



Universidade Federal de Uberlândia  
Faculdade de Engenharia Elétrica  
Graduação em Engenharia Eletrônica e de Telecomunicações

**THIAGO PENSO BEVILACQUA**

**AVALIAÇÃO DE PERFORMANCE DE ALGORITMOS DE  
ESCALONAMENTO EM REDES LTE**

Uberlândia  
2019

**THIAGO PENSO BEVILACQUA**

**AVALIAÇÃO DE PERFORMANCE DE ALGORITMOS DE  
ESCALONAMENTO EM REDES LTE**

Trabalho apresentado como requisito parcial de avaliação na disciplina Trabalho de Conclusão de Curso de Engenharia Eletrônica e de Telecomunicações da Universidade Federal de Uberlândia.

Orientador: Prof. Dr. Paulo Roberto Guardieiro

---

Assinatura do Orientador

Uberlândia  
2019

Dedico este trabalho a minha família e aos meus colegas que tornaram possível a realização desse trabalho.

## **AGRADECIMENTOS**

Agradeço, primeiramente, à minha família, que me deu a possibilidade de ingressar em uma faculdade e de realizar meus estudos de forma integral.

A todos amigos que me ajudaram em vários momentos em que tive dificuldade em aprender algum conteúdo ou de realizar alguma tarefa e aos que proporcionavam sorrisos e alegrias em momentos de descontração.

À minha namorada, que mesmo não tendo ideia das matérias e conteúdo do meu curso, estava sempre ao meu lado.

Ao meu orientador pelo incentivo, ajuda, orientação e motivação.

## RESUMO

O objetivo deste trabalho foi realizar um estudo das redes *Long-Term Evolution* (LTE) com foco nos mecanismos de escalonamento e realizar uma comparação desses algoritmos de escalonamento em um software de simulação de redes LTE, o software escolhido foi o ns-3. Os algoritmos simulados foram o *Proportional Fair* (PF), *Channel and QoS Aware Scheduler* (CQA), *Priority Set Scheduler* (PSS) e *Token Bank Fair Queue Scheduler* (TBFQ), todos eles já estavam implementados no software. Para construção da simulação, foi necessária confecção de um código em C++ que usava as bibliotecas do simulador, assim todas as APIs que relacionam aos modelos simulados foram estudadas a fim de gerar a simulação. O cenário simulado considera uma rede com 2 tipos de usuários, um que requer QoS e outro que usa de um serviço *Best Effort*. Os resultados da simulação na forma de arquivos de texto foram processados no MATLAB® para organização dos dados e confecção dos gráficos. E, a fim de facilitar a execução das simulações, foi criado um *bash script* que realiza a simulação de cada algoritmo de forma automatizada. Como exposto na literatura, foi o algoritmo CQA o qual obteve melhor desempenho para garantir atraso limitado.

## **ABSTRACT**

The purpose of this work was to study the Long-Term Evolution (LTE) networks with a focus on the scheduling mechanisms and to compare these scheduling algorithms in the LTE network simulation software, the software chosen was ns-3. The simulated algorithms were the Proportional Fair (PF), Channel and QoS Aware Scheduler (CQA), Priority Set Scheduler (PSS) and Token Bank Fair Queue Scheduler (TBFQ), all of which were already implemented in the software. To construct the simulation, it was necessary to make a code in C++ that used the libraries of the simulator, so all the APIs that relate to the simulated models were studied in order to generate the simulation. The simulated scenario considers a network with two types of users, one that requires QoS and another that uses a Best Effort service. The simulation results in the form of text files were processed in MATLAB for data organization and graphing. And, in order to facilitate the execution of the simulations, a bash script was created that simulates each algorithm in an automated way. As shown in the literature, the CQA algorithm was the best performing to guarantee limited delay.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Evolução do tráfego de dados comparados com tráfego de voz durante os anos de 2009 a 2013.....	15
Figura 2 – Evolução do tráfego de dados comparados com tráfego de voz durante os anos de 2013 a 2018.....	16
Figura 3 – O serviço da arquitetura de rede.....	17
Figura 4 – Grade de recursos no diagrama Tempo-Frequência.....	19

## LISTA DE TABELAS

Tabela 1 – Quantidade de RB e RBG por largura de banda.....	20
Tabela 2 – Classes de QoS e suas características de vazão e perda de pacotes.....	21
Tabela 3 – Parâmetros usados na simulação.....	30



## LISTA DE GRÁFICOS

Gráfico 1 – Vazão média agregada X número de UEs.....	32
Gráfico 2 – Atraso médio X número de UEs.....	33
Gráfico 3 – Perda de pacotes X número de usuários.....	34
Gráfico 4 – Vazão Agregada X número de UEs.....	35

## LISTA DE ABREVIATURAS E SIGLAS

3GPP – *3rd Generation Partnership Project*  
AMC – *Adaptative Modulation and Coding*  
AMR-WB – *Adaptive Multi-Rate Wideband*  
ANATEL – *Agência Nacional de Telecomunicações*  
API – *Application Programming Interface*  
ARQ – *Automatic Repeat request*  
BET – *Blind Equal Throughput*  
Colta – *Carrier over Interference to average*  
CQA – *Channel and QoS Aware Scheduler*  
CQI – *Channel Quality Indicator*  
CSFB – *Circuit Switch FallBack*  
eNB – *Evolved NodeB*  
EPC – *Evolved Packet Core*  
E-ULTRAN – *Evolved-Universal Terrestrial Radio Access Network*  
FDD – *Frequency Division Duplexing*  
FD-TBFQ – *Frequency Domain*  
FEC – *Forward Error Correction*  
GBR – *Guaranteed Bit Rate*  
GSM – *Global System for Mobile Communication*  
HARQ – *Hybrid Automatic Repeat request*  
HOL – *Head of Line*  
HSS – *Home Subscriber Server*  
IP – *Internet Protocol*  
ITU – *International Telecommunication Union*

LTE – *Long Term Evolution*  
MME – *Mobility Management Entity*  
MT – *Maximum Throughput*  
NS-3 – *Network Simulator*  
OFDM – *Orthogonal Frequency Division Multiplexing*  
ONU – *Organização das Nações Unidas*  
PCFR – *Policy and Charge Rules Function*  
PF – *Proportional Fair*  
PGW – *Packet Data Network Gateway*  
PSS – *Priority Set Scheduler*  
QCI – *QoS Class Identifiers*  
QoS – *Quality of Service*  
RAN – *Radio Access Network*  
RB – *Resource Block*  
RBG – *Resource Block Group*  
RRM – *Radio Resource Management*  
SC\_FDMA – *Single Carrier Frequency Division Multiple Access*  
SGW – *Serving Gateway*  
SINR – *Signal to Interference plus Noise Ratio*  
TBFQ – *Token Bank Fair Queue Scheduler*  
TBR – *Target bit rate*  
TDD – *Time Division Duplexing*  
TD-TBFQ – *Time Domain*  
TTI – *Transmission Time Interval*  
TTL – *Time to Live*  
UDP – *User Datagram Protocol*  
UE – *User Equipments*  
VoIP – *Voice over IP*  
VoLTE – *Voice over LTE*  
WCDA – *Wide-Band Code-Division Multiple Access*

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
<b>1.1 Objetivo Geral.....</b>	<b>14</b>
<b>1.2 Objetivos Específicos.....</b>	<b>14</b>
<b>2 DESENVOLVIMENTO.....</b>	<b>15</b>
<b>2.1 Revisão Bibliográfica.....</b>	<b>15</b>
2.1.1 Redes <i>Long Term Evolution</i> (LTE).....	15
2.1.1.2 Arquitetura de rede.....	17
2.1.1.3 <i>Radio Bearer</i> .....	20
2.1.2 Algoritmos de escalonamento.....	22
2.1.2.1 <i>Token Bank Fair Queue Scheduler</i> (TBFQ).....	22
2.1.2.2 <i>Proportional Fair</i> (PF).....	23
2.1.2.3 <i>Priority Set Scheduler</i> (PSS).....	23
2.1.2.4 <i>Channel and QoS Aware Scheduler</i> (CQA).....	25
2.1.3 NS-3.....	27
2.1.3.1 LENA.....	28
<b>2.2 Materiais e Métodos.....</b>	<b>29</b>
<b>2.3 Resultados e Discussões.....</b>	<b>31</b>
<b>3 CONCLUSÕES.....</b>	<b>36</b>
<b>REFERÊNCIAS.....</b>	<b>37</b>

# 1 INTRODUÇÃO

A distribuição de recursos é sempre um grande problema na nossa sociedade. Há muitos séculos, filósofos discutem metodologias de divisão da terra e do capital. Não tão diferente disso é necessária a divisão do canal em redes de comunicação. A primeira divisão dos recursos de rádio é feita pela International Telecommunication Union (ITU), que é uma instituição ligada a Organização das Nações Unidas (ONU).

Um dos objetivos da ITU é alocar vários blocos de frequência para determinadas aplicações e regiões geográficas. A Agência Nacional de Telecomunicações (ANATEL) é o órgão responsável pelas comunicações no Brasil, faz parte da ITU e segue as determinações desse órgão internacional. Cada aplicação de comunicação sem fio usa uma parte do espectro que foi alocado pela ITU/ANATEL. Esse espectro é então limitado para estas aplicações, portanto as redes móveis, que são uma aplicação de comunicação, têm também recursos de canal limitado, assim esses recursos de rádio devem ser divididos entre os usuários.

Quando os usuários solicitam mais recursos que a rede pode fornecer é preciso decidir quais usuários serão atendidos. Essa decisão é feita com base em algum algoritmo, resultando em um agendamento dos momentos em que cada usuário irá usar o canal de comunicação, ou seja, é feito um escalonamento das transmissões de cada usuário. Portanto, os algoritmos de escalonamento são usados para promover a divisão desses recursos de rádio em vários tipos de redes de comunicação.

Esses algoritmos podem dividir os recursos de diversas maneiras, cada maneira pode resultar em diferentes performances da rede. Assim, a rede pode ter seu transporte de dados maximizado ou atender cada usuário de forma semelhante independentemente das diferenças de qualidade do canal de rádio ou ainda tentar fazer as duas coisas ao mesmo tempo. Neste trabalho, será analisado os seguintes algoritmos de escalonamento: *Proportional Fair* (PF), *Token Bank Fair Queue Scheduler* (TBFQ), *Priority Set Scheduler* (PSS) e *Channel and QoS Aware Scheduler* (CQA).

Como os algoritmos de escalonamento podem ser usados em vários tipos de rede, o estudo deles é de extrema importância. A rede *Long Term Evolution* (LTE) é uma rede que já está implantada e existem várias publicações sobre ela. Também já

estão desenvolvidos vários softwares de simulação para essa rede. Por esses motivos a rede LTE foi escolhida para simular vários algoritmos de escalonamento usando o simulador de redes *network simulator 3* (ns-3).

## 1.1 Objetivo Geral

O objetivo deste trabalho foi realizar uma simulação computacional usando o simulador ns-3 de um grupo selecionado de algoritmos de escalonamento que são utilizados em redes LTE com foco em estudar quais escalonadores garantem mais Qualidade de Serviço (QoS) no enlace de *downlink*.

## 1.2 Objetivos Específicos

- Estudar o funcionamento da rede LTE.
- Estudar os algoritmos de escalonamento já implementados no ns-3.
- Construir *scripts* para que sejam realizadas as simulações no ns-3.

## 2 DESENVOLVIMENTO

Para melhor entendimento do leitor, esse trabalho terá uma revisão bibliográfica para introduzir o funcionamento básico da rede LTE e o software ns-3.

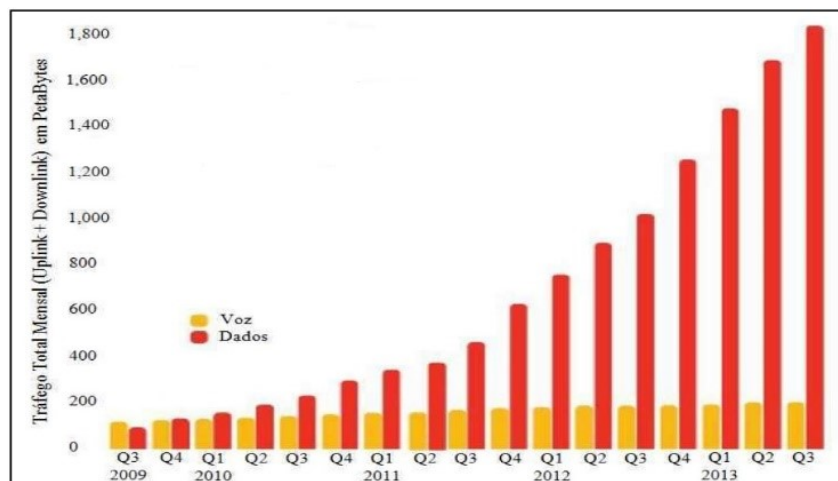
### 2.1 Revisão bibliográfica

#### 2.1.1 Redes *Long Term Evolution* (LTE)

A rede LTE é uma rede móvel, considerada de 4ª geração, sendo proposta pelo *3rd Generation Partnership Project* (3GPP). O principal objetivo no desenvolvimento do padrão dessa rede é atender à crescente demanda de conteúdo da Internet, mantendo a compatibilidade com as redes de 2ª e 3ª geração. Os *releases* 8 e 9 do 3GPP descrevem as especificações e metodologia de funcionamento, enquanto o *release* 10 trata do *LTE-Advanced*, que apresenta algumas melhorias em relação ao LTE [1].

Existem outros padrões de rede de 4ª geração, porém a mais adotada é o LTE, isso é devido a possibilidade de compatibilidade com as redes GSM e WCDMA. A diferença da rede LTE com suas predecessoras é a operação totalmente baseada em IP, enquanto, nas outras, o foco era o transporte de voz sendo possível o transporte de dados, mas de forma secundária. Isto justifica-se pela crescente demanda de dados e pode ser observado nos dados históricos apresentados na Figura 1. Observa-se que a demanda de dados ultrapassou a demanda de voz entre o último quadrimestre de 2009 e o primeiro de 2010.

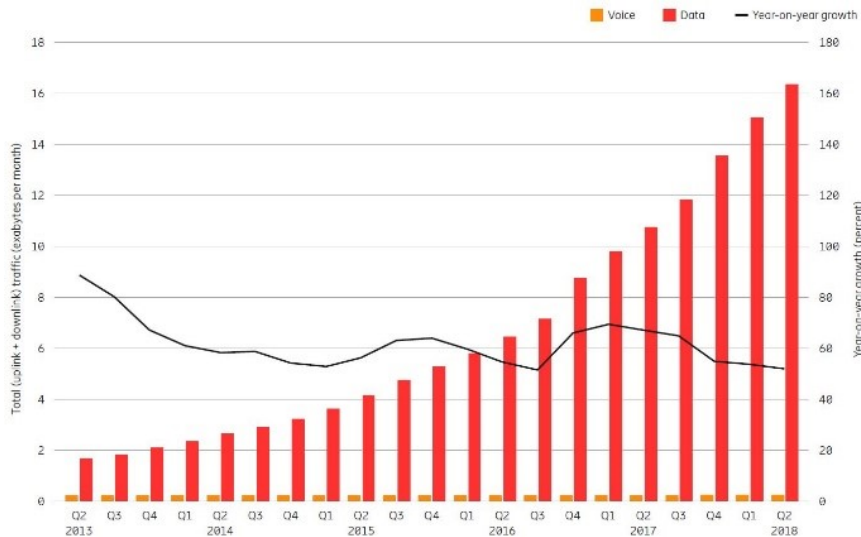
Figura 1 – Evolução do tráfego de dados comparados com tráfego de voz durante os anos de 2009 a 2013.



Fonte: [2].

Já a Figura 2 mostra que essa tendência continuou após 2013.

Figura 2 – Evolução do tráfego de dados comparados com tráfego de voz durante os anos de 2013 a 2018.



Fonte: [3].

Como a rede é constituída somente de comutação de pacotes, a voz pode ser tratada via *Voice over IP* (VoIP), *Voice over LTE* (VoLTE) ou *Circuit Switch FallBack* (CSFB). Tanto no VoIP quanto no VoLTE a voz é amostrada, quantizada e segmentos dessa voz são empacotados para transporte na rede. Para que haja qualidade, são necessários limites de latência e vazão mínima. Em outras palavras, a rede deve implementar Qualidade de Serviço (QoS).

Já no CSFB, o usuário é redirecionado para as redes de 2ª ou 3ª geração, que usam comutação de circuitos, para realização da chamada. Essa é a abordagem mais usada, pois como essas redes legadas ainda estão em funcionamento essa possibilidade é mais fácil de ser implementada [4].

As metas do 3GPP para o LTE são a escalabilidade da largura de banda, redução de custos, simplificação e minimização do número de interfaces, aumentar a eficiência, capacidade e otimização na fronteira da célula [5]. Essas metas começaram a ser discutidas em 2004 [4]. A escalabilidade da largura de banda é necessária devido a fatores regulatórios diversos nos países para ampliar a adoção há várias larguras de banda especificadas. Também é preciso que essas larguras de bandas sejam de vários tamanhos para facilitar a integração com sistemas de 2ª e 3ª geração no que tange a divisão espectral.

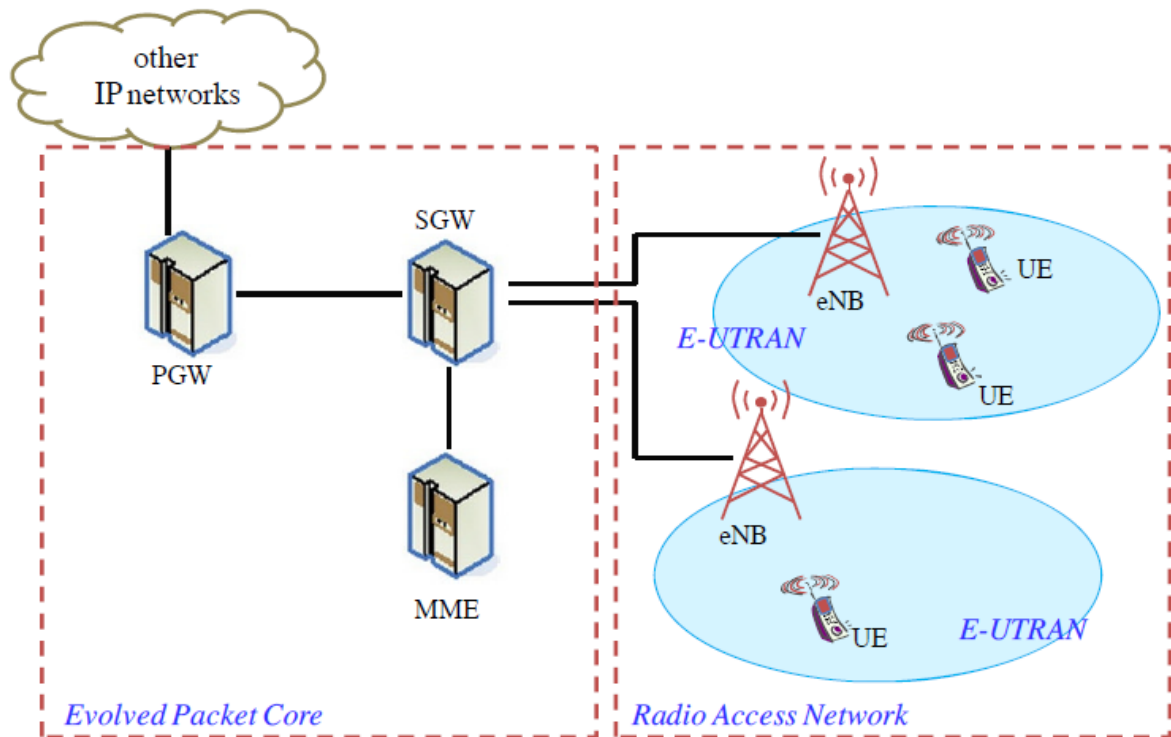


O aumento da eficiência é uma tendência em todas as áreas, não é exclusividade de redes móveis. Seu objetivo é possibilitar que mais usuários compartilhem a mesma banda, tendo maiores velocidades. Os requerimentos propostos para as redes LTE são a latência em torno de 10ms, velocidades de *downlink* de 100 Mbps e 50 Mbps para o *uplink* [4].

### 2.1.1.2 Arquitetura de rede

A arquitetura de rede teve de ser simplificada para atender os objetivos do 3GPP, ela foi subdividida em duas partes (Figura 3), a *Radio Access Network* (RAN) e a *Evolved Packet Core* (EPC).

Figura 3 Arquitetura de rede simplificada.



#### Legend

**PGW:** Packet Gateway  
**MME:** Mobility Management Entity

**SGW:** Service gateway  
**E-UTRAN:** Evolved Universal Terrestrial Radio Access Network

Fonte: [6].

A RAN, que também é denominada *Evolved-Universal Terrestrial Radio Access Network* (E-UTRAN) é composta pelas *Evolved NodeB* (eNB) e pelos *User Equipments* (UE) [6]. Os UE são os dispositivos móveis conectados, no caso telefones, Smartphones, tablets, computadores pessoais, entre outros.

Como previsto pelo 3GPP, esses dispositivos estão se tornando mais variados, hoje existem até carros que são conectados à Internet por redes móveis, principalmente a rede LTE.

A eNB, que também é conhecida como estação rádio-base [4], agrega várias funções que nas redes de 2ª e 3ª geração eram executadas por outros elementos da rede. Ela tem como função a transmissão dos dados, coordenação da interferência intercelular, balancear a carga nas diferentes células, sincronização e mobilidade [5].

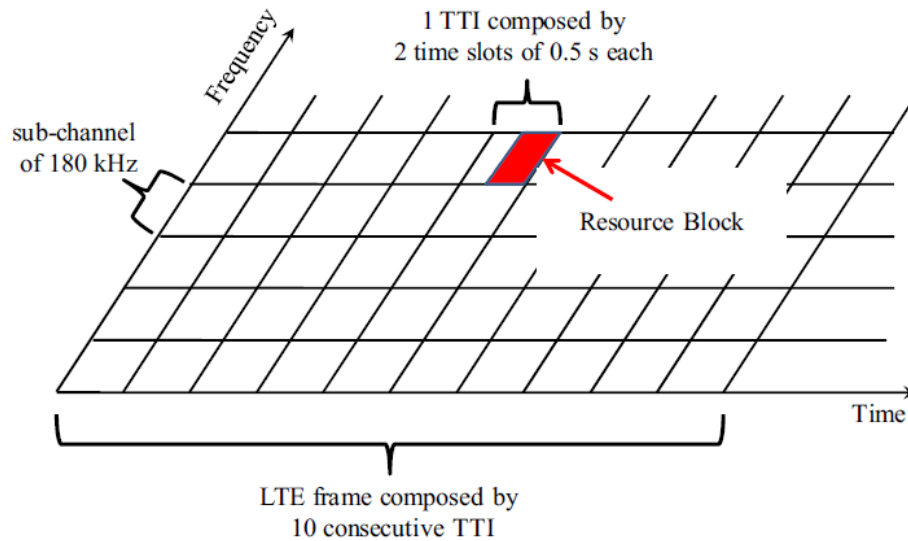
A transmissão de dados é executada na interface de rádio, e é a eNB que controla como será o comportamento do canal, fazendo o escalonamento dos usuários, tanto no *uplink* quanto no *downlink*. Essas funções são conhecidas como *Radio Resource Management* (RRM), que são todas as funções de gerenciamento dos recursos de rádio. Como um dos objetivos do LTE é a interoperabilidade, o canal de rádio opera tanto no modo de *Frequency Division Duplexing* (FDD) quanto no modo de *Time Division Duplexing* (TDD). No FDD, o canal de *downlink* e de *uplink* operam simultaneamente em frequências diferentes, já no TDD ambos operam na mesma frequência, mas em momentos distintos tendo um *slot* para tráfego de *downlink* e outro para tráfego de *uplink*.

O canal de rádio pode ter 1,4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz e 20 MHz de largura de banda [5] tanto no *uplink* quanto no *downlink*. No *downlink* a técnica de modulação usada é o *Orthogonal Frequency Division Multiplexing* (OFDM), em que portadoras ortogonais são usadas para maximalizar o uso do espectro e cada portadora pode carregar informações diferentes para cada usuário. Enquanto no *uplink* é usado o *Single Carrier Frequency Division Multiple Access* (SC-FDMA) em que o espectro das portadoras não se sobrepõe, isso simplifica os transmissores e a recepção de cada portadora.

As várias portadoras do OFDM podem ser atribuídas a transmissão *downlink* destinadas a usuários diferentes, para isso são definidos *resource blocks* (RB). O RB é formado por 12 subportadoras que tem 180 KHz de largura de banda no total e duração de um *Transmission Time Interval* (TTI) que equivale a 1 ms.

O RB é usado como unidade de alocação de recurso, ou seja, os RB são alocados a usuários diferentes pelo mecanismo de escalonamento. A Figura 4 ilustra como são os RB de uma forma visual intuitiva.

Figura 4 – Grade de recursos no diagrama Tempo-Frequência.



Fonte: [6].

Além disso, a qualidade desse canal é reportada pelo UE através do *Channel Quality Indicator (CQI)*, usando os canais de controle. Dessa forma, podem ser escolhidos diferentes esquemas de modulação e codificação, isto é, o padrão prevê o uso de *Adaptive Modulation and Coding (AMC)*.

No *release 10* o 3GPP propõe o uso das modulações BPSK, QPSK, 16QAM e 64QAM para o *downlink*. Além da *Hybrid Automatic Repeat reQuest (HARQ)*, que combina as metodologias de *Forward Error Correction (FEC)* e *Automatic Repeat reQuest (ARQ)*. No HARQ o receptor tenta primeiro corrigir os erros de recepção, em seguida se não for possível a correção ele envia uma mensagem avisando o transmissor que não pode recuperar a mensagem, então o transmissor envia somente uma redundância diferente, para então o receptor usar a nova redundância em conjunto com a mensagem corrompida.

Como o LTE é definido para várias larguras de banda, cada uma delas tem um número diferente de subportadoras, resultando em várias quantidades de RB. A Tabela 1 demonstra quantos RB existem para cada largura de banda. Para a alocação desses recursos os RB são agrupados formando *Resource Blocks Groups (RBG)*, que podem ser constituídos de várias formas diferentes.

Tabela 1 – Quantidade de RB e RBG por largura de banda.

Largura de Banda (MHz)	1,4	3	5	10	15	20
RB	6	15	25	50	75	100
RBG	1	2	2	3	4	4

Fonte: Adaptado de [7] e [8].

O padrão LTE determina vários modos que podem ser alocados, porém como o foco desse trabalho é a realização de simulação e o simulador usado implementa apenas a *allocation type0*, somente a quantidade de RB em um RBG definido por essa metodologia será mostrada. Como o número de RBG nem sempre é divisor do número de RB, em alguns casos um RB não é usado no RBG.

A EPC é composta basicamente pela *Mobility Management Entity* (MME), *Serving Gateway* (SGW) e pela *Packet Data Network Gateway* (PGW) [6]. Além disso, em [4] também é incluído o *Home Subscriber Server* (HSS) e o *Policy and Charge Rules Function* (PCRF) como entidades do EPC.

O MME é responsável pela mobilidade do usuário, autenticação, segurança, estabelecimento de conexões, autorização de serviços e é quem faz o *handover* entre sistemas de 3ª e 2ª geração. O SGW atua como roteador entre a rede RAN e os elementos da rede EPC como MME e PGW, encaminhando pacotes entre esses elementos da rede, além de rotear pacotes a redes de 2G e 3G. Alguns fabricantes implementam esses dois elementos da rede em um mesmo hardware físico [4].

A integração com a rede IP é feita pelo PGW, ele também é responsável pela atribuição de endereços IP ao UE e que pode filtrar os pacotes por usuário de acordo com a política do operador da rede. O banco de dados contendo a identidade dos usuários é executada pelo HSS, já o controle de carga da rede é administrado pelo PCRF que também faz a tarifação com base no fluxo de dados.

#### 2.1.1.3 Radio Bearer

A *Radio Bearer* é um canal lógico que liga o UE e a eNB, um único UE pode ter vários canais lógicos de comunicação com a eNB. Como um dos objetivos do 3GPP na elaboração do LTE é que seja possível provisionar QoS, há diferenciação de classes de serviço.

Por isso foram criadas 9 classes de serviço conhecidas como *QoS Class Identifiers* (QCI), cada *Radio Bearer* pode ser classificada de acordo com uma

dessas 9 classes. Essas classes podem ser subdivididas em dois grupos, um que prove a taxa garantida de dados, em inglês *Guaranteed Bit Rate* (GBR) e outra que não oferece GBR. Cada uma das classes é indicada para um tipo de serviço, a Tabela 2 apresenta essas classes bem como suas características de vazão e perda de pacotes.

Tabela 2 – Classes de QoS e suas características de vazão e perda de pacotes.

QCI	Tipo de recurso	Prioridade	Máximo atraso de pacote (ms)	Taxa de perda de pacote	Exemplo de serviço
1	GBR	2	100	$10^{-2}$	Voz interativa
2	GBR	4	150	$10^{-3}$	Vídeo interativo
3	GBR	5	300	$10^{-6}$	Vídeo Armazenado
4	GBR	3	50	$10^{-3}$	Jogos interativos
5	Sem GBR	1	100	$10^{-6}$	Sinalização
6	Sem GBR	7	100	$10^{-3}$	Voz, Vídeo e jogos interativos
7	Sem GBR	6	300	$10^{-6}$	Vídeo Armazenado
8	Sem GBR	8	300	$10^{-6}$	Best Effort
9	Sem GBR	9	300	$10^{-6}$	Best Effort

Fonte: [9].

Um canal lógico padrão é estabelecido durante a conexão, sendo assim o UE ao solicitar um serviço, tem outra *Radio Bearer* estabelecida. Dessa forma, é provisionada a QoS na E-UTRAN, em cada nova conexão uma nova *Bearer* é estabelecida, cada uma delas é classificada com um QCI e o eNB conhecendo essas *bearers* pode tratar elas de maneira diferente.

A forma como serão tratados esses diferentes fluxos, ou seja, quais que usarão efetivamente o canal de comunicação, é selecionado por algum tipo de algoritmo de escalonamento.

## 2.1.2 Algoritmos de escalonamento

Em qualquer rede pode haver momentos em que a disponibilidade de recursos é escassa. Portanto, cria-se uma necessidade de estabelecer qual usuário vai ter acesso aos recursos e a que momento. Os algoritmos de escalonamento são feitos para determinar quais usuários serão servidos a partir de alguma métrica. Cada métrica tem uma finalidade diferente como a de promover justiça, maximalizar eficiência no uso do canal, garantir vazão mínima ou a garantia total de QoS.

Neste trabalho, o enfoque de estudo é no canal de *downlink*, porém os escalonadores também são usados no *uplink*. Como o foco é prover QoS, serão apresentados apenas os escalonadores que são conscientes de QoS ou que são conscientes de canal e de QoS e que estão implementados no simulador ns-3. São eles o TBFQ, PSS e CQA. Além deles, serão apresentados os resultados para o algoritmo PF.

### 2.1.2.1 Token Bank Fair Queue Scheduler (TBFQ)

O TBFQ é um escalonador QoS-aware, que parte do princípio do balde furado (*leaky-bucket*). Segundo [10], nessa metodologia, cada fluxo de um usuário ( $i$ ) é caracterizado pelos parâmetros de taxa de chegada de pacotes ( $t_i$ ), taxa de geração de tokens ( $r_i$ ), tamanho de estoque de tokens ( $p_i$ ) e o número de tokens emprestados ou depositados do banco de tokens ( $E_i$ ). A cada  $k$  bytes de dados, são consumidos  $k$  tokens e, quando esse consumo de tokens é maior que a taxa de geração de tokens, pode-se emprestar tokens do banco seguindo a métrica  $\frac{E_i}{r_i}$ . Acompanhando essa métrica, o usuário que tem uma maior contribuição para o banco de tokens tem mais prioridade de emprestar tokens do banco.

Para limitar que algum usuário empreste tokens demais, existe um limite de débito ( $d_i$ ), assim como também existe um número máximo que um usuário pode pegar emprestado por vez ( $c_i$ ) e um número de tokens que um usuário deve depositar no banco para poder tomar um empréstimo novamente ( $C$ ) quando ele atingir o limite de débito.

Outra utilidade do escalonador é de implementar a taxa de geração de tokens como limitador da velocidade de conexão do usuário. Cada usuário também tem um token pool para usar em *burst* de tráfego sem precisar recorrer ao banco de tokens.

Para o *downlink*, esse escalonador pode ser implementado de duas maneiras, no domínio do tempo ou no domínio da frequência. Quando é implementado no domínio do tempo, todos os RB de um determinado TTI são alocados para o usuário com maior métrica.

Quando a implementação do escalonador é no domínio da frequência, cada RB pode ser alocado a um usuário diferente, considerando o CQI de cada usuário. Quanto melhor o CQI, maior prioridade aquele usuário terá para transmitir no RB. Porém, na implementação atual dos escalonadores, os RB são agrupados em grupos que são denominados *Resource Block Group* (RBG).

#### 2.1.2.2 Proportional Fair (PF)

O algoritmo PF é um algoritmo que tenta trazer justiça mantendo a eficiência espectral. Para isso ele é feito combinando-se os escalonadores *Maximum Throughput* (MT) e *Blind Equal Throughput* (BET). O MT é um escalonador que maximiza a vazão da célula concedendo o RB ao usuário que pode ter a maior taxa nele. Já o BET usa a vazão média passada como métrica, fazendo com que a vazão tenda a ser igual entre os usuários, alocando mais recursos ao usuário com pior qualidade de canal [6].

Essa combinação é feita simplesmente multiplicando as métricas do MT e do BET. Com isso a vazão média passada atua como contrapeso, fazendo com que usuários com canal em más condições sejam servidos por um certo tempo. A métrica final pode ser calculada segundo a Equação 1 [10], em que  $T_j$  é a vazão média passada e  $R_j(k, t)$  é a vazão máxima de determinado RBG baseado no CQI.

$$M_{pf}(t) = R_j(k, t) / T_j \quad (1)$$

#### 2.1.2.3 Priority Set Scheduler (PSS)

O *Priority Set Scheduler* é outro escalonador QoS-aware. Diferente do TBFQ, o PSS atua no domínio do tempo e da frequência, sendo ele dividido em dois algoritmos de escalonamento diferentes, um para domínio do tempo e outro no domínio da frequência [11].

Esse algoritmo provê justiça na rede por meio de uma taxa de bits alvo, em inglês *Target bit rate* (TBR). A decisão é feita primeiro no domínio do tempo, para saber quais são os usuários que irão dividir o *timeslot*, os usuários são divididos em

dois grupos. O primeiro grupo é formado pelos usuários em que a vazão média passada é menor que o TBR, essa vazão média passada é definida como  $T_j$  e a métrica usada para definir a prioridade desse grupo é dada na Equação 2 [10].

$$p_k^1 = 1/T_j \quad (2)$$

Essa métrica dá preferência ao usuário em que a vazão média passada está mais abaixo do TBR. Já o segundo grupo é formado por usuários que tem a vazão média passada maior que o TBR e a métrica é calculada segundo a Equação 3 [10].

$$p_k^1 = R_j(k, t) / T_j \quad (3)$$

Em que  $R_j(k, t)$  é a vazão máxima de determinado RBG baseado no CQI. Dessa forma tem maior prioridade o usuário que tem maior vazão possível, maximizando a vazão na rede para esse grupo.

Os usuários do primeiro grupo são considerados com maior prioridade sobre os do segundo grupo. Portanto, uma lista de prioridade única pode ser formada colocando a lista completa de prioridade do primeiro grupo na frente da lista de prioridades do segundo grupo. Dessa lista combinada, são escolhidos  $N_{mux}$  usuários para serem repassados ao escalonador no domínio da frequência. No domínio da frequência é possível escolher entre duas métricas, o *Proportional Fair* (PF) e o *Carrier over Interference to average* (Colta).

Na métrica PF é usada a mesma métrica do algoritmo PF, sendo que  $T_j$  é substituído por  $T_{schj}$  que é semelhante à vazão média passada, porém ela só é calculada quando o usuário é servido. Assim, o algoritmo entrega aos usuários que melhor podem usar o RB dentro da lista pré-definida pelo escalonador no domínio do tempo.

O algoritmo Colta leva em conta a estimativa da relação *Signal to Interference plus Noise Ratio* (SINR) para determinar a qual usuário se deve alocar o RB. O RB  $k$  é alocado ao usuário  $j$  que tem a maior razão estimativa de SINR sobre a soma das estimativas para os demais RBG. Ou seja, essa técnica escolhe o usuário que pode melhor aproveitar o RB em relação aos demais RB. Portanto, usuários que estão no limite de uma célula experimentando grande interferência em determinado RB podem ter alocado RB em que poderão ter maior eficiência.



A Equação 4 mostra a métrica usada no escalonador, onde  $CoI[j, k]$  é a estimativa de SINR no RB  $k$  para o UE  $j$  [10,11].

$$m(t) = \frac{CoI[j, k]}{\sum_{k=0}^N CoI[j, k]} \quad (4)$$

Ambos os métodos de escalonamento no domínio da frequência são independentes do domínio do tempo. Para o caso de se haver poucos usuários também há uma métrica de peso para ajudar no controle de justiça, ela é representada por meio da Equação 5 [10].

$$w[n] = \max\left(1, \frac{TBR}{T_j(t)}\right) \quad (5)$$

Por fim, o usuário que é escolhido para transmitir no RB  $k$  é o que maximiza o produto da métrica do domínio da frequência pela métrica de peso. Dessa maneira, esse algoritmo garante que o usuário que está experimentando uma baixa qualidade de canal tenha uma taxa aproximada ao seu TBR [10].

#### 2.1.2.4 Channel and QoS Aware Scheduler (CQA)

Este escalonador foi proposto para maximizar os recursos de rádio enquanto provê QoS de duas maneiras. Os demais algoritmos que foram apresentados só promovem QoS no que tange a taxa de dados, porém, há também fluxos de dados que são sensíveis ao atraso de fila que os pacotes experimentam. Como aplicações interativas como chamadas de voz e/ou vídeo tem características que exigem baixo atraso, o escalonador que leva esses parâmetros na decisão pode obter melhor característica [12].

Visando atender ambos esses parâmetros, o escalonador foi desenvolvido para atuar tanto no domínio da frequência quanto no domínio do tempo. De forma semelhante ao escalonador PSS, primeiramente a decisão é tomada no domínio do tempo, posteriormente, no domínio da frequência. Os RB são novamente agrupados em RBGs e eles tem diferentes quantidades de RB dependendo da largura de banda e tipo de alocação.

No domínio do tempo, o escalonador seleciona a cada TTI dentre o universo de usuários os que ainda não atingiram o MBR programado na *Radio Bearer*. Esses

usuários são agrupados de acordo com o atraso de cabeça de fila (HOL), isso é feito pelo cálculo da métrica mostrada na Equação 6 [12].

$$m_{td}^j = \frac{d_{HOL}^j}{g} \quad (6)$$

Em que,  $d_{HOL}^j$  é o atraso de cabeça de fila do fluxo  $j$  e  $g$  é o parâmetro de agrupamento que determina a granularidade desses grupos, ou também quantos fluxos serão considerados pela métrica de escalonamento do domínio da frequência.

Esse parâmetro afeta a importância do  $d_{HOL}^j$  na decisão de escalonamento. Se  $g$  é pequeno, reduz a diversidade de usuários, ou seja, diminui os ganhos do escalonador no domínio da frequência. E maior a importância de  $d_{HOL}^j$  no processo de escalonamento. Selecionar esse parâmetro pequeno é útil quando os usuários têm condições de canais semelhantes. Logicamente, se  $g$  é grande, o  $d_{HOL}^j$  tem menor impacto na métrica final e, portanto, os ganhos do escalonador do domínio da frequência tendem a ser maiores [12]. Esse parâmetro deve ser escolhido de acordo com a capacidade da rede e a quantidade de usuários esperados.

Em seguida, os grupos com maiores métricas são repassados ao escalonador do domínio da frequência até que todos os RBGs tenham fluxos atribuídos. No domínio da frequência um dos  $k$  RBG é atribuído ao fluxo  $j$  que tem a maior métrica, como pode ser visto nas Equações 7 e 8 [12]:

$$m_{fd}^{(j,k)} = d_{HOL}^j \times m_{GBR}^j \times m_{ca}^{k,j} \quad (7)$$

em que,

$$m_{GBR}^j = \frac{GBR^j}{\overline{R^j}} \quad (8)$$

Sendo  $GBR^j$  a taxa de dados garantida estabelecida nos parâmetros da *Radio Bearer* e  $\overline{R^j}$  a vazão média passada calculada com média móvel. A função de  $d_{HOL}^j$  e  $m_{GBR}^j$  é o provisionamento de QoS, já a  $m_{ca}^{k,j}$  tem função de dar conhecimento das condições do canal, fazendo com que possa ser maximizada a capacidade do canal e pode ser escolhida dentre duas metodologias. A primeira delas é baseada no algoritmo PF denotada por  $M_{pf}$ , já a segunda é a  $M_{ff}$  que é algoritmo Colta.

Portanto,  $m_{ca}^{k,j}$  se assemelha com o algoritmo PSS no domínio da frequência podendo ter métricas iguais [11, 12].

### 2.1.3 NS-3

Softwares de simulação são grandes aliados de pesquisadores e engenheiros, pois reduzem custos com a montagem de um modelo real. No caso desse trabalho, uma infraestrutura de rede. Também tem papel vital em termos de escalabilidade, reprodutibilidade, rápida prototipagem e educação [13].

O ns-3 foi desenvolvido com o objetivo de melhorar o realismo, ou seja, quanto os modelos empregados no simulador se assemelham com a realidade. A linguagem escolhida para o simulador foi C++, pois linguagens mais abstratas podem se distanciar da implementação real, causando a divergência dos resultados autênticos.

Outro benefício do ns-3 é a possibilidade de ser emulado em dispositivos reais possibilitando a fácil transição de simulação para teste em dispositivos físicos [14]. Um exemplo dessa possibilidade de integração é o ORBIT, uma plataforma de teste que usou quatrocentos computadores, em um ambiente grande, com interfaces de rede usadas na simulação [13].

O ns-3 suporta a simulação de redes IP e de redes não IP, porém seu foco é a simulação de redes sem fio e redes IP [14]. É um simulador completo, modela todos os elementos de rede possíveis, incluindo computadores, dispositivos de rede e canal de comunicação, além de modelar também, vários protocolos de rede publicados em RFCs e alguns que ainda estão em fase de pesquisa. Os modelos de pacotes podem conter somente os cabeçalhos ou também a carga útil do pacote. Quando usa o modelo somente com os cabeçalhos, o tamanho dos dados é armazenado em uma variável, dessa forma não há desperdício de memória durante a simulação.

Além de implementar vários modelos de rede, é possível também salvar em um *log* de tudo o que ocorre na simulação. Existem vários métodos de criar esse log. Um método é criar um arquivo “pcap” que pode ser aberto em qualquer programa de captura de rede como o Wireshark, sendo assim plausível analisar detalhes da simulação [13].

Existem outras formas de *log*, que são usadas para armazenar os resultados das simulações. Esses resultados podem ser revisados e estudados a partir de

outros programas. Essa metodologia foi abordada pois não há necessidade de se reinventar ferramentas de análise reduzindo assim a complexidade do simulador.

Um dos objetivos dos criadores do ns-3 é facilitar o seu uso. Para isso acontecer, é necessário que linguagem de programação seja amplamente conhecida, como é o caso do C++. Além disso, foi feito de maneira que a depuração de uma determinada simulação seja fácil [13]. Existe a possibilidade de utilizar a linguagem Python empregando API (*Application Programming Interface*), isso permitiu o rápido crescimento da comunidade de usuários e desenvolvedores.

Como é um simulador aberto, a manutenção e adição de ferramentas é feita pela própria comunidade de usuários e desenvolvedores, a cada três meses, é publicada uma nova versão contendo novos modelos [14]. A comunidade pode implementar modelos livremente, contudo para ser listada nas versões é preciso validação, seja com outros softwares de rede ou implementação real. O processo de se adicionar ferramentas e modelos segue uma rigorosa metodologia com práticas de codificação, revisão de codificação e teste. As metodologias aprovadas são acrescentadas a versão oficial do ns-3, tendo seu código fonte incluído na biblioteca do simulador [13].

#### 2.1.3.1 LENA

LENA é um projeto de código aberto, baseado no ns-3, que tem como objetivo possibilitar o projeto e avaliação de performance de vários parâmetros de redes LTE. É um produto orientado a empresas que vendem micro e macrocélulas.

O desenvolvimento do projeto é de código aberto a comunidade para propiciar a adoção e por conta disso teve contribuições tanto do setor industrial quanto de pesquisadores acadêmicos [15]. A versão mais estável do simulador é incluída na versão oficial do ns-3, sendo que toda documentação dos módulos está incluída no ns-3.

O simulador é implementado com o uso de várias classes, cada uma delas implementando uma parte do LTE. Dessa forma, quando se constrói um código para usar o simulador, é feito de maneira a usar-se dessas classes, e como são várias, pode-se usar uma classe diferente para implementar peculiaridades, por exemplo no caso desse trabalho pode-se criar uma classe que implemente um escalonador diferente do que já está presente no simulador.

## 2.2 Materiais e Métodos

O estudo da rede LTE foi realizado principalmente com base nas referências [6] e [10] e tendo o suporte das demais referências presentes nesses artigos. Assim, o estudo da rede foi feito de maneira mais objetiva na parte que se relaciona com o enlace de rádio e seu uso, a fim de melhor entendimento dos mecanismos que são realizados os escalonamentos e a forma de obtenção das variáveis de entrada desses algoritmos.

O software de simulação ns-3 foi escolhido por ser um software livre, que já continha alguns módulos que realizam a rede LTE e o escalonamento. A versão instalada foi o ns-3.29 no sistema operacional Ubuntu 18.04.

O simulador dá como resultado vários arquivos de texto contendo diversas informações. Para analisar essas informações e plotar, foi usado o MATLAB® 8.2 R2013b. Nele foram importados esses diversos arquivos e as informações presentes neles foram organizadas e plotadas em vários gráficos.

Na simulação, foram usadas duas formas de obtenção de vazão, na camada RLC do LTE e na camada de rede através do FlowMonitor [16]. Neste trabalho, foi usado os dados de perda de pacotes extraídos do FlowMonitor e os dados de vazão são da camada RLC do LTE.

O código usado para simular foi construído de forma que se possam passar os parâmetros de simulação no momento da execução como argumentos. Esses parâmetros podem ser o número de pares de usuários, a distância desses pares, o tempo de simulação e o escalonador a ser considerado. Também são passados alguns parâmetros adicionais que são únicos para todas as simulações como: a potência de transmissão da eNB e do UE, a figura de ruído de cada equipamento e o modelo de propagação a ser usado. Todos os parâmetros usados nas simulações estão na Tabela 3.

Tabela 3 – Parâmetros usados na simulação.

Modelo de perdas de propagação	<i>Friss</i>
Duplexação	FDD
Largura de banda de <i>uplink/downlink</i>	10MHz
Potência de transmissão da eNB	40dBm
Modelo de mobilidade	Posição constante
Agregação de portadoras	Desativada
HARQ	Desativada
Tempo de Simulação	120,1s

Fonte: Autor.

Os usuários foram colocados em pares. Os que formam um par foram colocados bem próximos a fim de estarem tendo as mesmas condições de canal. Um desses usuários irá receber um tráfego característico de uma chamada de voz e o outro recebe um tráfego *Best Effort* de 6 Mbps.

Para se assemelhar a um tráfego gerado no *downlink* por uma chamada de voz (VoIP/VoLTE), foi criado um gerador de tráfego UDP (*User Datagram Protocol*) na Internet. Ele tem uma ligação com o PGW de 100 Gb/s, essa velocidade é para assegurar que não haverá interferência desse link nos resultados da simulação.

No VoLTE, o codificador usado é o *Adaptive Multi-Rate Wideband* (AMR-WB) que tem uma taxa de bits de 18050 kbps para funcionamento ideal, ele gera pacotes de dados a cada 40 ms [17]. Para emular o comportamento desse codificador o gerador de tráfego UDP é configurado para enviar um pacote a cada 40ms, esse pacote tem tamanho de 722 bits. Como o *overhead* de um pacote UDP é de 12 bytes, foi inserido um pacote de 103 bytes no simulador. Como esse tráfego requer QoS, foi selecionado o QCI 1 na *Bearer* correspondente aos usuários desse grupo. Essa configuração impõe um tráfego de aproximadamente 20 Kbps.

Já para gerar o tráfego *Best Effort*, foi criado também um gerador de tráfego UDP. Ele é configurado para gerar um pacote de tamanho de 1500 bytes e envia um pacote a cada 2 ms. Como o tráfego é do tipo *Best Effort*, o QCI da *Bearer* foi escolhido como 7.

Os pares de UE são dispostos em um raio de 15 Km da eNB, sendo que o UE que está mais perto da eNB está ao lado dela e o que está mais distante está na marca de 15 km de distância. A simulação foi executada para 12, 24 e 48 usuários, de forma com que eles sempre estejam igualmente espaçados dentro do raio de simulação.

Os escalonadores simulados são:

- PF;
- PSS com a métrica do escalonador PF no domínio da frequência ( $PSS_{PF}$ );
- PSS com a métrica Colta no domínio da frequência ( $PSS_{Colta}$ );
- CQA com a métrica do escalonador PF no domínio da frequência ( $CQA_{PF}$ );
- CQA com a métrica Colta no domínio da frequência ( $CQA_{Colta}$ );
- TBFQ no domínio da frequência (FD-TBFQ);
- TBFQ no domínio do tempo (TD-TBFQ).

Para simular todos os algoritmos foi criado um *bash script* no Ubuntu que realiza a simulação para determinado cenário e salva os resultados de cada algoritmo de escalonamento em um diretório separado.

## 2.3 Resultados e Discussões

Para todas as quantidades de usuários e em todos os escalonadores os usuários que têm como tráfego com chamada de voz, a vazão média durante a simulação foi igual a taxa de envio do codificador, atendendo o tráfego de dados de voz completamente.

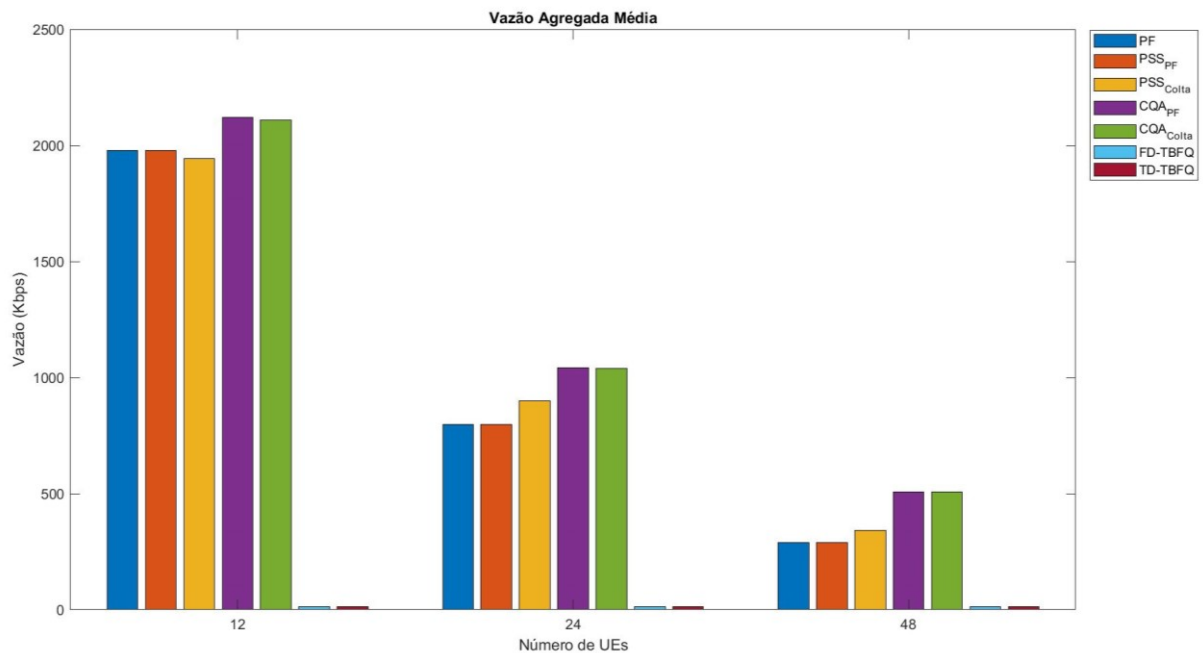
Para o TBFQ, tanto no domínio do tempo quanto para o domínio da frequência, ocorreram problemas com os dados coletados para os usuários de tráfego sem prioridade. Os fluxos desses usuários estavam com valores próximos de zero.

Quando se observa o comportamento no tempo desses fluxos, percebe-se que somente no início da simulação eles conseguiram transmitir pacotes. Isso ocorreu nos primeiros 250ms da transmissão, pois todos os usuários no início de suas transmissões têm seu *token pool* cheio, portanto, recebem RB para transmissão e assim que esse pool esvazia eles não têm mais direito. Esse

comportamento foi similar tanto para 12, 24 e 48 usuários. Portanto, os gráficos que tratam de atraso e perda de pacotes do TBFQ não serão discutidos.

O Gráfico 1 demonstra o comportamento da vazão média agregada de todos usuários, que foi calculada somando-se as vazões médias de cada usuário.

Gráfico 1 – Vazão média agregada X número de UEs.



Fonte: Autor.

Nota-se que para os escalonadores PF, PSS e CQA, quanto mais usuários compartilham os recursos da rede, em média, cada um fica com menos recursos. Já para o escalonador TBFQ os resultados, quando se considera os usuários do tráfego *Best Effort*, são próximos a zero, como explicado anteriormente.

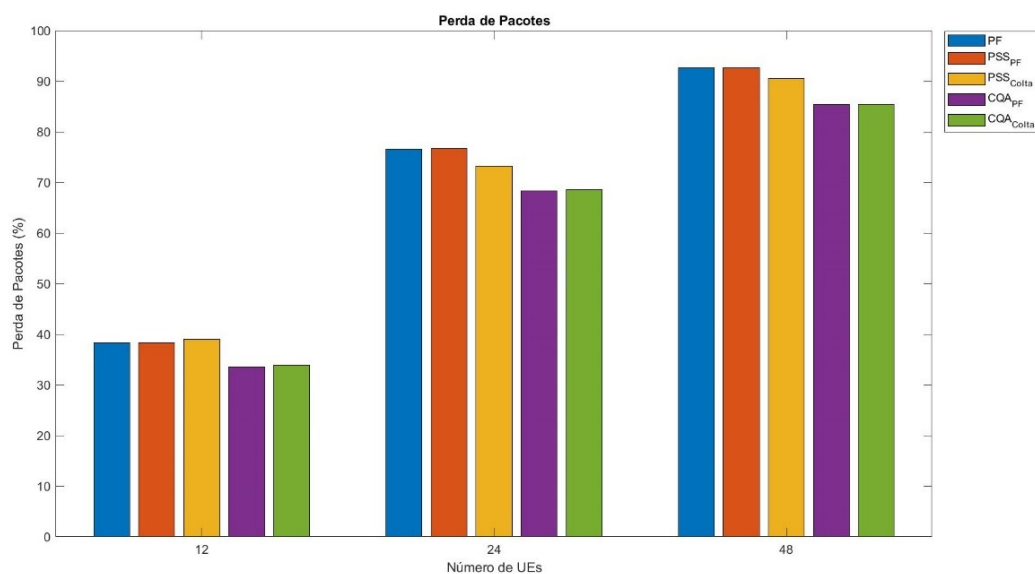
O comportamento entre PF e PSS<sub>PF</sub> é similar, pois o tráfego gerado pelos usuários do tipo GBR é pequeno, enquadrando a grande parcela do tráfego na métrica da Equação (3). No simulador, o parâmetro TBR do algoritmo PSS é controlado pelo GBR. Dessa forma, a lista formada pelos usuários é encaminhada ao escalonador PF.

Comparando a vazão média dos escalonadores PF/PSS<sub>PF</sub> com o escalonador PSS<sub>Colta</sub> observa-se que o Colta tem um melhor desempenho, isto se dá porque a métrica Colta seleciona o RB que o usuário poderá melhor aproveitar.



Assim ocorre um pequeno aumento da vazão em relação ao PF/PSS<sub>PF</sub>. Já o algoritmo CQA tem melhor vazão média, isso se deve possivelmente ao menor descarte de pacotes por expiração de TTL (*Time to Live*). Esse descarte é menor, pois os pacotes que têm maior atraso de HOL (*Head of Line*) são considerados de maior prioridade. O menor atraso pode ser visualizado no Gráfico 2 e Gráfico 3, apresentando que a perda de pacotes é reduzida.

Gráfico 2 – Atraso médio X número de UEs.



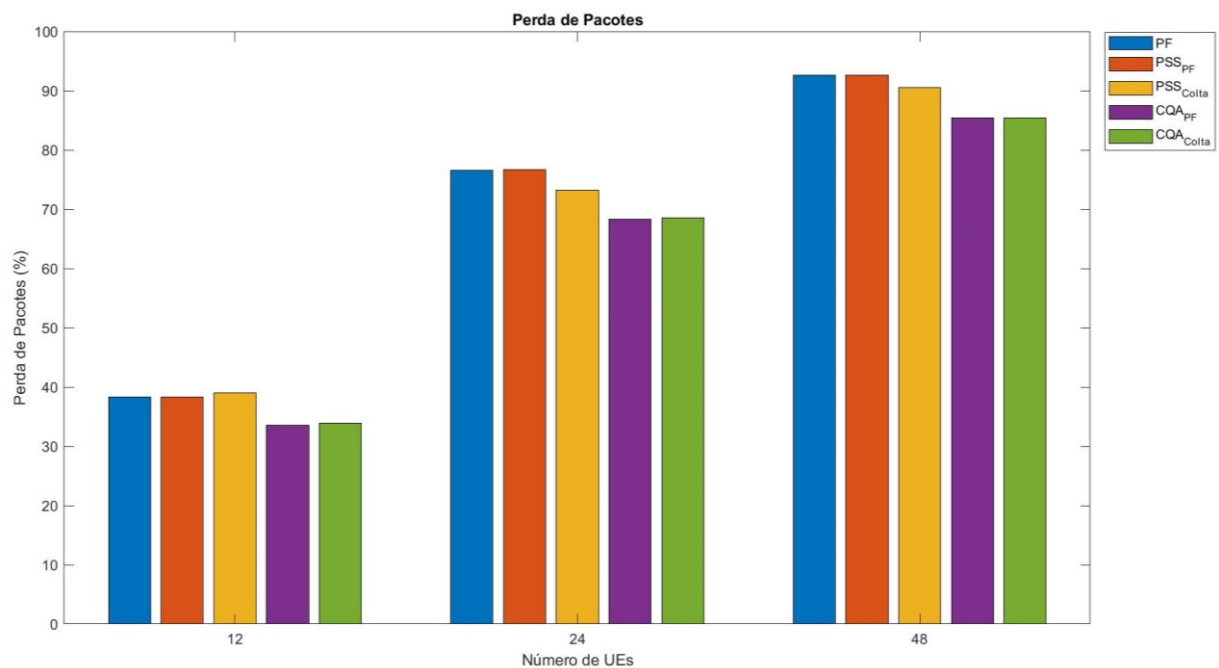
Fonte: Autor.

O atraso médio que um pacote sofre entre a fonte de tráfego e o UE, mesmo para usuários que tem tráfego de voz serem atendidos, o usuário de um tráfego sem garantias, pode ficar insatisfeito com uma demora excessiva no início da reprodução e com pausas durante a reprodução.

Comparando de forma geral o desempenho de atraso, nota-se que o CQA tem o menor deles, pois é o único que leva em conta o atraso gerado na fila de transmissão. Esse atraso para todos os algoritmos têm um nível aceitável quando a rede tem 24 usuários, porém, o PF e pelas consequências já abordadas o PSS<sub>PF</sub> também apresenta um crescimento excessivo do atraso (Gráfico 2), enquanto o PSS<sub>Colta</sub> e o CQA tiveram um crescimento de atraso mais modesto.

Já quando olhamos a perda de pacotes (Gráfico 3), é possível notar que todos os escalonadores têm um comportamento similar, quando ocorre o crescimento do número de UEs há um aumento da perda. Porém, esse aumento não é similar para todos os algoritmos. Nota-se que o PSS<sub>Cola</sub> tem menor aumento das perdas quando comparado ao PF/PSS<sub>PF</sub>. Isto também pode ser visto quando comparamos PF/PSS<sub>PF</sub> com o CQA.

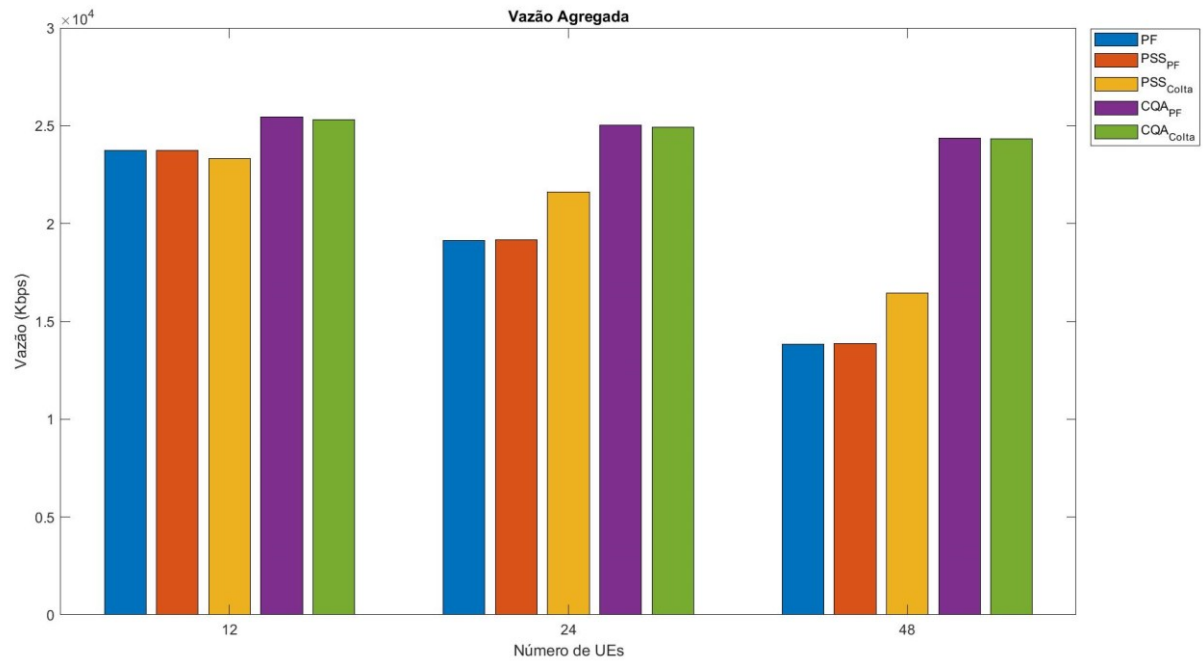
Gráfico 3 – Perda de pacotes X número de usuários.



Fonte: Autor.

Com relação a eficiência geral da célula, percebe-se que com o aumento dos usuários os algoritmos PF e PSS tem tendência de redução da vazão agregada. Já o CQA mantém essa vazão bem próxima, ocorrendo a redução, mas como visualiza-se no Gráfico 4 a redução no caso do CQA é desprezível. Ou seja, esse escalonador consegue manter a eficiência do uso dos recursos mesmo quando o número de usuários cresce.

Gráfico 4 – Vazão Agregada X número de UEs.



Fonte: Autor.

### 3 CONCLUSÕES

O TBFQ não é um escalonador bom para lidar com tipos de tráfegos diferentes, pois ele foi originalmente criado para servir a redes com tráfego em rajadas e que possuem taxa de bits alvo, tipicamente encontrada em aplicações que usam streaming de vídeo, não havendo diferenciação de prioridade entre valores de QCI. Portanto, ele puramente não pode ser usado em uma rede real, mas sim pode fazer parte de um escalonador, como foi feito com as métricas PF e Colta nos escalonadores PSS e CQA. O mecanismo de banco de tokens é útil em aplicações em rajadas, fazendo com que tenha uma vazão mínima garantida e ao mesmo tempo tenha capacidade de suportar *burst* de tráfego.

O cenário simulado não possibilitou ver diferença entre o escalonador PSS<sub>PF</sub> e o PF, pois os usuários que geravam tráfego na rede não têm TBR, portanto a métrica no domínio da frequência é a mesma para os dois fluxos. Apesar disso todos os usuários são servidos e, também, os usuários com tráfego de voz, que tem menor vazão média que os demais usuários.

Dos escalonadores analisados o que obteve melhor comportamento é o CQA, tanto usando a métrica PF quanto a métrica Colta. Ele faz com que o tráfego de voz tenha o menor atraso fim a fim, bem como todos usuários sejam atendidos e de forma justa quando se trata do algoritmo Colta, além disso maximiza a vazão agregada da célula.

Uma proposta de trabalho futuro é implementar um cenário de simulação que considere dois fluxos diferentes, ambos com QCI com GBR. Isso deve ser feito porque o PSS e o TBFQ na implementação do ns-3 usam o GBR para controlar o Target Bit Rate e o *Token Generation Rate*, respectivamente.

## REFERÊNCIAS

- [1] 3GPP. **Releases**. Disponível em: <<https://www.3gpp.org/specifications/releases>>. Acesso em 17 set 2018.
- [2] ERICSSON. **Ericsson Mobility Report: November 2013**. 32 p. Disponível em: <<https://www.ericsson.com/assets/local/news/2013/11/ericsson-mobility-report-november-2013.pdf>>. Acesso em 18 mar 2019.
- [3] ERICSSON. **Ericsson Mobility Report: August 2018**. 4 p. Disponível em: <<https://www.ericsson.com/assets/local/mobility-report/documents/2018/emr-q2-update-2018.pdf>>. Acesso em 18 mar 2019.
- [4] FILHO, C. A. L. B. **Análise de Desempenho do Protocolo TCP em Redes LTE**. Rio de Janeiro, 2014. 64 p. Dissertação de Mestrado (Pós-Graduação em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro.
- [5] RODRIGUES, C. F. R. **Avaliação da Qualidade e Desempenho de Serviços em Redes LTE**. Lisboa, 2012. 103 p. Dissertação de Mestrado (Pós-Graduação em Engenharia Eletrônica e Telecomunicações) – Área Departamental de Engenharia de Eletrônica e Telecomunicações e Computadores, Instituto Superior de Engenharia de Lisboa.
- [6] CAPOZZI, F. et al. **Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey**. IEEE Communications Surveys Tutorials, vol. 15, n. 2, Second Quarter 2013. p. 678 - 700.
- [7] 3GPP. **LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception** (3GPP TS 36.101 version 8.5.1 Release 8, 2009-04). p.13. Disponível em: <[https://www.etsi.org/deliver/etsi\\_ts/136100\\_136199/136101/08.05.01\\_60/ts\\_136101v080501p.pdf](https://www.etsi.org/deliver/etsi_ts/136100_136199/136101/08.05.01_60/ts_136101v080501p.pdf)>. Acesso em 18 mar 2019.
- [8] LTE Quick Reference. **Resource Allocation Type**. Disponível em: <[https://www.sharetechnote.com/html/Handbook\\_LTE\\_RAType.html](https://www.sharetechnote.com/html/Handbook_LTE_RAType.html)>. Acesso em 18 mar 2019.
- [9] UMAR, M. M. et al. **Downlink Scheduling Algorithms for Real-Time Multimedia Services in 3GPP LTE Mobile Networks**. International Journal of Information Processing and Communication (IJIPC), vol. 6, n. 1. p. 155 – 162, jun. 2018.
- [10] ZHOU, D.; BALDO, N.; MIOZZO, M. Implementation and Validation of LTE Downlink Schedulers for ns 3. *In*: CONFERÊNCIA INTERNACIONAL SOBRE FERRAMENTAS E TÉCNICAS DE SIMULAÇÃO, 6, 2013, Cannes – FR. **Implementation and Validation of LTE Downlink Schedulers for ns-3**. Cannes – FR, 2013. p. 211-218.

- [11] MONGHAL, G. et al. QoS Oriented Time and Frequency Domain PacketSchedulers for The UTRAN Long Term Evolution. *In: IEEE VEHICLE TECHNOLOGY CONFERENCE*, 67, 2008. **QoS Oriented Time and Frequency Domain PacketSchedulers for The UTRAN Long Term Evolution**. 2008. p. 2532-2536.
- [12] BOJOVIC, B.; BALDO, N. A new channel and QoS aware scheduler to enhance the capacity of voice over LTE systems. *In: IEEE INTERNATIONAL MULTI-CONFERENCE ON SYSTEMS, SIGNALS & DEVICES*, 11, 2014, Barcelona – Es. **A new channel and QoS aware scheduler to enhance the capacity of voice over LTE systems**. Barcelona – ES, 2014. 6 p.
- [13] RILEY, G. F.; HENDERSON, T.R. The ns-3 Network Simulator. *In: WEHRLE, K.; GUNES, M.; GROSS, J. Modeling and Tools for Network Simulation*. New York: Springer, 1998, p. 15-34.
- [14] NS-3. **What is ns-3**. Disponível em: <<https://www.nsnam.org/about/what-is-ns-3>>. Acesso em 05 jun 2019.
- [15] CTTC. **LENA**. Disponível em: <<http://networks.cttc.es/mobile-networks/software-tools/lena>>. Acesso em 07 set 2018.
- [16] NS-3. **Flow Monitor**. Disponível em: <[www.nsnam.org/docs/models/html/flow-monitor.html](http://www.nsnam.org/docs/models/html/flow-monitor.html)>. Acesso em 20 jun 2019.
- [17] CIORNEI, S.; BOGDAN, I; SCRIPCARIU, L. HD Voice Modem for End to End Secure Call. *In: Telecommunications Forum (TELFOR)*, 20, 2012, Belgrade - Servia. **HD Voice Modem for End to End Secure Call**. Belgrade – Servia, 2012. P. 580-583.