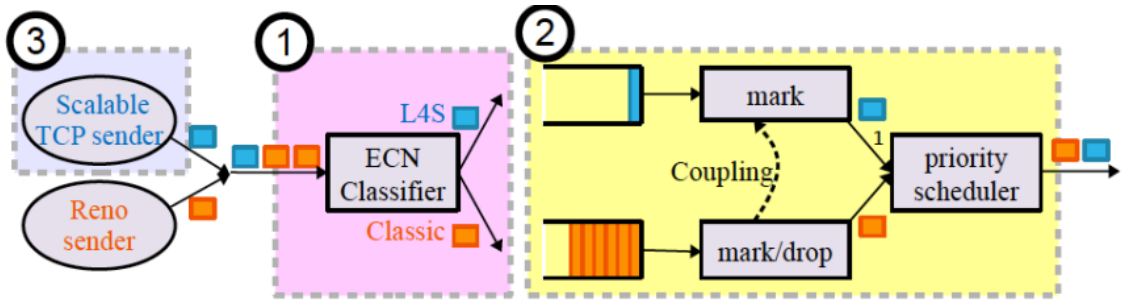

Figura 2. Arquitetura L4S

Fonte: Apresentação *L4S Architecture Low Latency, Low Loss, Scalable Throughput Internet Service*, por Bob Briscoe.

A arquitetura L4S foi desenvolvida para atender às exigências de aplicações sensíveis à latência, proporcionando tempos de fila reduzidos, minimizando perdas de pacotes por congestionamento e otimizando a utilização do enlace através de mecanismos de controle de congestionamento escaláveis (Briscoe et al., 2023). Diferentemente das abordagens convencionais, o L4S parte do princípio de que o atraso de fila resulta, em grande parte, dos algoritmos de controle de congestionamento clássicos, como TCP Reno e TCP Cubic, que priorizam a maximização da ocupação do meio sem considerar adequadamente o impacto do acúmulo excessivo de pacotes nas filas ao longo da rede.

No paradigma L4S, a redução da latência não é simplesmente um benefício derivado da infraestrutura física, mas uma consequência direta do comportamento preditivo e eficiente dos controladores de congestionamento escaláveis utilizados pelos emissores compatíveis com L4S. O papel fundamental da rede nesta arquitetura é garantir o isolamento do tráfego L4S que opera de maneira controlada e disciplinada, através de uma fila dedicada em cada salto na rede como ilustrado na cor azul na Figura 2, em relação ao tráfego "clássico", que continua dependendo de estratégias tradicionais de enfileiramento que introduzem atrasos significativos (Briscoe et al., 2023).

A implementação do L4S representa um avanço significativo no controle de congestionamento, permitindo a transição de algoritmos tradicionais — que sofrem com a sincronização de reações a perdas e a progressão lenta da janela de congestionamento — para uma nova geração de mecanismos escaláveis. Estes novos algoritmos utilizam uma versão melhorada do protocolo de Notificação Explícita de Congestionamento (*Explicit Congestion Notification* – ECN), possibilitando que a rede sinalize proativamente o congestionamento iminente, antes mesmo da ocorrência de perdas de pacotes, resultando em ajustes mais frequentes e mais precisos na taxa de transmissão.

Código Binário	Nome	Descrição
00	Non-ECT	Transporte não compatível com ECN
01	ECT (1)	Transporte compatível com o L4S
10	ECT (0)	Transporte compatível com ECN
11	CE	Congestionamento Experienciado

Figura 3. Descrição dos bits ECN

Em detalhes, como ilustrado na Figura 3, os 2 bits reservados no cabeçalho IPv4 para o protocolo ECN possuem os seguintes significados: Not-ECT (não compatível com ECN), ECT(0), ECT(1) (ambos indicando que o transporte é compatível com ECN) e CE (Congestion Experienced, ou Congestionamento Detectado). Um pacote marcado com o código CE é considerado *ECN-marked*, ou simplesmente marcado (Schepper; Briscoe, 2023).

O ECN é um mecanismo que permite à rede sinalizar congestionamento iminente aos sistemas finais sem necessidade de descartar pacotes. Quando um roteador detecta os primeiros sinais de congestionamento, ele marca o cabeçalho do pacote com um sinal de CE, em vez de descartá-lo. O receptor, ao identificar essa marcação, notifica o remetente sobre o congestionamento através do bit ECE (ECN-Echo) no cabeçalho TCP.

Com essa abordagem, um controle de congestionamento escalável reduz sua janela de congestionamento proporcionalmente à fração de pacotes marcados com CE, de maneira significativamente menos disruptiva do que ocorreria com perdas de pacotes (Briscoe et al., 2023). Assim, com notificações mais frequentes e precisas sobre o estado de congestionamento da rede, os controles de congestionamento escaláveis ajustam dinamicamente suas taxas de transmissão, mantendo a latência de fila em níveis mínimos e maximizando a utilização do enlace, resultando em uma experiência substancialmente mais eficiente e responsiva para os usuários. Esta abordagem não apenas minimiza a latência, mas também otimiza a eficiência global da rede, posicionando o L4S como uma solução altamente promissora para aplicações sensíveis a atrasos, incluindo streaming de alta definição, videoconferência interativa e jogos online em tempo real.

Em termos de implementação, como demonstrado na Figura 2, o L4S emprega algoritmos de controle de congestionamento escaláveis, como o *Data Center* TCP (DCTCP) e o TCP Prague. Estes protocolos adotam uma abordagem mais assertiva na busca por capacidade disponível, enquanto reduzem drasticamente a latência de enfileiramento ao reagir com precisão às marcações de congestionamento nos cabeçalhos dos pacotes. Para garantir a coexistência eficiente entre fluxos L4S e fluxos clássicos, a arquitetura L4S implementa o mecanismo de gerenciamento ativo de fila DualPI2. O DualPI2 isola o tráfego L4S em uma fila de baixa latência, enquanto mantém um acoplamento estratégico com a fila clássica para assegurar justiça na alocação de recursos e distribuição equilibrada da capacidade de banda(Briscoe et al., 2023).

A preservação dos fluxos clássicos faz parte de um aspecto fundamental e estratégico do L4S que é a sua possibilidade de implantação incremental na rede, que permite a coexistência entre os novos controles de congestionamento escaláveis (L4S) e os tradi-

cionais, clássicos, em uma infraestrutura de rede compartilhada. O objetivo central é que o L4S proporcione aos usuários uma latência substancialmente menor e vazão (throughput) significativamente melhor (e raramente piores), sem comprometer o desempenho dos controles clássicos existentes.

Adicionalmente, experimentos conduzidos pela Ericsson evidenciam que o L4S é particularmente adequado para implementação em redes 5G, onde a coexistência de diversos tipos de tráfego e a variabilidade intrínseca das condições do canal exigem soluções adaptativas e altamente eficientes (Ericsson, 2021). A capacidade do L4S de adaptar-se rapidamente às mudanças dinâmicas nas condições da rede, mantendo consistentemente baixa latência e elevada utilização dos recursos, consolida-o como uma solução promissora para aplicações sensíveis ao atraso em ambientes de rede heterogêneos e altamente dinâmicos como os encontrados nas implementações de última geração.

2.3. Rede de Acesso ao Rádio

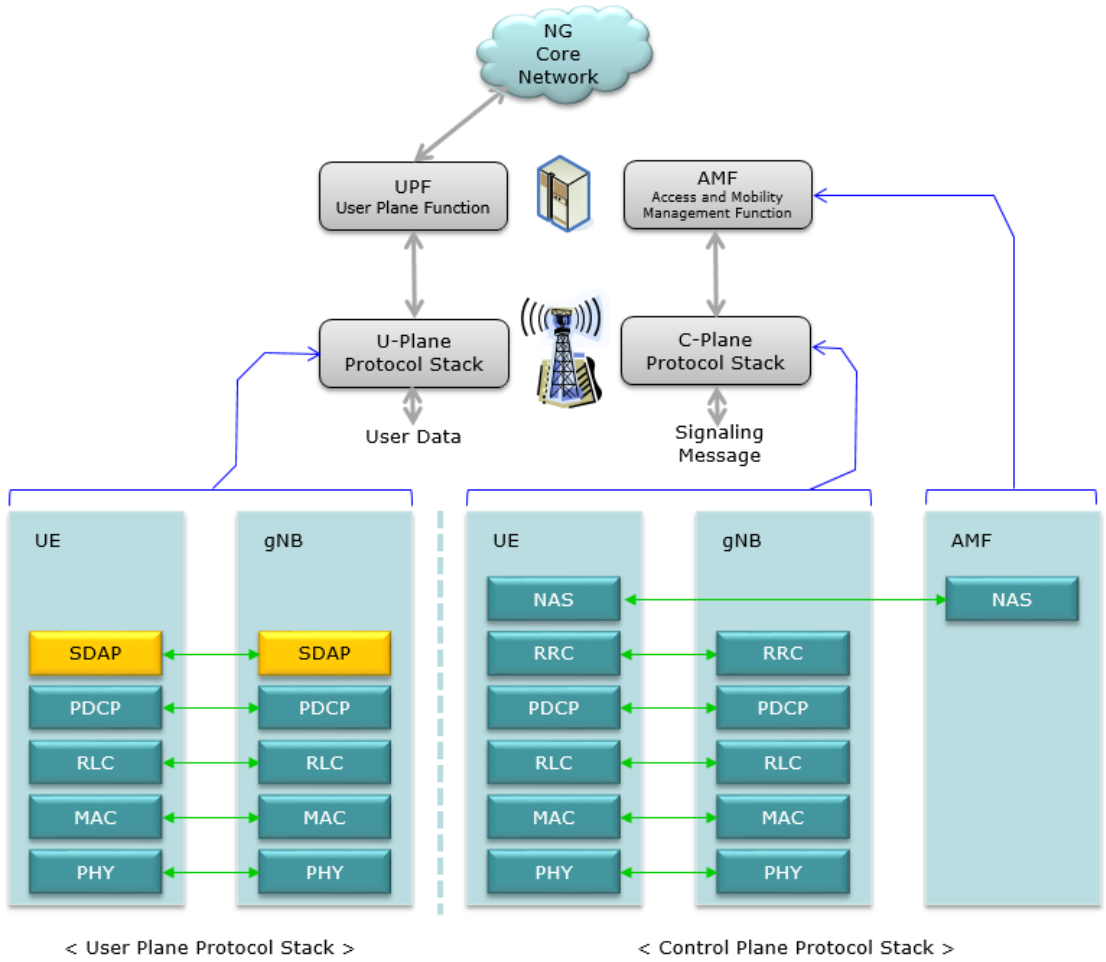

Figura 4. Pilha de protocolos da rede de acesso ao rádio

Fonte: Blog ShareTechnote - **Matéria:** *5G/NR - Radio Protocol Stack Architecture*.

A Rede de Acesso por Rádio (RAN - *Radio Access Network*) no 5G é um dos componentes mais críticos das redes móveis modernas, sendo responsável por gerenciar a interface

de rádio entre os dispositivos dos usuários (UE - *User Equipment*) e a rede central (Core Network - CN). A RAN desempenha um papel essencial na otimização da alocação de espectro, na gestão eficiente de interferências e na garantia de qualidade de serviço (QoS), fatores essenciais para suportar aplicações emergentes como veículos autônomos, realidade aumentada e cirurgias remotas (Dahlman; Parkvall; Sköld, 2020; 3GPP, 2019).

A arquitetura da RAN no 5G evoluiu significativamente em relação às redes móveis anteriores, permitindo uma maior flexibilidade na alocação de recursos de rádio, suportando o conceito de *network slicing* e reduzindo a latência por meio de novas topologias de rede, como a Centralized-RAN (C-RAN).

A pilha de protocolos da interface de rádio (*Radio Protocol Stack*), ilustrada na Figura 4, é dividida em dois planos principais:

(i) **Plano de Usuário (U-Plane)**: Responsável pelo transporte dos pacotes de dados das aplicações, garantindo a transmissão eficiente de serviços como streaming de vídeo, chamadas de voz sobre IP (VoIP) e navegação na internet.
(ii) **Plano de Controle (C-Plane)**: Transporta mensagens de sinalização e controle para estabelecer, manter e liberar conexões de rede, além de gerenciar a mobilidade do dispositivo e a seleção de células (3GPP, 2019).

A arquitetura da RAN no 5G também é organizada em camadas funcionais que garantem a comunicação eficiente entre os dispositivos e a rede central:

- **Camada Física (PHY)**: Responsável pela transmissão e recepção dos sinais de rádio. Utiliza técnicas avançadas como MIMO massivo e ondas milimétricas para maximizar a eficiência espectral (Dahlman; Parkvall; Sköld, 2020).
- **Controle de Acesso ao Meio (MAC)**: Gerencia o acesso ao meio físico, alocando dinamicamente recursos de rádio e lidando com retransmissões em caso de erros na transmissão (3GPP, 2020a).
- **Controle do Enlace de Rádio (RLC)**: Implementa técnicas de retransmissão e controle de erro para garantir a entrega confiável dos pacotes de dados (3GPP, 2020c).
- **Protocolo de Convergência de Dados de Pacote (PDCP)**: Responsável pela criptografia, compressão de cabeçalhos e controle de integridade dos dados transmitidos (3GPP, 2020b).
- **Adaptador de Dados de Fluxo de Serviço (SDAP)**: Introduzido no 5G, o SDAP assegura a correta associação dos fluxos de dados aos diferentes níveis de QoS, garantindo que os requisitos específicos de cada aplicação sejam atendidos (3GPP, 2020d).
- **Controle de Recursos de Rádio (RRC)**: Gerencia a configuração das conexões de rádio, a mobilidade dos dispositivos e a sinalização entre o UE e a rede, sendo essencial para a operação eficiente da RAN (3GPP, 2020d).

Já a camada NAS (*Non-Access Stratum*), entre o UE e o AMF (*Access and Mobility Management Function*), do 5G é responsável pela sinalização entre o equipamento do usuário e o core da rede. Ou seja, como ilustrado na Figura 4, ela não passa pela camada de acesso (gNB), mas sim diretamente entre o UE e o AMF. Atuando como um protocolo de controle, ela permite ao UE se registrar na rede, manter sua conectividade e mobilidade, além de garantir que a comunicação seja segura e autenticada.

Na RAN do 5G, as camadas RLC e PDCP podem possuir múltiplas instâncias para suportar diferentes portadoras (*bearers*), cada uma projetada para atender requisitos específicos de confiabilidade, latência e prioridade de transmissão. Isso possibilita a entrega diferenciada de serviços críticos, como comunicação entre dispositivos IoT de baixa latência ou *streaming* de conteúdo em alta qualidade.

2.4. Camada RLC

Neste estudo, implementamos o gerenciador de fila DualPI2, componente da arquitetura L4S, na camada RLC-UM (*Unacknowledged Mode*) da RAN do 5G, especificamente no *downlink*. Esta subseção detalha o funcionamento desta camada e contextualiza nossa implementação.

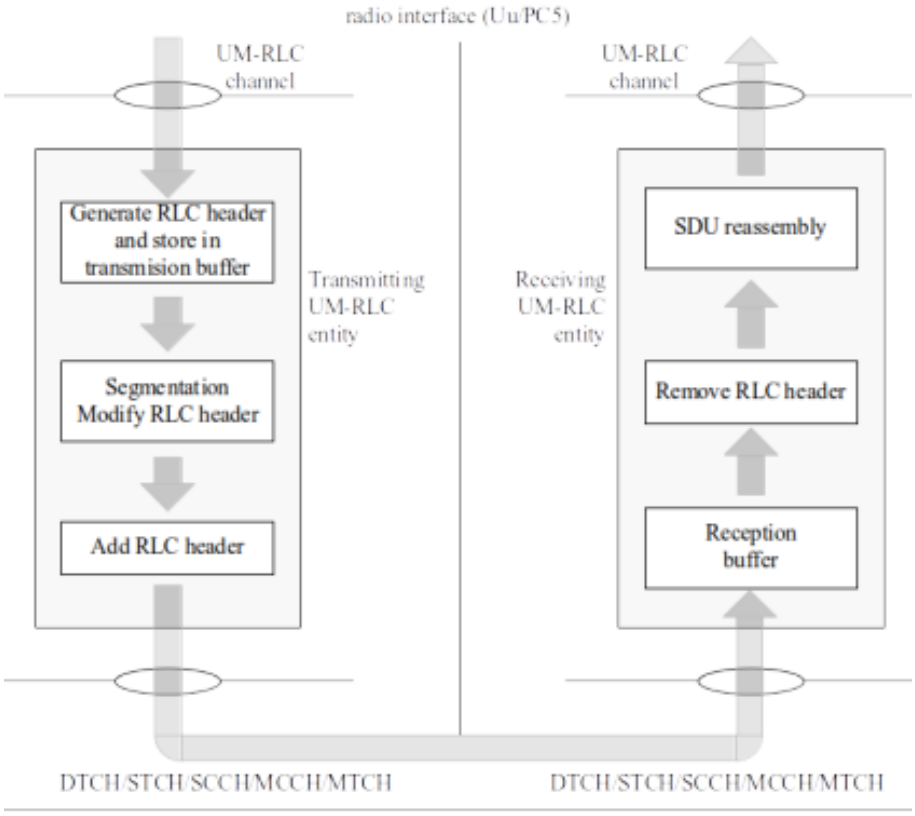

Figura 5. Arquitetura funcional da camada RLC-UM

Fonte: *Radio Link Control (RLC) protocol specification* (3GPP, 2020c).

A camada RLC-UM é responsável pela transmissão e recepção de dados sem garantia de entrega, sendo particularmente adequada para aplicações que toleram perdas moderadas de pacotes, como streaming de vídeo, áudio e voz sobre IP (VoIP). A Figura 5 ilustra sua arquitetura funcional.

Na entidade transmissora da RLC-UM (*transmitting UM RLC entity*), os RLC SDUs (*Service Data Units*) recebidos da camada superior são armazenados na fila de transmissão. A entidade então gera PDUs (*Protocol Data Units*) do tipo UMD (*Unacknowledged Mode Data*) para cada RLC SDU, anexando os cabeçalhos RLC apropriados. Quando a camada inferior sinaliza uma oportunidade de transmissão, a entidade RLC-UM

realiza a segmentação dos RLC SDUs conforme necessário, ajustando os UMD PDUs resultantes, com seus respectivos cabeçalhos, ao tamanho total ("créditos") indicado pela camada inferior (3GPP, 2020c).

A entidade receptora da RLC-UM (*receiving UM RLC entity*), por sua vez, desempenha as seguintes funções principais:

- Detectar a perda de segmentos de RLC SDU nas camadas inferiores;
- Reconstruir os RLC SDUs a partir dos UMD PDUs recebidos e entregá-los à camada superior assim que estiverem completos;
- Descartar UMD PDUs que não podem ser reconstruídos em um RLC SDU completo devido à perda de segmentos pertencentes ao mesmo RLC SDU (3GPP, 2020c).

Além disso, é importante contextualizar que, além do modo UM, a especificação do protocolo RLC define dois outros modos de operação:

O modo TM (*Transparent Mode*) é utilizado para transmissões que não requerem processamento adicional pela camada RLC, como em canais de controle ou transmissões de broadcast. No modo TM, os dados são transferidos diretamente entre as camadas adjacentes, sem adição de cabeçalhos, segmentação ou concatenação (3GPP, 2020c).

Já o modo AM (*Acknowledged Mode*) é empregado em transmissões que exigem alta confiabilidade, como dados de controle ou aplicações sensíveis a perdas. No modo AM, a entidade RLC implementa sofisticados mecanismos de retransmissão, controle de fluxo e reordenação para garantir a entrega correta e sequencial dos pacotes (3GPP, 2020c).

A escolha do modo UM para implementação do DualPI2 em nosso estudo, assim como em (Dai et al., 2017), deve-se à preservação das mensagens de controle e sinalização que transitam na rede através dos outros modos da camada.

2.5. Simulação de Redes com o ns-3

O ns-3 é um simulador de redes de computadores baseado em eventos discretos, amplamente utilizado na pesquisa acadêmica e em projetos de desenvolvimento e validação de protocolos e arquiteturas de redes. Ele é um software de código aberto, escrito principalmente em C++, e é voltado para simulações que abrangem desde redes cabeadas até redes sem fio, incluindo tecnologias como Wi-Fi, LTE e 5G NR (ns-3 Consortium, 2023).

O funcionamento do ns-3 se baseia em uma linha do tempo onde eventos são agendados e processados em sequência, em vez de rodar em tempo real. Isso permite que o simulador avance de um evento ao próximo sem desperdiçar tempo computacional em períodos de inatividade. Esse método é conhecido como **simulação discreta por eventos** (ns-3 Consortium, 2023).

Em sua arquitetura, o ns-3 é estruturado em nós (nodes), que representam dispositivos de rede como computadores, celulares ou roteadores. Esses nós podem ser configurados com aplicações que geram e consomem dados, como aplicações TCP, UDP, FTP, entre outras. Além disso, cada nó possui uma pilha de protocolos composta pelas camadas de aplicação, transporte (TCP, UDP), rede (IP, roteamento), enlace (MAC), e física (PHY), permitindo que o comportamento de cada camada do modelo TCP/IP seja simulado.

Os nós se comunicam por meio de dispositivos de rede (NetDevices), que simulam interfaces como Wi-Fi, LTE ou Ethernet, e esses dispositivos são conectados através de canais (Channels), que representam os meios físicos de transmissão, como cabos ou enlaces de rádio.

Um dos pontos fortes do ns-3 é a disponibilidade de modelos avançados de tecnologias modernas, como redes LTE e 5G NR, além da simulação detalhada de redes Wi-Fi, redes cabeadas e até redes de sensores. Também há suporte para simulações envolvendo mobilidade, ou seja, os nós podem se deslocar conforme trajetórias definidas, simulando redes móveis e redes veiculares (VANETs). Outro aspecto importante é a presença de modelos de propagação de rádio, que permitem representar cenários urbanos, rurais ou ambientes internos, afetando a qualidade do sinal entre os dispositivos.

Além disso, uma das grandes vantagens de utilizar o ns-3 é que ele é um simulador de redes de código aberto, ou seja, seu código-fonte está disponível publicamente e pode ser livremente modificado e estendido pelos usuários. Isso permite que pesquisadores e desenvolvedores adicionem novas funcionalidades, implementem protocolos personalizados e adaptem o simulador às suas necessidades específicas, como é o desse estudo, onde o DualPI2 foi adicionado à pilha de protocolos da RAN do 5G.

Por fim, o ns-3 conta com uma comunidade ativa composta por pesquisadores e engenheiros, além de uma documentação bastante completa, que inclui tutoriais e exemplos práticos. Isso faz com que o ns-3 seja uma ferramenta robusta e bastante utilizada tanto no meio acadêmico quanto na indústria para experimentação e desenvolvimento de novas soluções em redes de computadores.

3. Desenvolvimento e Implementação

3.1. Arquitetura L4S na RAN

A implementação do algoritmo DualPI2 na RAN do 5G apresenta vantagens significativas quando comparada a uma implementação no core da rede. Uma das principais razões para essa escolha é a necessidade de uma resposta ágil às variações dinâmicas das condições do canal na RAN. Como a interface rádio é altamente variável devido a fatores como mobilidade dos usuários, interferências e alocação dinâmica de espectro, a aplicação do gerenciamento ativo de filas diretamente na estação base (gNB) permite um controle mais responsivo e preciso sobre o congestionamento da rede.

Além disso, essa implementação também possibilita que as decisões de controle de congestionamento sejam tomadas mais próximas da origem da variabilidade, reduzindo a latência e minimizando o impacto do *bufferbloat*. Como resultado, há um controle mais eficiente do atraso na fila e da perda de pacotes, beneficiando aplicações sensíveis ao atraso, que exigem uma latência baixa e ultraconfiável (URLLC) para manterem a qualidade do serviço. Além disso, essa abordagem permite uma melhor coordenação com mecanismos de controle da camada física e da camada MAC, otimizando a alocação de recursos de rádio em tempo real.

Outra alternativa seria a implementação do AQM no *User Plane Function* (UPF), que atua no plano de dados do Core da rede. No entanto, essa abordagem apresenta desafios consideráveis, principalmente devido à distância entre a RAN e o Core. O tempo necessário para que informações sobre o estado do rádio sejam propagadas até o UPF

e, posteriormente, processadas para ajuste do congestionamento pode resultar em uma reação tardia, reduzindo a eficácia do controle de filas. Em um ambiente de redes móveis altamente dinâmico, essa latência adicional compromete o desempenho das aplicações que exigem respostas em tempo quase real.

3.2. Integração da Arquitetura L4S

Esta subseção apresenta uma visão geral das principais alterações realizadas no código, abrangendo modificações nas camadas RAN e MAC, configuração do ambiente de testes e métricas de avaliação do experimento.

Como o ns-3 não oferece suporte nativo à arquitetura L4S, utilizamos como base o repositório ns3-tcp-prague[1]. Este repositório implementa uma versão inicial do protocolo TCP Prague e uma implementação preliminar do gerenciador de filas DualPI2, desenvolvida por Shravya em 2017. Apesar de ter sido criada há alguns anos, esta implementação continua sendo uma referência importante na área, sendo adotada inclusive por organizações como a CableLabs.

O TCP Prague, um controle de congestão escalável, representa uma implementação de referência para a arquitetura L4S (Briscoe et al., 2023). Derivado do DCTCP, ele possui a capacidade de identificar enlaces de saída *clássicos* e enlaces sem suporte ao ECN, adaptando automaticamente seu comportamento para funcionar como um controle de congestão clássico (semelhante ao TCP Cubic ou TCP Reno). Essa característica possibilita a coexistência com tráfegos *clássicos* na rede (Schepper; Tilmans et al., 2024). O TCP Prague também se destaca por reagir rapidamente aos sinais ECN, mantendo alta eficiência mesmo em redes de alta velocidade. Como algoritmo de controle de congestão escalável, seu tráfego é direcionado para filas de baixa latência (representada pela fila azul na Figura 2), quando estas estão disponíveis na infraestrutura de rede.

Devido à incompatibilidade com a versão 3.42 do ns-3, foram necessárias diversas adaptações no código original, incluindo a atualização de APIs obsoletas, reestruturação do gerenciador de filas DualPI2 e adição de novas funcionalidades. As mudanças mais relevantes são detalhadas na próxima subseção e estão disponíveis no repositório do projeto no GitHub[2].

[1] https://github.com/shravya-ks/ns-3-tcp-prague
[2] https://github.com/tecs2000/ns3-dualpi2

3.3. Principais Modificações

3.3.1. Redesign do Cabeçalho PDCP

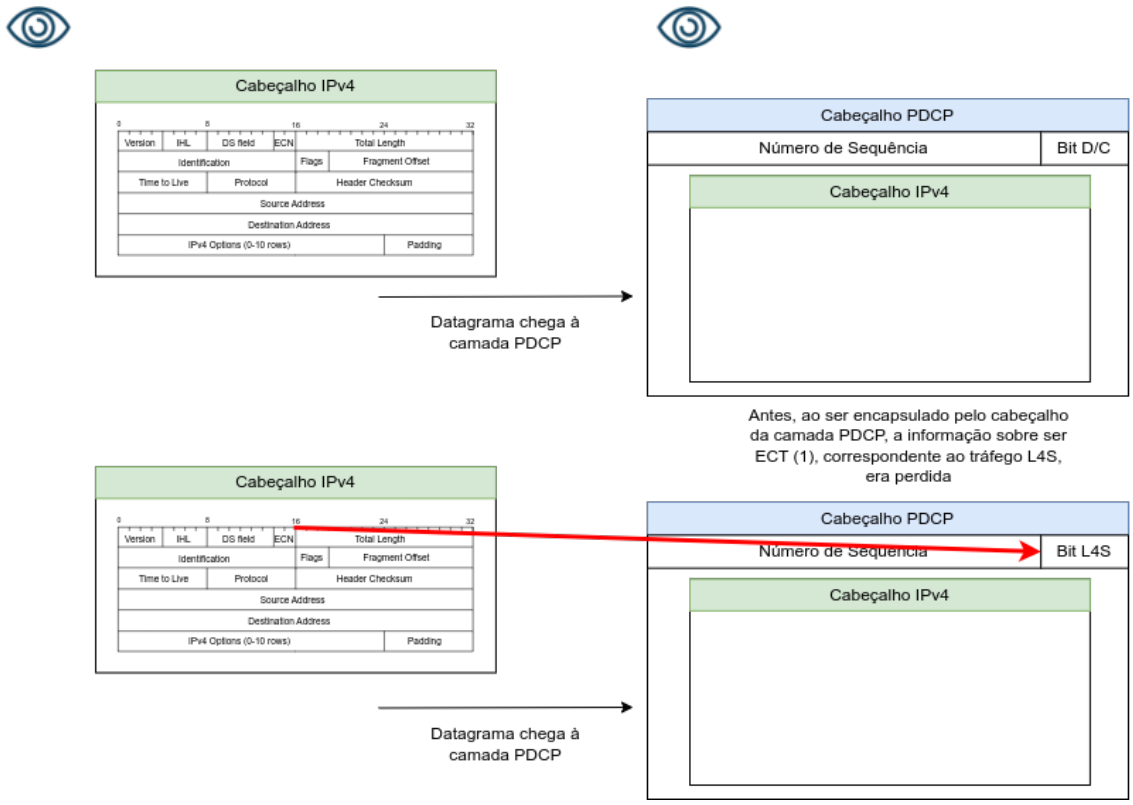

Figura 6. Procedimento de atualização do cabeçalho PDCP

Para permitir que a camada RLC diferencie pacotes entre clássicos e escaláveis com base nos bits ECT do cabeçalho IPv4, o bit D/C do cabeçalho PDCP foi redefinido para indicar se o pacote é ECT (1), referente a um fluxo escalável. Essa modificação, vista na Figura 6, contornou uma limitação do ns-3, onde só é possível consultar o último cabeçalho adicionado ao pacote. Como é possível conferir na Figura 4, no downlink, a camada PDCP vem imediatamente antes da camada RLC. Assim, o cabeçalho PDCP adicionado ao pacote sobrepõe-se ao cabeçalho IPv4, fazendo com que este não possa ser acessado na camada RLC. Essa informação foi confirmada por Tom Henderson, um dos principais desenvolvedores do ns-3, no fórum 5G-LENA.

3.3.2. Atualização e Validação do Código

Os arquivos referentes ao DualPI2 foram revisados e atualizados para que fossem compatíveis com a versão 3.42 do ns-3. A suíte de testes *dual-q-coupled-pi-square-queue-disc-test-suite.cc* também foi adaptada para validar as novas funcionalidades.

3.3.3. Novos Itens de Fila

```
class DualQueueL4SQueueDiscItem : public QueueDiscItem
{
public:
  DualQueueL4SQueueDiscItem (Ptr<Packet> p, const Address &
      addr, uint16_t protocol);
  ~DualQueueL4SQueueDiscItem ();
  void AddHeader (void) override;
  bool Mark (void) override;
  bool IsL4S (void) override;
  ...
};

class DualQueueClassicQueueDiscItem : public QueueDiscItem
{
public:
  DualQueueClassicQueueDiscItem (Ptr<Packet> p, const
     Address & addr, uint16_t protocol);
  ~DualQueueClassicQueueDiscItem ();
  void AddHeader (void) override;
  bool Mark (void) override;
  bool IsL4S (void) override;
  ...
};
```

Algoritmo 1. Definição das classes *DualQueueL4SQueueDiscItem* e *DualQueueClassicQueueDiscItem*.

Como ilustrado no Algoritmo 1, foram criados os tipos *DualQueueL4SQueueDiscItem* e *DualQueueClassicQueueDiscItem* para diferenciar o tráfego escalável (L4S) do clássico no AQM. No ns-3, as filas armazenam objetos do tipo `QueueDiscItem`, que encapsulam pacotes e metadados essenciais, como o tempo de chegada na fila. Sem essas representações, não era possível enfileirar os pacotes no AQM.

3.3.4. Representação do Tamanho da Fila

Para compatibilizar a tomada de decisão da camada RLC-UM, uma função foi adicionada à classe DualQCoupledPiSquareQueueDisc para armazenar seu tamanho em bytes, independentemente de estar configurada para representar suas métricas em número de pacotes, e que atualiza o seu valor a cada novo enfileiramento e retirada de pacote das filas.

3.3.5. Ajustes na Classe `QueueDisc`

```

void
QueueDisc::Drop(Ptr<QueueDiscItem> item, char* reason)
```

```
{
    ...
    m_nPackets--;
    m_nBytes -= item->GetSize ();
    m_stats.nTotalDroppedPackets++;
    m_stats.nTotalDroppedBytes += item->GetSize();
    ...
}
```

Algoritmo 2. Método `Drop` antes das alterações.

```

Ptr<QueueDiscItem>
DualQCoupledPiSquareQueueDisc::DoDequeue ()
{
   ...
   Ptr<QueueDiscItem> p = GetInternalQueue(0)->Dequeue();

   if(m_classicDropProb / (m_k * 1.0) > m_uv->GetValue()) {

       if (!p->Mark ()) {
          Drop (p, "Drops due to drop probability");
       }
       ...
   }
   ...
}
```

Algoritmo 3. Chamada ao método `Drop` pelo AQM.

Como ilustrado no Algoritmo 2, a função de `Drop`, antes dos ajustes, subtraía das informações da fila o pacote em bytes, e em número de pacotes. Porém, como ilustrado no Algoritmo 3, o método de `Drop` só era chamado após o método `Dequeue`, de modo que essas métricas eram modificadas duas vezes. Essa inconsistência além de contaminar os resultados do AQM, também fazia o experimento falhar em vários asserts ao longo da simulação.

```

void
QueueDisc::Drop(Ptr<QueueDiscItem> item, char* reason)
{
    ...
    m_stats.nTotalDroppedPackets++;
    m_stats.nTotalDroppedBytes += item->GetSize();
    ...
}
```

Algoritmo 4. Método `Drop` após das alterações.

Assim, a função foi corrigida para atualizar corretamente as métricas da fila, Algoritmo 4, evitando contagens duplicadas de perda de pacote e de bytes armazenados.

A outra alteração feita na classe foi no método `Requeue`, que antes era privado e foi tornado público para permitir o rearmazenamento de pacotes segmentados pela RLC que não puderam ser enviados à camada MAC por falta de créditos.

3.3.6. Acesso ao CQI na Camada RLC-UM

Agora, além da camada MAC, as camadas RLC e PDCP também podem acessar os dados de qualidade do canal (CQI) reportados pelos UEs. Conforme mencionado anteriormente, a implementação de um AQM na gNB permite melhor coordenação com os mecanismos de controle das camadas física e MAC, uma vez que as métricas de qualidade do canal podem ser incorporadas ao AQM na avaliação do estado da fila. Isso possibilita decisões mais precisas com latência extremamente baixa, devido à proximidade física entre os usuários e a rede.

```
1
2  void
3  NrGnbMac::DoReceiveControlMessage(Ptr<NrControlMessage> msg
       )
4  {
5      ...
6      case (NrControlMessage::DL_CQI): {
7          Ptr<NrDlCqiMessage> cqi = DynamicCast<
               NrDlCqiMessage>(msg);
8          DlCqiInfo cqiElement = cqi->GetDlCqi();
9          NS_ASSERT(cqiElement.m_rnti != 0);
10         m_dlCqiReceived.push_back(cqiElement);
11
12         /**
13          * Pipe the info up to the RLC Layer.
14          *
15          * After observation, was noticed that the vector
                only contains data in its first position.
16          */
17         auto rnti = cqiElement.m_rnti;
18         auto outerIt = m_rlcAttached.find(rnti); // returns
               every RLC instance associated to this rnti
19
20         NS_ASSERT(outerIt != m_rlcAttached.end());
21
22         uint8_t cqiValue = cqiElement.m_wbCqi;
23
24         if (outerIt != m_rlcAttached.end()) {
25             const std::unordered_map<uint8_t, NrMacSapUser
                   *>& innerMap = outerIt->second;
```

```
26              for (const auto& innerPair : innerMap) {
27                  uint8_t lcid = innerPair.first;
28                  NrMacSapUser* rlcInstance = innerPair.
                        second;
29                  rlcInstance->SetCqi(rnti, lcid, cqiValue);
30              }
31          }
32          else
33          {
34              NS_LOG_INFO(this << " RNTI not found in the map
                    .");
35          }
36
37          break;
38      }
39      ...
```

Algoritmo 5. Adições ao método `DoReceiveControlMessage` da camada MAC na gNB.

Para viabilizar esse acesso, como representado no Algoritmo 5, o CQI passou a ser recuperado na camada MAC e enviado às camadas superiores através do método `SetCqi`. O `SetCqi` foi implementado e propagado através das classes derivadas da classe `NrMacSapUser`, com o mesmo procedimento sendo aplicado à interface SAP entre as camadas RLC e PDCP. Esta implementação abre caminho para futuras pesquisas sobre a utilização do CQI na tomada de decisão do AQM, permitindo adaptar a penalização dos pacotes conforme as condições do canal.

3.3.7. Modificações na Camada RLC

```
1
2  bool
3  NrRlcUmDualpi2::isL4S(Ptr<Packet> packet)
4  {
5      NrPdcpHeader pdcpHeader;
6      if (packet->PeekHeader(pdcpHeader))
7      {
8          if (pdcpHeader.GetEct() == 1)
9          {
10             return true;
11         }
12         return false;
13     }
14     return false; // PDCP header not found
15 }
16
17 void
```

```
18 NrRlcUmDualpi2::DoTransmitPdcpPdu(Ptr<Packet> p)
19 {
20     ...
21     Ptr<QueueDiscItem> item;
22     if (isL4S(p))
23     {
24         item = Create<DualQueueL4SQueueDiscItem>(p, dest, 0);
25     }
26     else
27     {
28         item = Create<DualQueueClassicQueueDiscItem>(p, dest,
               0);
29     }
30
31     aqm->Enqueue(item);
32     ...
33 }
```

Algoritmo 6. Uso do bit ECT do cabeçalho PDCP no método DoTransmitPdcpPdu da camada RLC-UM

O Algoritmo 6 ilustra como as funções `DoNotifyTxOpportunity` e `DoTransmitPdcpPdu` foram reestruturadas para garantir compatibilidade com o gerenciador de fila DualPI2. A função `DoTransmitPdcpPdu` é responsável pelo gerenciamento da transmissão de PDUs, aplicando políticas de descarte baseadas em atraso e realizando a classificação dos pacotes em L4S ou Clássico. Enquanto que a função `DoNotifyTxOpportunity`, por sua vez, encarrega-se da segmentação e transmissão dos segmentos para a camada inferior, além de realizar o rearmazenamento de pacotes quando a quantidade de bytes disponibilizada (créditos) pela camada MAC é insuficiente. Para este processo de rearmazenamento, foi utilizada a função Requeue mencionada anteriormente.

Estas modificações envolveram a substituição da fila de transmissão nativa (txBuffer) da camada RLC-UM pelo mecanismo DualPI2, definido simplesmente como a variável `aqm` no código. É importante destacar que tais alterações foram fundamentais para a execução do experimento e utilizaram os mecanismos previamente implementados: o bit que indica se um pacote é ECT(1) no cabeçalho da camada PDCP foi utilizado para encapsular os pacotes no tipo apropriado de item de fila (clássico ou L4S), permitindo que o AQM segregasse o tráfego corretamente.

3.3.8. Diferenciação de Tráfego

Nativamente, no ns-3, o DCTCP, assim como TCP Cubic, usa o ECT(0) como bit indicador de tráfego ECT. Assim, a representação do DCTCP foi alterada de ECT(0) para ECT(1), associando ECT(1) exclusivamente ao tráfego L4S. Isso foi necessário porque, sem essa alteração, os dois tráfegos eram considerados de baixa latência e enfileirados na fila L4S. Sem falar que o acesso direto ao cabeçalho da camada de transporte na camada

RLC já não era possível, tendo apenas as informações disponíveis no cabeçalho IPv4 para diferenciar os dois tipos de tráfego, como ilustrado na Figura 6.

3.3.9. Adição de restrição à chamada do método `Drop`

Antes dessa modificação, o método `Dequeue` permitia que um pacote fosse eliminado mesmo se fosse o único na fila, desde que a probabilidade de descarte, calculada após o desenfileiramento, indicasse essa ação, como pode ser visto no Algoritmo 3. Essa abordagem resultava no desperdício de recursos, como o tempo de espera do pacote na fila e os créditos enviados pela camada MAC. Para resolver esse problema, foi adicionada uma restrição ao método: agora, um pacote só é descartado se não for o último pacote da fila. Essa mudança garante que recursos valiosos não sejam desperdiçados desnecessariamente, melhorando a eficiência do gerenciamento de filas.

3.4. Topologia do Ambiente de Testes

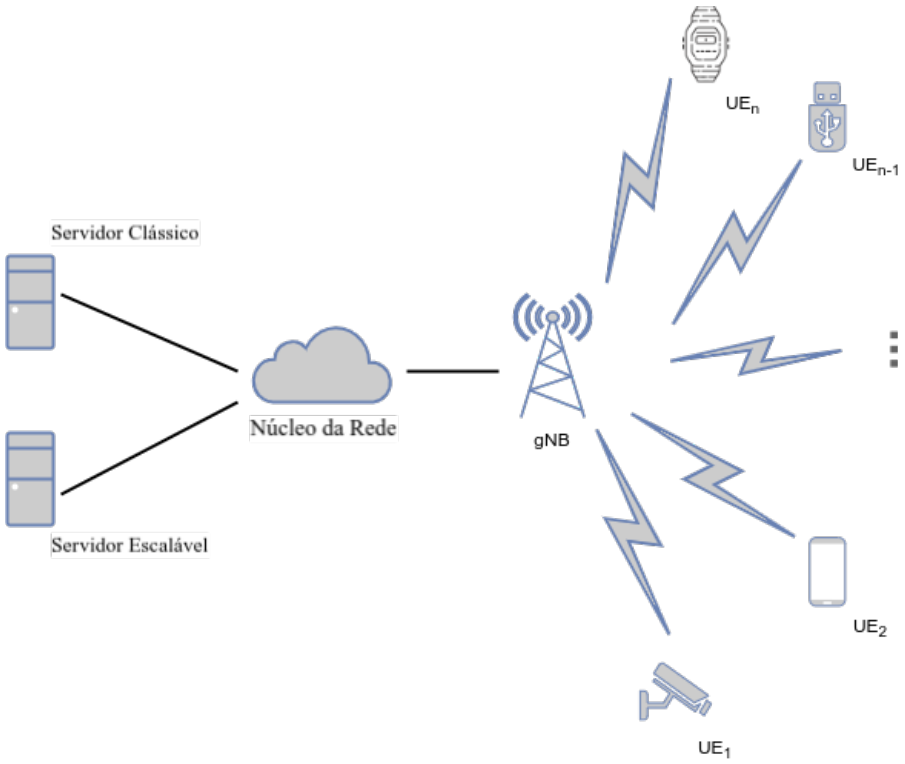

Figura 7. Topologia do ambiente de testes

Como é possível ver na Figura 7, a topologia consiste em um conjunto de usuários conectados diretamente a um único gNB, que por sua vez se comunica com um núcleo de rede EPC (*Evolved Packet Core*), responsável por estabelecer a conexão entre os UEs e os servidores remotos. Apesar de no 5G o core apresentar outra definição, como 5GC, por exemplo, ainda não é o caso do módulo 5G-LENA do ns-3, onde o EPC ainda é utilizado para fazer o gerenciamento da rede, e dos usuários. O EPC é composto por um *Serving Gateway* (SGW) e um *Packet Data Network Gateway* (PGW), que encaminham os pacotes para a Internet. Além disso, dois dispositivos remotos são utilizados como servidores

de tráfego para os UEs, permitindo a análise do desempenho de diferentes protocolos de transporte.

A topologia segue um modelo de acesso único, em que todos os UEs compartilham o mesmo gNB. A comunicação entre o gNB e o EPC ocorre através de um enlace ponto a ponto, enquanto a conectividade entre o EPC e os servidores remotos é estabelecida por um enlace de alta capacidade (10 Gb/s) e latência de 5 ms. Dessa forma, a principal fonte de variação na latência e no desempenho da rede reside na interface sem fio entre os UEs e o gNB.

Os UEs são distribuídos aleatoriamente dentro de um raio de 600 metros do gNB e utilizam um modelo de mobilidade randômica (*RandomWalk2dMobilityModel*) para simular deslocamento contínuo, como o de um carro, ou de um pedestre em movimento. Já o gNB e os elementos do núcleo da rede permanecem estáticos, sendo modelados com o método *ConstantPositionMobilityModel*, que permite que uma posição fixa no mapa seja atribuída a um nó. Já na camada física, o modo de esvanecimento (*fading*) padrão utilizado no canal de propagação do sinal é o modelo MIMO de *fast-fading* baseado na recomendação (3rd Generation Partnership Project (3GPP), 2022), como parte da classe `ThreeGppChannelModel`.

A rede utiliza as configurações padrões do 5G para operar, transmitindo sinais na frequência de 4 GHz com uma faixa de 10 MHz de largura. O sistema da antena celular distribui o acesso aos dispositivos de forma equilibrada, priorizando conexões estáveis (como em ambientes internos ou para dispositivos pouco móveis). Além disso, a potência de transmissão foi ajustada para 10dBm, garantindo uma cobertura eficiente para os dispositivos conectados.

A comunicação entre os nós é baseada em tráfego TCP, onde cada UE recebe dados de dois servidores remotos distintos, sendo dois fluxos no total. Um fluxo emprega o protocolo TCP Cubic (representando um tráfego clássico), enquanto o segundo utiliza TCP DCTCP, integrado à arquitetura L4S. Ambos os fluxos são iniciados após 2 segundos de simulação e continuam até o término da execução. Para a transmissão de pacotes, são utilizados os módulos do ns-3 *BulkSendHelper* (para envio de dados) e *PacketSinkHelper* (para a recepção), que são configurados para gerar dados de maneira contínua, a fim de sobrecarregar a rede.

A analise do desempenho da rede foi feita através de um sistema de monitoramento configurado para a vazão, a quantidade de pacotes perdidos e os atrasos na comunicação entre os servidores e os dispositivos móveis. Além disso, a camada RLC foi ajustada para registrar informações detalhadas sobre seu funcionamento, tanto quando utiliza o gerenciador de filas DualPI2 quanto quando opera com o sistema de armazenamento padrão, o txBuffer. Esses dados incluem o tamanho da fila de transmissão, o tempo médio que os pacotes ficam aguardando para serem enviados e a quantidade de recursos liberados pela camada MAC para o envio de informações.

A configuração do ambiente de simulação foi meticulosamente planejada para reproduzir condições realistas de congestionamento em redes 5G, permitindo uma avaliação precisa da efetividade da arquitetura L4S na redução da latência e na melhoria da qualidade de serviço. A escolha do DCTCP justifica-se por múltiplos fatores: além de ser um protocolo plenamente compatível com a arquitetura L4S, demonstrando-se escalável

e comprovadamente eficaz na mitigação de problemas de congestionamento como o *bufferbloat*, já estava implementado no framework ns-3, adequando-se assim às restrições temporais estabelecidas para a realização do projeto.

3.5. Metodologia de Avaliação

Esta seção descreve a metodologia adotada para conduzir o estudo, detalhando os critérios e procedimentos utilizados para avaliar o desempenho da arquitetura L4S em redes 5G NR simuladas no ns-3. O objetivo principal foi analisar o impacto da implementação do gerenciador de filas DualPI2 em diferentes cenários de carga e congestionamento, comparando os resultados com uma configuração tradicional da camada RLC, ou seja, sem AQM.

Para isso, os serviços configurados foram projetados para estressar a rede, permitindo a observação do comportamento da arquitetura L4S sob condições extremas. A seguir, são descritos os cenários de simulação e as métricas utilizadas para a avaliação.

3.5.1. Configuração dos Cenários de Simulação

Foram simulados diversos cenários, variando fatores-chave para avaliar o desempenho da arquitetura L4S em diferentes condições. Os principais fatores configurados incluem:

- **Número de UEs**: O número de dispositivos conectados foi variado entre 2, 5, 7 e 10, com o intuito de analisar o comportamento da rede sob diferentes níveis de concorrência e carga.
- **RLC com e sem DualPI2**: Para cada cenário definido pelo número de UEs, a simulação foi executada em duas configurações distintas: com o AQM DualPI2 ativado e sem ele. Essa abordagem permitiu comparar o desempenho da arquitetura L4S com o de uma rede tradicional, sob as mesmas condições de carga e congestionamento.

3.5.2. Métricas de Avaliação

Para avaliar o desempenho da solução proposta, as seguintes métricas foram empregadas:

- **Latência na Fila (*Queue Delay*)**: Tempo médio que os pacotes permanecem na fila da camada RLC do gNB, tanto com o DualPI2 ativado quanto sem ele. Essa métrica é crucial para avaliar a eficiência do gerenciamento de filas.
- **Descarte e Marcação de Pacotes**: Quantidade média de pacotes descartados, e marcados, pelo AQM em comparação com a execução sem AQM. Essa métrica ajuda a entender o impacto do DualPI2 na prevenção de congestão na rede.
- **Vazão**: Taxa efetiva de entrega de pacotes aos UEs, medida em Mbps. Essa métrica reflete a capacidade da rede de manter um alto desempenho mesmo sob condições de carga elevada.

4. Resultados e Discussão

Com o objetivo de aumentar a confiabilidade dos resultados, cada configuração foi executada 5 vezes. Essa abordagem busca mitigar possíveis influências de variações aleatórias

de desempenho, oferecendo uma base mais consistente para a avaliação da arquitetura L4S no contexto do 5G.

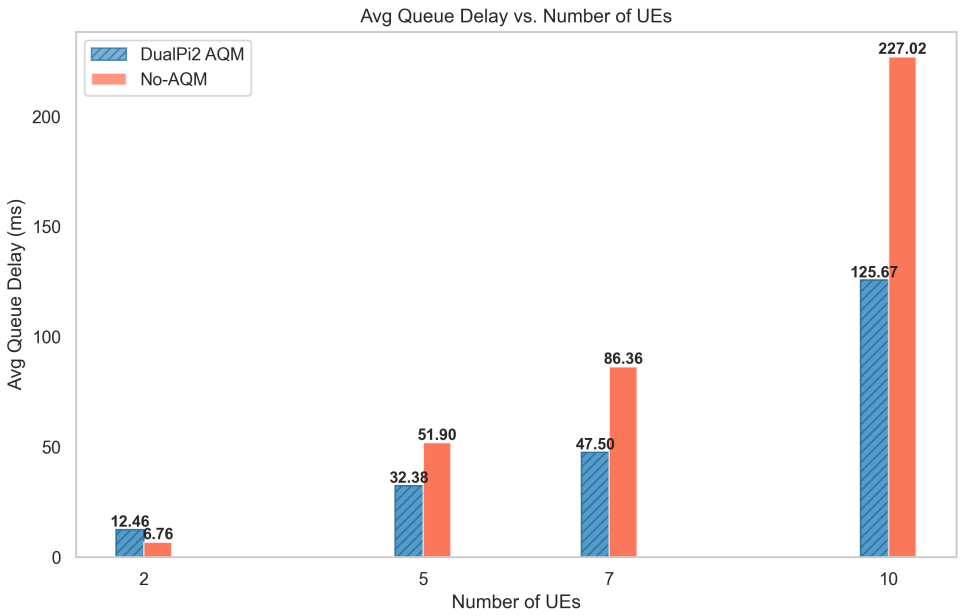

Figura 8. Comparação do atraso de fila entre versões da camada RLC

Através dos experimentos, foi possível constatar que a implementação do gerenciador de filas DualPI2 na camada RLC proporcionou uma melhoria significativa no desempenho da rede, como pode ser observado na Figura 8. No experimento com 10 UEs, a redução do atraso de fila na camada RLC foi de aproximadamente 55% em comparação com a configuração sem AQM. Esse padrão se repetiu para os demais cenários, com exceção do caso com apenas dois UEs, onde a diferença foi menos pronunciada.

No cenário com apenas dois usuários, o uso do DualPI2 na camada RLC pode ter causado um desempenho pior justamente por não haver congestionamento real nas filas — o que leva o algoritmo a descartar pacotes de forma desnecessária, antecipando um congestionamento que não ocorre. Como o DualPI2 atua com base na latência da fila, ele pode interpretar pequenas variações como sinal de sobrecarga e agir de forma agressiva, resultando em perdas prematuras, aumento do *jitter* e, consequentemente, maior latência média. Além disso, a lógica de fairness embutida no algoritmo pode introduzir atrasos artificiais mesmo em ambientes de baixa competição, o que acaba comprometendo o desempenho em vez de otimizá-lo.

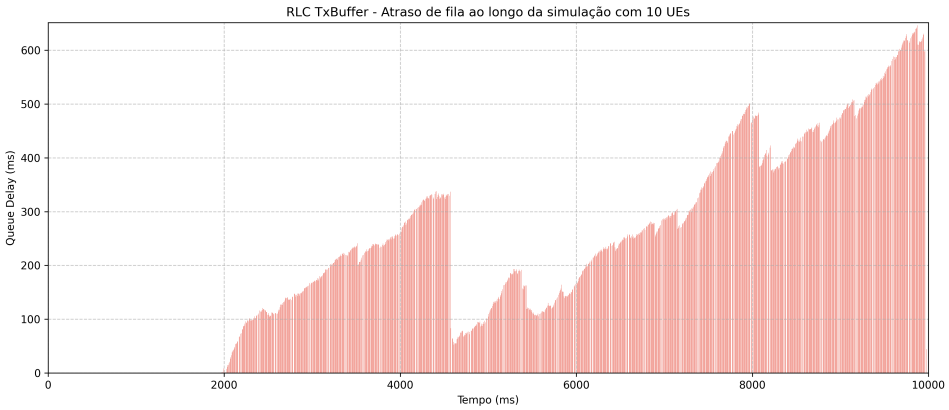

Figura 9. Evolução do atraso de fila da camada RLC-UM com o txBuffer para 10 dispositivos conectados por 10s

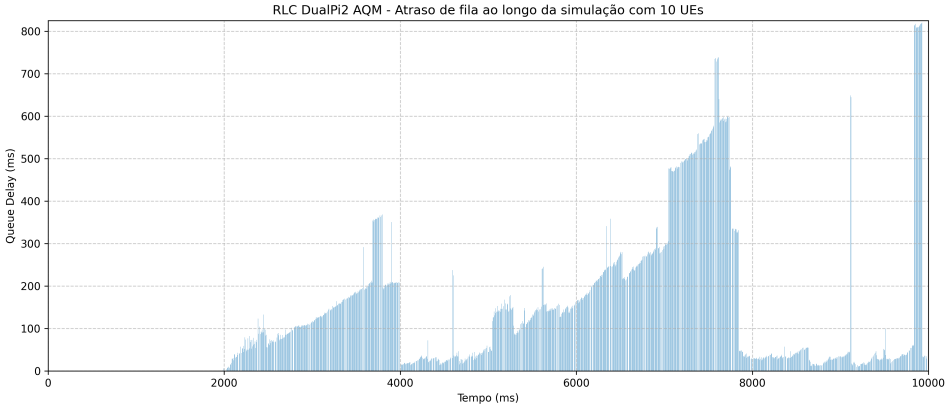

Figura 10. Evolução do atraso de fila da camada RLC-UM com o DualPI2 para 10 dispositivos conectados por 10s

Nas Figuras 9 e 10, podemos ver a evolução do atraso de fila da camada RLC com o txBuffer e com o DualPI2, respectivamente, ao longo de toda a simulação. Essas figuras reforçam o DualPI2 como um mecanismo capaz de reduzir drasticamente a latência de fila nos dispositivos em que é implementado.

A camada RLC original, embora não implemente uma política ativa de gerenciamento de filas, realiza o descarte de pacotes em dois cenários específicos:

1. Quando o número de bytes armazenados na fila atinge o limite definido (10 * 1024 bytes).
2. E quando o descarte de PDUs da camada PDCP está habilitado e o tempo de espera de um pacote, desde sua chegada na camada PDCP até seu processamento na camada RLC, excede uma tolerância pré-definida.

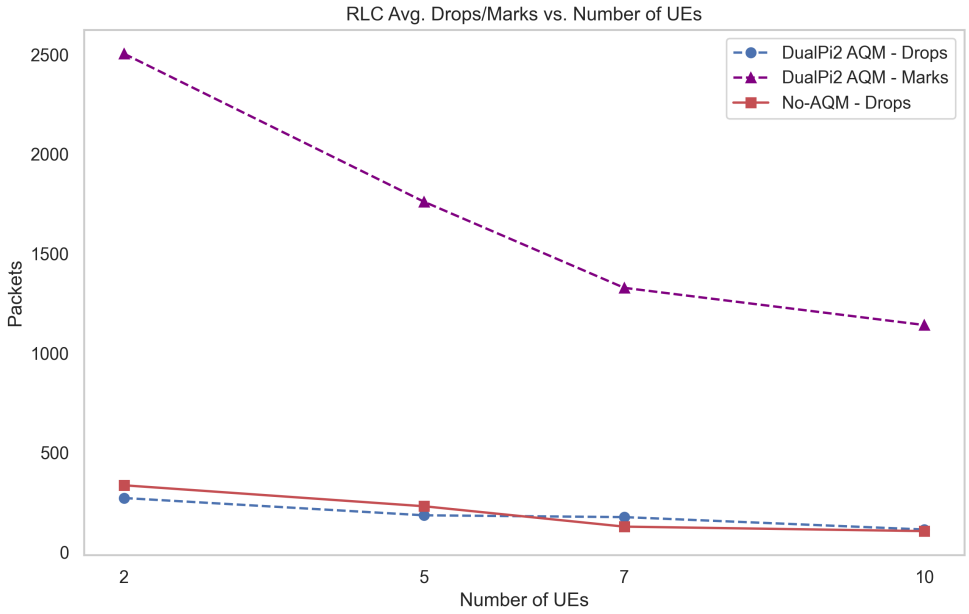

Figura 11. Descartes e Marcações

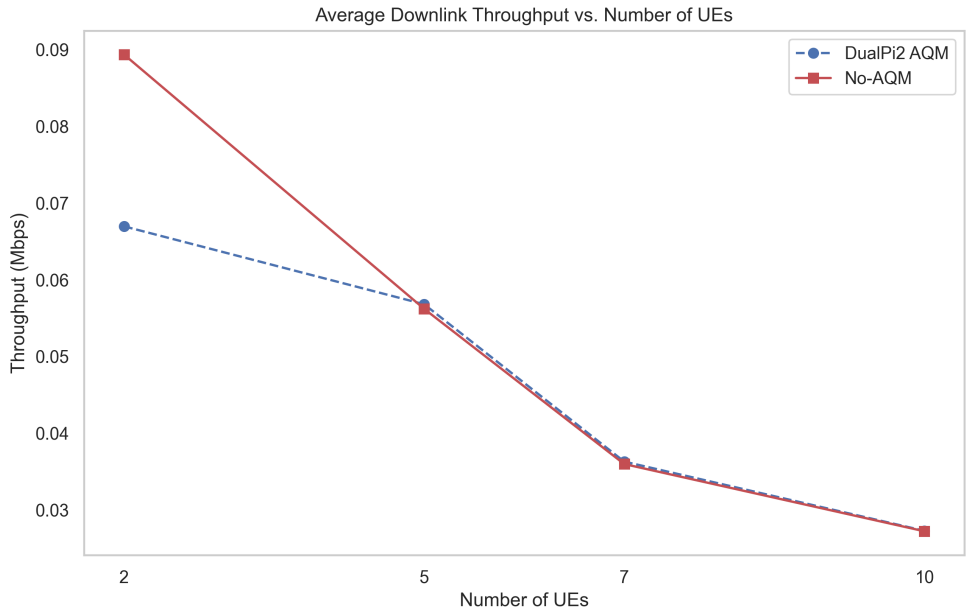

Figura 12. Vazão

Com a adição do DualPI2 à camada RLC, políticas adicionais que podem levar ao descarte de pacotes, além das já existentes na camada RLC original, foram introduzidas. No entanto, como mostra a Figura 11, o número de pacotes descartados pelo DualPI2 foi similar ao da camada RLC original, reforçando a eficiência do DualPI2 em gerenciar

a rede, uma vez que ele não precisou aumentar significativamente o descarte de pacotes para alcançar uma redução no atraso na fila.

Além disso, o DualPI2 mostrou-se eficaz ao utilizar políticas de marcação de pacotes para controlar o congestionamento, evitando a necessidade de reduzir a vazão da rede. Essa eficiência é confirmada pelos dados de vazão apresentados na Figura 12, onde se observa que o DualPI2 manteve um vazão estável e competitivo em comparação com a configuração sem AQM, mesmo aplicando políticas simultâneas de descarte e marcação de pacotes inexistentes na versão original da camada.

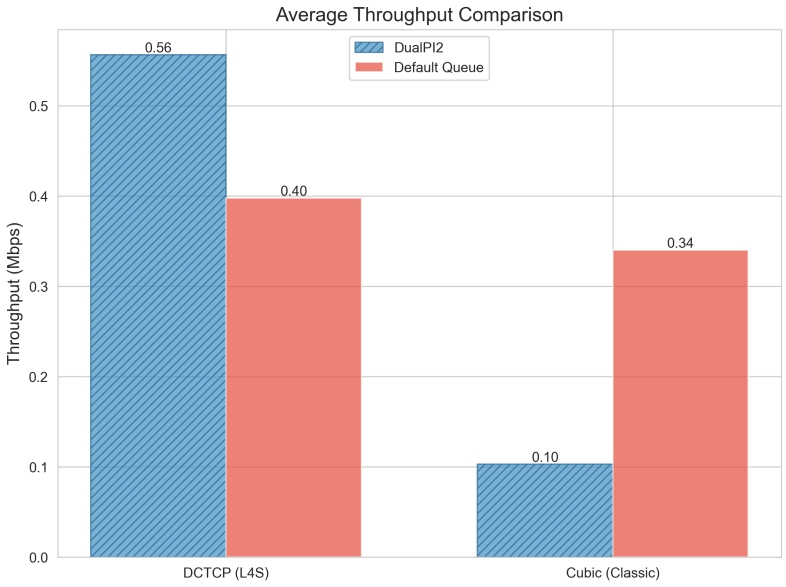

Figura 13. Comparação da vazão entre os fluxos clássicos e escaláveis para 10 UEs no downlink

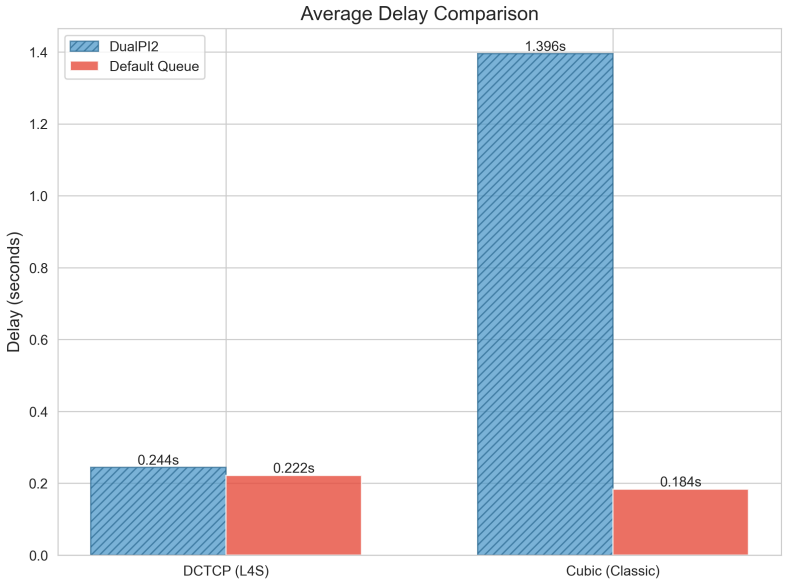

Figura 14. Comparação do atraso entre os fluxos clássicos e escaláveis para 10 UEs no downlink

Agora, olharemos diretamente para os fluxos clássicos, com o TCP Cubic, e escaláveis, com o DCTCP, no sentido de *downlink*. Os resultados da comparação entre eles, exibidos nas Figuras 13 e 14, revelam aspectos importantes sobre o comportamento do DualPI2 em condições de tráfego misto no *downlink*. Como evidenciado pelos dados exibidos na Figura 13, os fluxos DCTCP, na cor azul, sob o controle do DualPI2 apresentaram um aumento significativo da vazão (39,88%) em comparação com a implementação padrão, alcançando 0,56 Mbps contra 0,4 Mbps. Entretanto, é importante notar que essa melhoria para os fluxos L4S ocorre às custas do desempenho dos fluxos clássicos, ilustrados na cor vermelha na Figura 13, que experimentaram uma redução drástica de vazão (-69,72%) e um aumento significativo no atraso médio (de 0,183s para 1,4s), como pode ser visto na Figura 14. A razão da vazão entre DCTCP e Cubic aumentou de 1,17 para 5,41 com o DualPI2, indicando um forte favorecimento dos fluxos L4S. Esse comportamento pode ser atribuído à natureza mais agressiva do algoritmo DCTCP na utilização do enlace, aproveitando-se da separação de filas e do tratamento diferenciado proporcionado pelo DualPI2. Embora esse desequilíbrio possa ser preocupante em termos de equidade, ele demonstra a eficácia do DualPI2 em priorizar fluxos L4S em cenários de congestionamento, cumprindo assim um dos objetivos principais da arquitetura L4S, que é permitir transmissões com baixa latência mesmo em condições de alta utilização da rede.

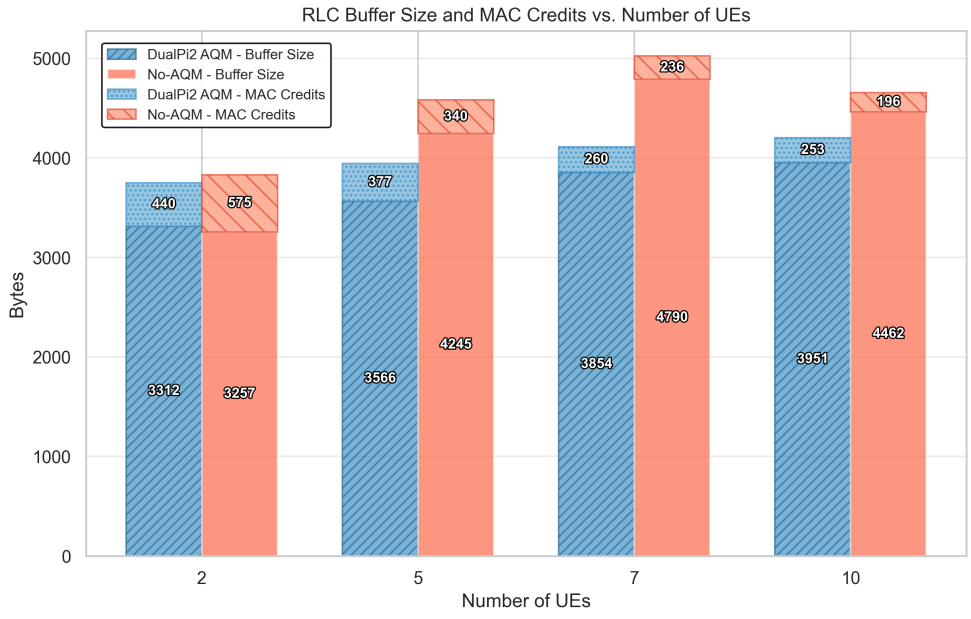

Figura 15. Comparação entre o tamanho da fila e os créditos da camada MAC

A partir do gráfico da Figura 15, podemos obter uma compreensão mais ampla do atraso de fila apresentado na Figura 8. Em comparação com experimentos realizados em ambientes reais, atrasos na casa das dezenas de milissegundos já são suficientes para inviabilizar diversos serviços, especialmente aqueles que dependem de baixa latência, como videoconferências, jogos online e aplicações industriais de tempo real. Embora, atualmente, as restrições de atraso não sejam tão severas quanto as exigidas por aplicações

desenvolvidas especificamente para o 5G, como comunicações ultraconfiáveis e de baixa latência, valores como esses representam um desafio significativo para a qualidade da experiência do usuário e a eficiência da rede. Felizmente, através da comparação do delay de fila da camada RLC original, vemos que o alto delay apresentado é algo intrínseco ao ambiente de simulação ns-3 sob as circunstâncias de estresse ao qual foi submetido, sendo, assim, válidas as melhorias trazidas pela implementação do DualPI2 na RAN.

Além disso, observamos que, mesmo sem a implementação do DualPI2 na camada RLC, o tamanho da fila sob condições de estresse da rede é excessivamente grande em relação aos créditos disponibilizados pela camada MAC para a transmissão dos dados armazenados. Essa discrepância entre a capacidade da fila e os recursos alocados pela camada MAC resulta em um aumento significativo do atraso de fila, uma vez que os pacotes permanecem aguardando até que créditos suficientes para sua transmissão sejam liberados.

Essa análise sugere que, além da implementação de mecanismos avançados de gerenciamento de filas, como o DualPI2, é fundamental revisar a alocação de recursos na camada MAC para garantir que as filas não se tornem um gargalo na rede. Em particular, a sincronização entre as camadas RLC e MAC deve ser otimizada para evitar situações em que a fila fique sobrecarregada enquanto os créditos de transmissão são insuficientes para escoar o tráfego de forma eficiente.

Adicionalmente, os resultados destacam a importância de adaptar as configurações do DualPI2 e a concessão de créditos às condições específicas da rede, como o número de UEs e a carga de tráfego. Em cenários com maior concorrência e demanda por recursos, como os testados nesta simulação, a falta de ajustes dinâmicos pode levar a atrasos excessivos e degradação do desempenho geral da rede. Essa necessidade de orquestração entre camadas da RAN é destacada pelo estudo da Ericsson (Ericsson, 2021) e se confirma pelas métricas obtidas neste trabalho, que demonstram como a falta de coordenação pode levar a atrasos excessivos e à degradação do desempenho.

Por fim, esses insights reforçam a necessidade de uma abordagem holística no projeto e na otimização de redes 5G, onde não apenas o gerenciamento de filas, mas também a alocação de recursos e a coordenação entre camadas sejam considerados de forma integrada. A implementação do DualPI2, nesse sentido, representa um avanço importante ao garantir que os pacotes sejam processados e entregues com menor atraso, mesmo em cenários de alta carga, mas sua eficácia pode ser ainda maior quando combinada com outras melhorias na arquitetura da rede.

5. Trabalhos Relacionados

Trabalhos recentes abordam diferentes abordagens para mitigar os desafios associados à latência, à variabilidade do canal e à adaptação da taxa de transmissão. Nesta seção, revisamos três estudos relevantes que contribuem para o avanço dessas técnicas.

Inicialmente, abordaremos o estudo de Ericsson e Deutsche Telekom (Ericsson, 2021), que também explora a viabilidade da arquitetura L4S no 5G para suportar aplicações críticas em tempo real, sendo publicado como um convite à comunidade desenvolvedora para o teste e desenvolvimento da arquitetura L4S em contextos móveis. Além disso, o estudo propõe usar fluxos QoS dedicados para tráfego L4S, e também o CQI,

em conjunto com a métrica de atraso de fila usual, para uma reação mais proativa ao congestionamento. Enquanto (Ericsson, 2021) discute soluções em alto nível, o presente trabalho oferece uma implementação específica na camada RLC, fornecendo detalhes técnicos mais profundos sobre como essas melhorias podem ser realizadas no ns3.

Outro estudo relevante, desenvolvido por (Dai et al., 2017), propõe um algoritmo de gerenciamento de fila sensível à qualidade do canal para redes celulares LTE. O trabalho argumenta que o congestionamento em redes móveis não é apenas uma função do tamanho da fila, mas também da variabilidade do canal sem fio. Assim, os autores propõem um algoritmo que utiliza a métrica CQI para ajustar dinamicamente a probabilidade de descarte de pacotes, garantindo um tempo médio de fila reduzido. Os resultados de simulação mostram que a abordagem reduz em aproximadamente 40% o atraso médio de fila quando comparado ao algoritmo CoDel tradicional, sem comprometer a utilização do enlace. Nesse estudo, além de trazermos a implantação de um AQM na gNB do 5G, optamos por não adicionar o CQI como uma métrica do DualPI2 por restrições de tempo, porém os recursos necessários ao seu desenvolvimento em trabalhos futuros foram implementados e encontram-se disponíveis no repositório do projeto no GitHub.

Por fim, (Irazabal; Lopez-Aguilera; Demirkol, 2017) exploram o uso do AQM como facilitador da qualidade de serviço em redes 5G. O trabalho destaca que, apesar da padronização das classes de QoS pela 3GPP, os métodos específicos para garantir requisitos de latência e vazão permanecem indefinidos. Os autores avaliam diferentes mecanismos de AQM, incluindo o CoDel, na camada SDAP da rede 5G. Os experimentos revelam que a combinação de CoDel com filas de tamanho limitado nos Data Radio Bearers (DRB) reduz significativamente a latência para tráfego sensível ao tempo, ao mesmo tempo em que mantém o vazão eficiente para fluxos de dados em segundo plano.

6. Conclusão

Este trabalho demonstrou a eficácia da implementação do AQM DualPI2 na camada RLC para melhorar o desempenho de redes 5G simuladas no ns-3. Os resultados mostraram que o DualPI2 foi capaz de reduzir significativamente o atraso de fila em comparação com a configuração sem AQM, especialmente em cenários com maior carga de tráfego e concorrência entre dispositivos. Além disso, o DualPI2 manteve a vazão da rede estável sem aumentar consideravelmente o descarte de pacotes, destacando sua eficiência no gerenciamento de congestionamento.

A análise mais detalhada dos fluxos L4S e Classic trouxe insights importantes sobre o comportamento do DualPI2 em tráfego misto. Os resultados mostram que a implementação favoreceu significativamente os fluxos DCTCP (L4S), proporcionando um aumento de 39,88% da vazão e redução de perdas de pacotes de 18,36% para 8,83%. Entretanto, observou-se um desbalanceamento na distribuição de recursos, com os fluxos Cubic (clássicos) sofrendo uma redução de 69,72% no vazão e um aumento expressivo no atraso. Este comportamento, embora evidencie a capacidade do DualPI2 em priorizar tráfego sensível à latência, também aponta para a necessidade de ajustes nos parâmetros do algoritmo para cenários onde a equidade entre classes de tráfego seja desejável.

A análise também revelou a importância de otimizar a alocação de recursos na camada MAC e a sincronização entre as camadas RLC e MAC para evitar que os buffers se tornem um gargalo na rede. A disparidade observada entre o tamanho dos buffers e

os créditos MAC disponíveis, somada às diferenças de desempenho entre tipos de fluxo, reforça a necessidade de uma abordagem integrada no projeto e na otimização de redes 5G, onde múltiplos aspectos da arquitetura são considerados em conjunto.

Em resumo, esta pesquisa representa uma contribuição significativa para o avanço do estado da arte em gerenciamento ativo de filas e otimização de desempenho em redes 5G. Os resultados obtidos com a implementação do DualPI2 na camada RLC e a análise da interação entre fluxos L4S e Classic estabelecem uma base sólida para futuras investigações científicas dedicadas à otimização de infraestruturas de rede, balanceamento de recursos entre diferentes classes de tráfego e desenvolvimento de algoritmos de controle de congestionamento mais equitativos, visando viabilizar o pleno funcionamento de aplicações emergentes que demandam alta performance e baixa latência.

7. Trabalhos Futuros

Com base nos resultados e análises apresentados, algumas direções promissoras para trabalhos futuros são sugeridas, visando aprimorar o desempenho do DualPI2 e ampliar sua aplicabilidade em redes 5G.

Integração do CQI como métrica do DualPI2 O *Channel Quality Indicator* é uma métrica fundamental para a avaliação da qualidade do canal entre a estação rádio base e o usuário. A incorporação dessa métrica ao DualPI2 permitiria uma gestão mais inteligente do congestionamento, adaptando as decisões de marcação e descarte de pacotes às condições dinâmicas do canal. Em cenários de baixa qualidade de canal, onde há menos blocos de recurso disponíveis, o AQM poderia priorizar pacotes críticos, minimizando o impacto de atrasos excessivos e otimizando a utilização da largura de banda. Essa abordagem beneficiaria especialmente aplicações sensíveis à latência, garantindo maior eficiência e melhor experiência ao usuário final.

Experimentação com TCP Prague para fluxos L4S Uma direção promissora para trabalhos futuros consiste na implementação e avaliação do algoritmo TCP Prague como alternativa ao DCTCP nos fluxos L4S. Diferentemente do DCTCP, que demonstrou comportamento agressivo na utilização do enlace e resultou em um desbalanceamento significativo no compartilhamento de recursos com fluxos clássicos, o TCP Prague foi especificamente projetado para oferecer melhor coexistência e equidade com outros tipos de tráfego, mantendo os benefícios de baixa latência. A RFC 9331 (Schepper; Briscoe, 2023), que padroniza a arquitetura L4S, recomenda o Prague como o algoritmo de referência para aplicações L4S em ambientes de produção. Futuros experimentos poderiam avaliar comparativamente o desempenho do Prague e do DCTCP sob o controle do DualPI2 na camada RLC, analisando não apenas métricas de latência e vazão, mas também aspectos de equidade e estabilidade em diferentes condições de carga e número de dispositivos. Esta análise permitiria determinar se o TCP Prague consegue de fato mitigar o problema de desproporcionalidade observado entre fluxos L4S e clássico, oferecendo uma solução mais equilibrada para implantações de L4S em redes 5G reais.

Validação em Ambientes Reais Embora a simulação ofereça uma plataforma valiosa para experimentação, a validação do DualPI2 em ambientes reais é essencial para conso-

lidar sua aplicabilidade prática. Testes em plataformas de experimentação, como testbeds 5G, permitiriam avaliar o desempenho do algoritmo sob condições de rede reais, incluindo mobilidade dos usuários, interferências e variações dinâmicas de carga. Além disso, essa abordagem possibilitaria a comparação direta entre os resultados de simulação e cenários reais, permitindo ajustes mais precisos no modelo de controle de congestionamento.

Análise de Escalabilidade e Eficiência Energética Investigar a escalabilidade do DualPI2 em cenários com um grande número de usuários é crucial para entender seus impactos na infraestrutura da rede. Além disso, é relevante avaliar seu efeito na eficiência energética, uma vez que a gestão do congestionamento pode influenciar diretamente o consumo de recursos computacionais e energéticos dos dispositivos e da rede como um todo. Métodos para otimizar o processamento do AQM e reduzir o consumo energético sem comprometer a qualidade do serviço são aspectos que merecem atenção em estudos futuros.

Aprimoramento do Ambiente de Simulação O simulador ns-3 desempenha um papel essencial no desenvolvimento e teste de novas soluções para redes 5G. Contudo, aprimorar suas funcionalidades e incorporar características mais realistas pode proporcionar avaliações mais precisas e confiáveis. Trabalhos futuros podem explorar a implementação de novos modelos de propagação, aprimoramentos no suporte à mobilidade e simulação de diferentes configurações de rede. Além disso, contribuir para o desenvolvimento da comunidade ns-3 pode ampliar o impacto das pesquisas na área de redes sem fio.

Essas direções de pesquisa têm o potencial de aprimorar significativamente o DualPI2 e contribuir para a evolução das redes 5G, tornando-as mais eficientes, resilientes e adaptáveis às demandas futuras.

Referências

NS-3 CONSORTIUM. **About ns-3**. [S.l.: s.n.], 2023. https://www.nsnam.org/documentation/. Accessed: 2023-10-10. Disponível em: https://www.nsnam.org/documentation/.

3GPP. **5G System Overview**. [S.l.: s.n.], 2023. https://www.3gpp.org/technologies/5g-system-overview. Accessed: 2023-10-05. Disponível em: https://www.3gpp.org/technologies/5g-system-overview.

_____. **NR; Medium Access Control (MAC) protocol specification**. [S.l.], 2020.

_____. **NR; Overall description; Stage-2**. [S.l.], 2019.

_____. **NR; Packet Data Convergence Protocol (PDCP) specification**. [S.l.], 2020.

_____. **NR; Radio Link Control (RLC) protocol specification**. [S.l.], 2020.

_____. **NR; Radio Resource Control (RRC) protocol specification**. [S.l.], 2020.

3RD GENERATION PARTNERSHIP PROJECT (3GPP). **Study on channel model for frequencies from 0.5 to 100 GHz**. [S.l.], jun. 2022. https://www.3gpp.org/ftp/Specs/archive/38_series/38.901/38901-g00.zip.

BAKER, F.; FAIRHURST, G. **IETF Recommendations Regarding Active Queue Management**. [Online]: IETF, 2015. RFC 7567. Disponível em: https://datatracker.ietf.org/doc/html/rfc7567.

BRISCOE, B. et al. **Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture**. [Online]: IETF, 2023. RFC 9330. Disponível em: https://datatracker.ietf.org/doc/rfc9330.

DAHLMAN, Erik; PARKVALL, Stefan; SKÖLD, Johan. **5G NR: The next generation wireless access technology**. [S.l.]: Academic Press, 2020.

DAI, Yuhang et al. Channel Quality Aware Active Queue Management in Cellular Networks. **IEEE Xplore**, 2017.

ERICSSON. **Enabling Time-Critical Applications Over 5G with Rate Adaptation**. Online, 2021. White Paper. Disponível em: https://www.telekom.com/en/company/details/enabling-time-critical-applications-over-5g-with-rate-adaptation-628058.

GSMA. **The Mobile Economy 2022**. [S.l.: s.n.], 2022. https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf. Accessed: 2023-10-05. Disponível em: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf.

IRAZABAL, Mikel; LOPEZ-AGUILERA, Elena; DEMIRKOL, Ilker. Active Queue Management as Quality of Service Enabler for 5G Networks. **IEEE Xplore**, 2017.

PAN, Rong et al. PIE: A lightweight control scheme to address the bufferbloat problem. **IEEE HPSR**, jul. 2013. DOI: 10.1109/HPSR.2013.6602305.

SCHEPPER, K.; BRISCOE, B. **The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)**. [Online]: IETF, 2023. RFC 9331. Disponível em: https://datatracker.ietf.org/doc/rfc9331.

SCHEPPER, Koen De; TILMANS, Olivier et al. **Prague Congestion Control**. [S.l.], jul. 2024. 34 p. Work in Progress. Disponível em: https://datatracker.ietf.org/doc/draft-briscoe-iccrg-prague-congestion-control/04/.

VERIZON. **5G Understanding: eMBB, URLLC, and mMTC**. [S.l.: s.n.], 2023. https://www.verizon.com/about/news/5g-understanding-embb-urllc-mmtc. Accessed:

2023-10-05. Disponível em: https://www.verizon.com/about/news/5g-understanding-embb-urllc-mmtc.

ZHANG, Lixia et al. **Recommendations on Queue Management and Congestion Avoidance in the Internet**. [S.l.]: RFC Editor, abr. 1998. 17 p. RFC 2309. (Request for Comments, 2309). DOI: 10.17487/RFC2309. Disponível em: https://www.rfc-editor.org/info/rfc2309.