Marie Skłodowska-Curie Innovative Training Networks (ITN-ETN)

5Gaura

Application-aware User-centric pRogrammable Architectures for 5G multi-tenant networks

H2020 Grant Agreement number: 675806

WP3: Service-Oriented Mechanisms and Architectures

D3.3: RAN Analytics Mechanisms and Performance Benchmarking of Video, Time Critical, and Social Applications

Contractual Date of Delivery:	March 2019
Actual Date of Delivery:	March 2019
Responsible Beneficiary:	UNIKL
Contributing Beneficiaries:	UoY, UPC
Security:	Public
Nature:	Document
Version:	1.1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No *675806.*

Document Information

Version Date: March 2019
Total Number of Pages: 111

Authors

Name	Organization	Email
Mohammad Asif Habibi	UNIKL	asif@eit.uni-kl.de
Meysam Nasimi	UNIKL	nasimi@eit.uni-kl.de
Bin Han	UNIKL	binhan@eit.uni-kl.de
Mikel Irazabal	UPC	mikel.irazabal@upc.edu
Mahmudul Haque Kafi	UoY	mkk518@york.ac.uk

Document History

Revision	Date	Modification	Contact Person
0.9	08.02.2019	First accumulative version created	binhan@eit.uni-kl.de
1.0	04.03.2019	First closed version	binhan@eit.uni-kl.de
1.1	06.03.2019	Proofread by Bin, revised executive summary and conclusion, minor corrections in the lists	binhan@eit.uni-kl.de

Executive Summary

This is the final deliverable of Work Package 3 (WP3) of the 5GAuRA project, providing a report on the project's developments on the topics of Radio Access Network (RAN) analytics and application performance benchmarking. The focus of this deliverable is to extend and deepen the methods and results provided in the 5GAuRA deliverable D3.2 in the context of specific use scenarios of video, time critical, and social applications. In this respect, four major topics of WP3 of 5GAuRA – namely edge-cloud enhanced RAN architecture, machine learning assisted Random Access Channel (RACH) approach, Multi-access Edge Computing (MEC) content caching, and active queue management – are put forward.

Specifically, this document provides a detailed discussion on the service level agreement between tenant and service provider in the context of network slicing in Fifth Generation (5G) communication networks. Network slicing is considered as a key enabler to 5G communication system. Legacy telecommunication networks have been providing various services to all kinds of customers through a single network infrastructure. In contrast, by deploying network slicing, operators are now able to partition one network into individual slices, each with its own configuration and Quality of Service (QoS) requirements. There are many applications across industry that open new business opportunities with new business models. Every application instance requires an independent slice with its own network functions and features, whereby every single slice needs an individual Service Level Agreement (SLA). In D3.3, we propose a comprehensive end-to-end structure of SLA between the tenant and the service provider of sliced 5G network, which balances the interests of both sides. The proposed SLA defines reliability, availability, and performance of delivered telecommunication services in order to ensure that right information is delivered to the right destination at right time, safely and securely. We also discuss the metrics of slice-based network SLA such as throughput, penalty, cost, revenue, profit, and QoS related metrics, which are, in the view of 5GAuRA, critical features of the agreement.

Moreover, with the rapid advancement of technology, and the escalating number of devices communicating in absence of human intervention and involvement, Machine-

to-Machine (M2M) communication is anticipated in many applications. M2M communication is considered as one of the key 5G facilitators, which allows multiple devices to communicate directly with each other. It enables advanced applications and services involved in 5G, namely smart cities, automated vehicles, intelligent industry, etc. As an effort to accommodate the expanding number of M2M traffic, the Long-Term Evolution (LTE) and LTE-Advanced (LTE-A) have defined several QoS metrics for M2M service in Fourth Generation (4G) cellular networks. Nevertheless, 4G cellular networks are still mainly optimized for Human-to-Human (H2H) communication. For both H2H and M2M devices to initially attach to a LTE-A network, a Random Access (RA) procedure over RACH is executed to synchronize with the enhanced Node B (eNB) in uplink. Machine-type communication (MTC) is capable of originating numerous connection requests that engenders explosive load within inadequate time interval. When millions of MTC devices try to access a cellular Base Station (BS) simultaneously using the existing RACH protocol, as the probability of collision among M2M and H2H device increases, a system breakdown or a massive access delay may be resulted. Due to the low delay tolerance of some devices, delay critical applications cannot be served using the RACH protocol in legacy LTE technology. In this work, we present an outline of the LTE networks and discuss the subject associated with the M2M application on LTE. Then we review different RACH overload control mechanisms and the impact of Q-learning in minimizing the RAN congestion and delay. A simulation-based study for contention-based RACH access mechanism is conducted and we approach two different technique to minimize the delay for M2M user group.

Furthermore, in this deliverable, we propose a congestion control mechanism in the context of MEC aiming at reducing RAN congestion. The key idea is to delay latency-tolerant contents from being delivered in case of congestions, until the congestion vanishes. This mechanism is driven by the following contexts: i) the characteristic of data traffic (i.e., delay-tolerant data traffics) and ii) the network conditions (i.e., sudden traffic peaks). More precisely, the proposed mechanism could function within the framework of MEC. It is aiming at real time decision making for selectively buffering traffic, while taking account of network condition and QoS. In order to support a MEC-assisted scheme, the MEC server is expected to locally cache delay-tolerant data

during congestions. This enables the network to have a better control over the radio resource provisioning of higher priority data. To achieve this, we introduce a dedicated function known as Congestion Control Engine (CCE), which captures RAN condition through Radio Network Information Service (RNIS) function, and uses this knowledge to make real time decision for selectively offloading traffic, so that it can perform more intelligently.

Finally, predictability will play a major role in the next generation of cellular systems. Unfortunately, excessive packet accumulation is happening in actual network buffers, which impedes a rapid and predictable packet delivery. Since different traffic flows with different constraints will inevitably share some resources in the system, it is important to ensure that the system remains in a state where all the QoS requests can be fulfilled. In our work, we have focused in implementing, testing and benchmarking the different QoS enablers with each peculiarities at different entities in the 5G network.

Table of Contents

List of Figures

List of Tables

1. Customized Edge-cloud Enhanced RAN Architectures for Machine Type Communications

1.1 Introduction

The 5G communication system fulfills diverse service requirements of all aspects of human life. It enables different kinds of services and various types of vertical industries such as automotive, logistic, health-care, manufacturing, agriculture, etc. In addition, the 5G communication system is expected to support ultimate service experiences such as Ultra High Definition (UHD) video, online gaming, augmented/virtual reality, cloud desktop, etc. in scenarios with ultra-high traffic density, high mobility, extremely high connection density, and wider coverage area. Most existing communication networks are monolithic, where *One-Size-Fits-All* architecture is used to provide services. In order to support various types of 5G applications and fulfill diverse service requirements beyond 2020, the monolithic architecture is no longer sufficient. Therefore, the new concept of Network Slicing is emerged, where a network operator logically divides its network into multiple virtual networks called Slice [1]. All slices of an operator are maintained over the same infrastructure, while each slice has its own features such as QoS, engineering mechanism, architecture and configuration. Network slicing allows operators to partition networks in a structured, elastic, scalable and automated manner in order to reduce total cost, decrease energy consumption, and simplify network functions. As briefly discussed earlier, each use case of 5G communication system needs its own slice that consists of independent functions, requirements, and characteristics. For example, a slice may be dedicated to Critical-Machine Type Communication (C-MTC) such as remote surgery, which is typically characterized by high reliability, ultra-low latency and high throughput. Another network slice may be specified to support water meters reading, which requires a very simple radio access procedure, small payload volume and low mobility. Furthermore, the enhanced Mobile Broadband (eMBB) services may require a separate slice, which is characterized by a large bandwidth in order to support high data rate services such as HD video streaming. All these mentioned and other types of slices open new business opportunities, which require new business models. eMBB in contrast to MBB provides improved data rates, capacity and coverage. Ultra-reliable and Low-latency

Communication (URLLC) refers to critical types of communication supporting very low latency, high reliability as well as small to medium data rates. massive machine-type communication (mMTC) supports the Internet of Things (IoT) use cases with scenarios in which a very large number of (millions to billions) of small devices have to be connected efficiently, e.g. in an energy efficient way.

The requirements and characteristics of various service types in legacy telecommunication networks are almost identical, therefore, most SLAs between service provider and tenant contain same metrics. However, in slice-based 5G networks, every slice needs an individual SLA, which would have unique elements, metrics and structure in comparison to the SLAs of other slices within same network.

As its name implies, the SLA is an official agreement between service provider and tenant or between service providers, based on which the level of rendered service is precisely defined. According to the International Telecommunication Union (ITU), “the SLA is a formal agreement between two or more entities that is reached after a negotiating activity with the scope to assess service characteristics, responsibilities and priorities of every part” [2]. It agrees common understanding about a service with all relevant aspects such as performance, availability, responsibility, etc. Each SLA includes a specific number of elements, which are called metrics. These metrics are used to describe the level and volume of communication services and to measure the performance characteristics of the service objects. Every SLA includes technical, economic and legal statements in order to cover all aspects that are supposed to be agreed between the service provider and the tenant. In order to efficiently measure the performance and describe the level of service, the management of SLA should be automated for the sake of accountability of various network conditions and variety of user patterns over different slices. The automated management function of SLA is achieved through network programmability, virtualization, and controlling functions.

1.2 The Network Slicing Architecture towards 5G Communication

Network slicing, in its simplest description, is to use virtualization technology i.e. Network Function Virtualization (NFV) or Software Defined Networking (SDN) in order

to design, partition, organize and optimize communication and computation resources of a physical infrastructure into multi logical networks for the sake of enabling of variety of services [1]. With deployment of network slicing, a single physical network infrastructure is sliced/partitioned into multiple virtual networks, which is called Network Slice. Each slice can have its own architecture, applications, packet and signal processing capacity, and is responsible for provisioning of specific applications and services to specific end users. Examples of network slices can be: a slice to serve remote control function of a factory, a slice serving for a utility company, a slice dedicated to provide emergency health services, and so on. A slice is consisted of Virtual Network Functions (VNFs), which are appropriately composed to support and build up services that are supposed to be delivered to the end users.

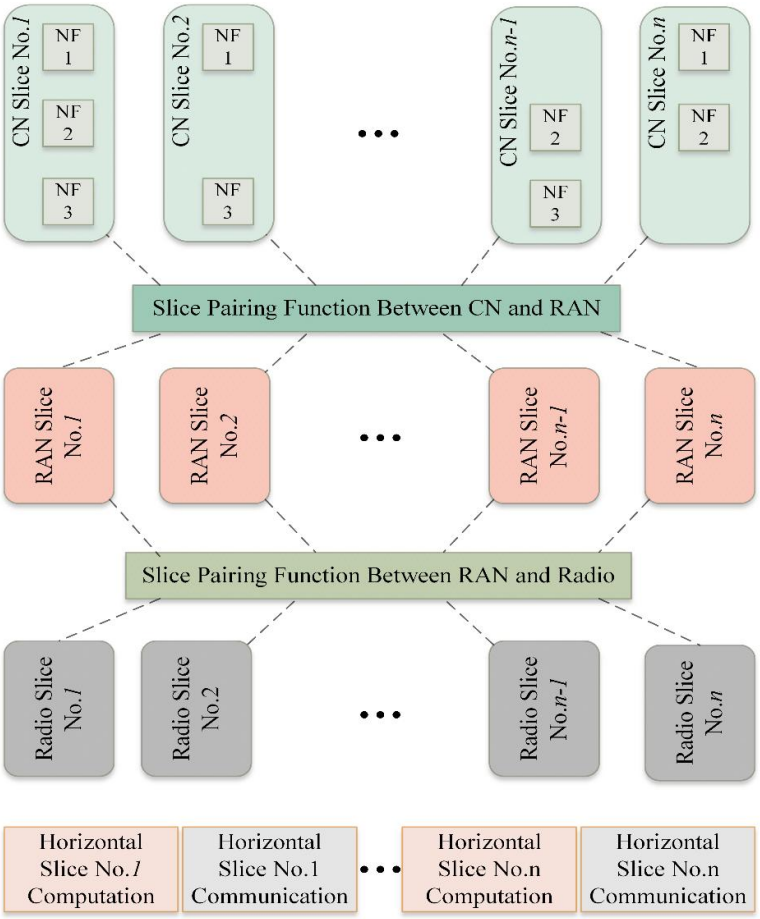

Figure 1: Network Slicing System Architecture

Network slicing deployment includes two main phases: creation and runtime [2]. In the slice creation phase, end user requests a slice from a network slice catalog, the tenant provides the slice immediately upon request. In the runtime phase, different functional blocks whiten each slice, which are already created are now operating and providing service according to the end user's request. Each network resource i.e. NFV and functional block within a specific slice should have its own security mechanisms and must ensure operation within expected parameters in order to prevent access to unauthorized entities. This will lead to guarantee that faults or attacks occurring in one slice are confined to that given slice and will not propagate across slice boundaries. Slicing helps operator to provide new services and applications only by deployment of a slice instead of rolling out a new network, which leads to decrease Capital Expenditure (CAPEX) and saves time.

Network slices are operating on a partially shared infrastructure. This infrastructure is consisted of dedicated hardware i.e. network elements in the RAN and shared hardware i.e. Network Functions Virtualization Infrastructure (NFVI) resources. Network functions running on shared resources are usually instantiated in a customized manner for each slice, however, this approach cannot be applied to the network functions relying on dedicated hardware. Therefore, designing and identification of common functions is one of the key research directions in network slicing. There are two different concepts and scenarios of using of network slicing in communication networks [3], slicing for the purpose of QoS and Slicing for the purpose of infrastructure sharing. Both dimensions of network slicing are described as of following:

- **Slicing for QoS:** The basic idea is to create various slices in order to offer different types of services to the end users, and to assure specific types of QoS requirements within specific slice. An example of this type of slicing can be a slice, which is created to provide service to a specific group of devices considering specific QoS requirements e.g. live video streaming, broadband connection to medical emergency response operation, and so on.

- **Slicing for Infrastructure Sharing:** The fundamental idea of this scenario of network slicing is to virtualize RAN domain of a wireless network, and further

share it among various operators. There is a slice owner and a slice tenant. The owner gives the slice to a tenant based on an agreement. The tenant has overall control on both functions and infrastructure of that slice. This concept of network slicing leads to optimize network cost model for increasing the overall revenue, and meanwhile providing network scalability.

The purpose of network slicing in 5G mobile communication is to allow operators to share infrastructure among each other in flexible and dynamic manner, and to manage resource efficiently considering increased number of devices and massive amount of user traffic. However, a detailed discussion on objectives and motivation of network slicing implementation can be found in [4]. Network slicing helps mobile operators to simplify creation, configuration, and operation of network services. In order to efficiently allocate network resources, two-tier priorities are introduced in [5]. The first tier is Inter-slice Priority, which refers to different priorities between various slices of a network. The priority of each of the slice is defined between owner and tenant of the slice. The second tier is Intra-slice Priority, which is referred to the priorities between different users of a single slice. These priorities are defined between users and service provider.

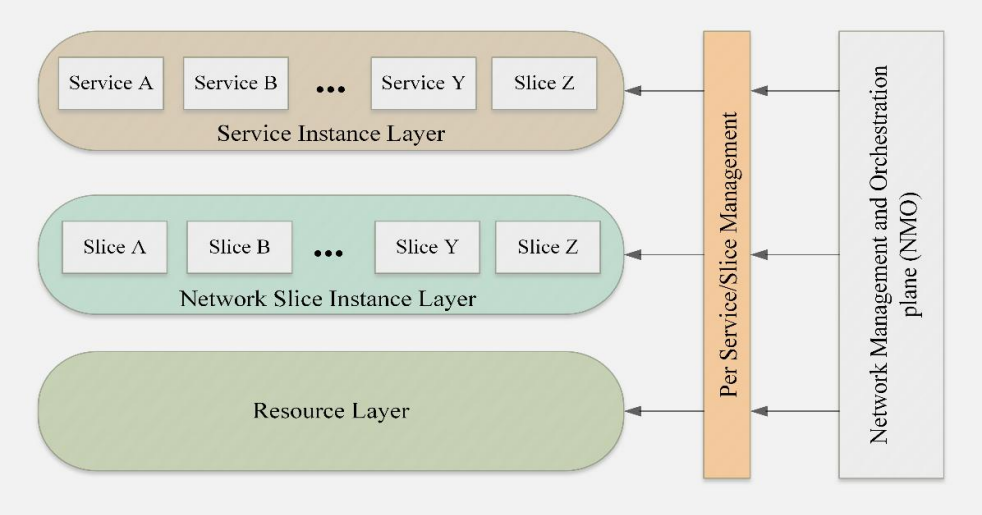

Figure 2: Network Slicing Management Architecture

Slicing can be deployed in two dimensions over 5G communication networks, Vertical Slicing and Horizontal Slicing. Vertical slicing enables vertical industries and services, and focuses on Core Network (CN). However, horizontal slicing improves system performance and increase end user experience, and mainly deals with RAN

architecture. We define in details what is understood by both types of slicing as of following [1]:

- **Vertical Slicing:** The development and deployment of vertical slicing has already started in late 4G and early 5G, and is mostly focusing on the core domain of mobile networks. Mobile broadband networks are sliced vertically in order to serve vertical industries and applications in a more cost efficient manner. It segregates traffic of vertical industries from the rest of general broadband services of mobile network, which leads to simplify traditional QoS engineering problems.

- **Horizontal Slicing:** Increased number of user equipments and massive amount of traffic generated at the edge of mobile network expand network slicing from core domain to the RAN and air interface, which is called horizontal slicing. It is designed to accommodate new trends for scaling of system capacity, enabling of cloud computing, and offloading of computation devices at the edge of mobile networks. Horizontal slicing enables resource sharing among nodes and devices of a network. For example, high capable network devices/nodes share their resources such as communication, computation, and storage with low capable network devices/nodes, which leads to enhance overall network performance.

Both vertical and horizontal slicing are independent from each other. End-to-end traffic flow in a vertical slice is transited between CN and user devices. While in a horizontal slice, it is usually transited locally between two ends of a slice e.g. between a portable device and a wearable device [1]. In vertical slicing, each of the nodes of a network deploys similar functions among slices; however, in a horizontal slice new functions could be added and created at a network node.

Fig. 1 shows the concept and system architecture of network slicing. The architecture consists of CN slices, RAN slices, and radio slices. Each slice in CN is built from a set of Network Functions (NFs); some NFs can be used across multiple slices while some are tailored to a specific slice. There are at least two slice pairing functions, which connect all of these slices together. The first pairing function is between CN slices and

RAN slices, and the second pairing function is between RAN slices and radio slices. The paring function routes communication between radio slice and its appropriate CN slice in order to provide specific services and applications. The pairing function between RAN and CN slices can be static or semi-dynamic configuration in order to achieve required network function and communication. The mapping among radio, RAN and CN slices can be 1:1:1 or 1:M:N, it specifically means that a radio could use multiple RAN slices, and a RAN slice could connect to multiple CN slices.

End-to-end slicing architecture shown in Fig. 1 represents logical decomposition of network slicing, and takes specific network domain functions i.e. CN and radio network domains into account. From the operational perspective, Next Generation Mobile Networks (NGMN) defines that network slicing concept is consisted of three layers: Service Instance Layer, Network Slice Instance Layer, and Resource layer. Each of these three layers are described below and shown in Fig. 2. [6].

- **Service Instance Layer:** It represents end user and business services, which are expected to be supported by the network. Each service is represented by a Service Instance. These services can either be provided by the network operator or by a third party.

- **Network Slice Instance Layer:** A network operator uses a Network Slice Blueprint in order to create a Network Slice Instance. The network slice instance provides the network characteristics required by a service instance. The network slice instance may be shared across multiple service instance, which are provided by a network operator. The network slice instance can be consisted of none, one or more Sub-network Instances, which may be shared by another network slice instance. Sub-network Blueprint is used to create a sub-network instance to form a set of network functions, which runs on the physical/logical resources.

- **Resource Layer:** The actual physical and virtual network functions are used to implement a slice instance. At this layer, network slice management function is performed by the resource orchestrator, which is composed of NFV Orchestrator (NFVO), and of application resource configurators.

Network Management and Orchestration (NMO) plane shown in Fig. 2 provides orchestration and management functions of above mentioned three layers. NMO functions need to allow for the orchestration and management in a per-slice level.

1.3 Service Level Agreement in Telecommunication Networks

Recently, the SLA in telecommunication networks has been exclusively studied. The ITU proposed a generic structure of SLA in multi-service providers telecommunication environment in recommendation E.860 [2]. The proposed SLA defines all QoS-related terms, and furthermore describes the entire procedure of an end-to-end SLA. The European Telecommunications Standards Institute (ETSI) has conducted numerous studies on SLA that are available in [3], [4], and [5]. The reference [3] explores two main aspects of SLA, the development phases and the template, and then discusses further about the contents, technical features, QoS metrics and commitments, charging and billing, and reporting of an SLA. The reference [4] investigates the life cycle of SLA and penalty. The reference [5] studies user demands and various offers, which are provided to the tenant. Moreover, an end-to-end structure of QoS-oriented SLA and a framework of real-time management of SLA of multi-service packet networks are investigated in [6]. The authors presented a monitoring scheme, which is capable to generate revenue by admission flows, and calculates penalty when flows are lost. Although, no study to date has been conducted to explore the SLA between tenant and service provider of slice-based 5G network.

1.4 The Structure of Proposed Service Level Agreement for Slice-based 5G Mobile Network

We introduce and thoroughly describe an end-to-end structure of our proposed slice-based SLA between tenant and service provider of 5G communication system. Moreover, we discuss two types of slice-based SLA, Static SLA and Dynamic SLA, which we think are useful to simplify the operation process of different categories of services over different kinds of slices of a single 5G communication network.

1.4.1 Types of SLAs

The static SLA is predefined SLA, where all metrics, the quality of assured service, legal and financial matters, etc. are predefined between tenant and service provider. When the static SLA starts, the service runs according to the agreement, neither of the parties could bring any change such as increasing the throughput, decreasing the latency, etc. during its lifetime. However, in the lifetime of dynamic SLA, the values of metrics randomly change according to the requirements of the tenant. For example, the tenant of a low latency slice could pay according to the amount of bandwidth, the more he/she spends the bandwidth the more he/she has to pay. Or may require full control on the slice and assured extremely low latency service during remote surgery, but, when the surgery completes, the slice may stops providing the service.

1.4.2 The Proposed Structure

The entire life-cycle of a sliced-based SLA consists of three phases: the creation phase, the operation phase, and the termination phase. In the creation phase, the tenant chooses a service provider that is able to fulfill its requirements. After that both sides agreeing and establishing the SLA, the service starts running over the slice. In the operation phase, the service remains under maintenance and consistently monitored by both sides. In case of any violation of the SLA, a corresponding penalty is executed. In the termination phase, which can be triggered by either violation of agreement or contract expiration, the slice stops providing services and the SLA is terminated. Once decided to eliminate the slice and terminate the SLA, it is recommended to remove all information associated with service configuration, service requirements of the tenant, and service maintenance from the system. However, some tenants or service providers may prefer to archive the information related to their services for a certain period. The detailed procedure of our proposed SLA is depicted in Fig. 3. In the creation phase, the tenant and service provider agree on all terms and conditions of agreement. In the context of this agreement, the tenant is promised to be provided with assured QoS for a certain period of time, which is called the lifetime of SLA. Upon agreement, the service provider and the tenant sign the documents and the SLA is officially established.

The detailed procedure of our proposed SLA is depicted in Fig. 3. In the creation phase, the tenant and service provider agree on all terms and conditions of agreement. In the context of this agreement, the tenant is promised to be provided with assured QoS for a certain period of time, which is called the lifetime of SLA. Upon agreement, the service provider and the tenant sign the documents and the SLA is officially established.

1.4.3 Types of Incidents

In the operation phase, the operator provides and maintains service to the tenant thorough an individual slice, which is acknowledged by the tenant. Meanwhile, a set of QoS metrics of the slice service, such as security, power, throughput, latency, etc., are constantly monitored in real time. The monitoring function of should be accessible to both sides in order to ensure proper service configuration, management, and maintenance. In the context of slice-based SLA, incidents that may happen to a slice, which we categorize into three levels: minor incidents (I_{mi}), major incidents (I_{ma}) and critical incidents (I_{cr}). The I_{mi} indicates a noncritical condition on the slice that, if left unchecked, might cause an interruption to service or degradation in performance. When it occurs, it does not usually interrupt the entire slice, but may damage a small portion of service. The I_{ma} always requires an immediate response, because the integrity of the network is severely at risk such as low/high load of traffic. The I_{cr} indicates a more critical situation on the slice, which is mostly resulted by hardware components failures.

Once an incident occurs, all monitoring metrics shall be automatically checked for troubleshooting and evaluation of contract breach as well as to figure out the types of incident. If the I_{mi} happens to the slice, it should be solved as soon as possible. After solving the the I_{mi}, a penalty P is calculated according to the source and degree of incident, which the service provider is supposed to pay the tenant. In case of I_{mi}, we recommend the service provider and tenant to agree on an individual threshold of each monitoring metric for penalty. In this context, the tenant does not impose any penalty on the service provider despite of an incident, if it can be solved without violating any

of the predefined thresholds. Otherwise, the tenant imposes a penalty P on service provider, and explicitly remind the service provider to solve the incident as soon as possible and furthermore assure the quality of agreed services to the tenant.

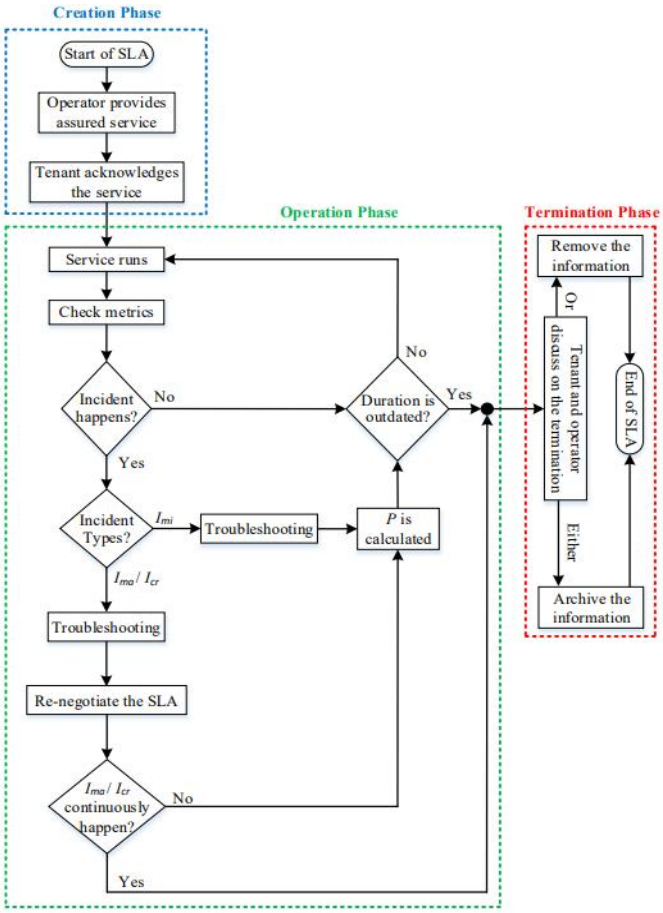

Figure 3: An End to End Structure of Proposed SLA

Assuming that either I_{ma} or I_{cr} happens to the slice, compared to the I_{mi}, I_{ma} and I_{cr} are usually more challenging to solve. Therefore, we recommend there to be a clear agreement in the SLA about how effectively I_{ma} and I_{cr} have to be solved. In case of I_{ma} or I_{cr}, the tenant and service provider can re-negotiate and furthermore optimize the SLA upon major and critical incidents, which helps both sides to avoid further interception to the service.

Furthermore, a long-term track on the occurrences of I_{ma} and I_{cr} is designed, so that the tenant can terminate the SLA in prior to its lifetime and turn to other qualified service providers, in case such serious incidences continuously happen. Otherwise, the SLA

remains valid until that it finally expires its lifetime, where the slice stops running the service, and both service provider and tenant finalize all matters including financial and legal during their business period.

1.5 Metrics of Proposed SLA

One of the main purposes of SLA is to define appropriate and realistic elements for the service that the provider is delivering to the tenant. These metrics are needed to be constantly monitored in order to detect agreement breaches. In this section, we discuss some critical concepts in the slice-based network SLA, including the service availability, penalty, cost, revenue, profit and QoS-related metrics.

1.5.1 Service Availability

The measurement of service availability has a long history in telecommunication industry. It is one of the most important metrics of SLA for both tenant and service provider, and has to be defined as much clear and convenient as possible in order to avoid any misunderstanding between both sides. The International Organization for Standardization (ISO) defines the availability as “the ability of a functional unit to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval, assuming that the required external resources are provided” [7]. In its simplified manner, the availability is the successful transferring of service/data from point A to point B, which is measured in either percentage or unit of time (e.g. hour, mint, etc.). The time a network/slice is not able/delivering service/data to the customer/tenant is defined as downtime. If we consider the total time Th of a service dedicated to a certain slice and the total unavailable time Tu of that specific service to that slice, the service availability can be provided by

$$T_a = \frac{T_h - T_u}{T_h}$$

We divided service availability of a slice into three ranges: high availability (e.g. = 100%), average availability (e.g. ≥ 99.5%), and low availability (e.g. < 99%) in order to

help both the service provider and the tenant evaluate whether the measured metrics of a slice meet, exceed, or fall below the predefined levels in a certain period of time. Both sides should formally agree upon conditional guarantees, e.g. if the average availability of service of a slice in a certain period of time is less than 99%, then the service provider has to pay penalty to the tenant.

1.5.2 Modelling penalty

Most of the time, telecommunications service providers are promising guarantee high level of network performance. These promises are not always kept, therefore, it is recommended for both service provider and tenant to predefine an appropriate penalty value in the SLA. This penalty should be imposed by tenant, when the service provider fails to deliver assured services. In the context of an SLA, some limited levels of incidents or unavailability of service could be acknowledged, but below than those limited levels would not be accepted and the service provider should be punished according to the agreement. Sometimes, the tenant tries to maximize penalty in order to push service provider to ensure proper level of service. On the other hand, service provider may try to convince tenant to accept low level of penalty in the case of failure occurrence, or may try to include some terms in the contract, which lead to decrease level of services. However, smart service providers/tenants would not agree to such terms, which could result them in very large penalties/decreased services. It is worth nothing that “penalty” as the most common term used by both tenant and service provider is not legally correct. If readers are interested to use the most legal terminology for this concept, the “fee reduction” phrase is recommended [7].

1.5.3 Linear and non-linear penalty

We have divide penalty into two types: Linear Penalty and Non-linear Penalty. In linear penalty, tenant charges service provider with a certain predefined amount of penalty when the availability of service falls down by a given predefined level. As depicted in Fig. 2, we have considered 100% as agreed availability, 99.8% as accepted availability, and 98.4% as terminated availability. In between terminated and accepted

availabilities, the penalty should be imposed considering certain predefined value. We have further assumed that by each 0.2% of shortfall in availability, the service provider is charged 5% of penalty. Based on these assumptions, we can analyze from the result shown in Fig. 2 that with 99.6% of availability 5% of penalty is imposed, with 99.4% of availability 10% of penalty is charged, etc. In non-linear penalty, the service provider and the tenant agree on irregular predefined amount of penalty considering different predefined levels of availability. It specifically means that there is no regularity or linear relationship between level of availability and amount of penalty. We have assumed that service provider should be imposed by 5% of penalty, if the availability falls 0.2% below than accepted availability, and then it should be charged with 2% of extra penalty for each of extra 0.1% shortfall until it reaches 99.1% of availability. Moreover, if the level of availability falls below than 99.1%, the service provider should be imposed with 10% of penalty until it reaches 99%, and below than 99% of availability 5% of penalty should be imposed until it reaches the terminated availability. Based on these assumptions, and according to the result shown in Fig. 2, the amount of penalty reaches 25% when the level of availability falls down to 99%, with 98.8% of availability 35% of penalty is imposed, etc.

It is worth nothing that if the availability of service falls below than predefine terminated availability, the tenant may terminate the slice and shift to a different operator, who is capable of providing assured QoS. Moreover, if we compare both linear and non-linear penalties, we can figure out that correlation among level of availability and amount of penalty is the only point that make them different from each other.

Both above mentioned linear and non-linear penalties do not answer to all questions of various scenarios of complicated slice-based network's SLA. In sliced-based network, the demands of tenants are different from slice to slice, on the other hand, each slice would also have its own quality of service requirements, therefore, we need to further investigate various dimensions of penalty such as the importance of the moment when the breach happens to the slice, the total numbers of failures in a certain period of an SLA, duration of each failure, and total duration of all failures in a certain duration of an SLA. In order to find answers to all these questions, we need to further

mathematically develop the concept of penalty in the context of slice-based network SLA [8].

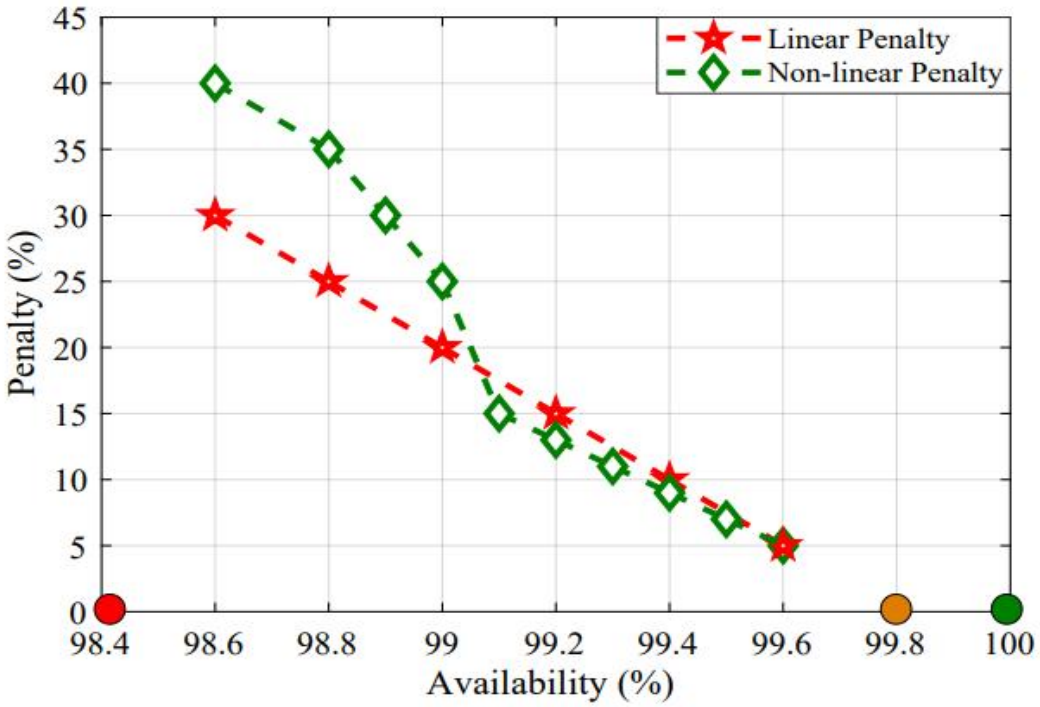

Figure 4: Linear Penalty and Non-Linear Penalty

1.5.4 Modeling profit

The cost models of legacy telecommunication networks are usually built based on CAPEX and Operational Expenditure (OPEX). Both CAPEX and OPEX in classical models are estimated according to the traffic volume, number of base stations, and energy consumption [9]. However, this methodology is no longer appropriate to be used for estimation of cost models of slice-based 5G networks. In sliced networks, each resource can be shared by several slices, and the slicing scheme does also vary from one resource to another. Therefore, OPEX cannot be estimated for the entire slice-based physical network, and we need to define a novel slice-oriented cost model in order to estimate total cost, revenue, profit, and penalty of every single slice, which leads to clarify the SLA between tenant and provider.

As we mentioned in section I, every slice is defined to support a specific use case, and has is own characteristics, QoS mechanisms, and architecture, thus, it is needed to be identified by a subset of Key Performance Indicator (KPI) requirements that is obtained from a given set of KPI requirements k = [k1, k2, . . . , kL] through Virtual Network Function (VNF). In order to estimate the required volume of network resources, we need to consider the VNF implementation (v) and the size of slice (s) (the maximal number of user applications, which can be served by a slice). There are various kinds of network resources, which can be enumerated such as spectrum/bandwidth, power, time, human resources, infrastructure, etc. If we record the required amount of them in a vector r = [r1, r2, . . . , rN], where (N) is the number of resource types. Considering cost of each resource, we can further convert resource requirements into the expenditure, in a similar way as in classical network cost models. So that we have:

$$EXP = EXP(r),$$

$$r = r(k, s, \upsilon)$$

We also know that a certain price must be paid by the tenant for the service that is provided by the slice. Thus, given the service price (p), the slice size (s) and the customer size (c) (the number of user applications requesting service from the slice), the revenue (REV) of a slice can be modeled as:

$$REV = REV(p, s, c)$$

In order to obtain the profit (w) generated by a slice, we subtract the cost from the revenue as shown:

$$\omega = REV(p, s, c) - EXP(r) = \omega(r, p, s, c)$$

It is important to remember that the KPI-to-resource mapping as described in Eq. 10 is very complex and highly dependent on the selection of VNF implementation (v). Nevertheless, as the network operator is responsible for the VNF implementation, it always holds a full knowledge about it. Therefore, in the operators point of view, it is reasonable to assume the function r(k, s, u) as a-priori known.

1.5.5 QoS related metrics

In slice-based 5G networks, each unit of QoS related metrics such as latency, delay, data rate, capacity, throughput, mobility, security, energy consumption, connection density, response time, level of service, etc. are already predefined by standardization organizations i.e. ITU, ETSI, etc. As widely discussed in the literature, the slice-based 5G network supports 1000-fold gains in system capacity, 10 Gbps maximum and 100 Mbps average individual user experience, prolonged battery life of 1000-fold lower energy per bit, 90% reduction in network energy usage, 500 Km/hr mobility for high speed users (e.g. high speed trains), 3-fold spectrum efficiency, perception of 99.99% availability, 100% coverage, and latency from one millisecond to few millisecond [10] [11]. Each slice is created from a subset of these metrics in order to server specific number of users. The business model, the structure of SLA, the specification of QoS, and the level of service are different from slice to slice. Neither tenant nor the service provider are able to bring changes in the volume of these metrics, however, it is possible to decrease or increase the value by multiplying or subtracting the units of these metrics. Therefore, the tenant and service provider are requested to include the volume of these standardized metrics in the SLA according to the standardization organizations.

1.6 Conclusion

We have presented a comprehensive end-to-end structure of SLA between tenant and service provider of slice-based 5G network, which aims to balance the interests of both sides. Our proposed SLA is expected to define reliability, availability, and performance of delivered telecommunication services in order to ensure that right information gets to the right destination at right time, safely and securely. We have also discussed the metrics of slice-based network SLA such as throughput, penalty, cost, revenue, profit, and QoS related ones, which we think are critical during the agreement. In future, we intend to explore different types of slice-based network SLA i.e. shared (an SLA to be shared between specific number of tenants that use the same slice) or hybrid SLA (an SLA that is expected to serve certain tenants first and then serves the authorized

tenants of the same slice). Moreover, this work should be complemented with a deep analysis of some extra QoS related metrics such as tightening the security, decreasing the latency, and increasing the bandwidth.

2. Impact of Q-learning in RACH for delay critical M2M communication over cellular network

2.1 Introduction

The ongoing technological development is growing at a rapid pace and changing the view of wireless communication from traditional human-centric outlook to human independent communication. As a consequence, the number of smart devices is on the rise and there will be around 125 billion smart devices by 2030 (forecasted by IHS Markit). These bulging number of devices contain some that can operate without human intervention as the direct control of all the machines by humans will be difficult. The human centric communications are termed as H2H communication and the existing cellular networks are highly optimized and suitable for this type of communication. On the other hand, the devices compulsory for automated applications communicate with each other through a connection called M2M communication or Device-to-device (D2D) communication [12] [13].

According to ETSI terminology, an M2M device is a mobile terminal capable of transmitting data autonomously [14]. M2M devices may be sensors, actuators, embedded processors, Radio Frequency Identification (RFID) tags, smart meters, etc., [15]. These devices will connect to the wireless network using the short or long-range wireless communication. Some short range wireless networks namely Wireless Personal Area Network (WPAN) technologies (such as IEEE 802.15, UWB, Zigbee, Bluetooth) or Wireless Local Area Network (WLAN) such as Wi-Fi are proposed for M2M as well as long range networks namely cellular networks Global System for Mobile Communication (GSM), General Packet for Radio Service (GPRS), Worldwide Interoperability for Microwave Access (WIMAX), LTE/LTE-Advanced and 5G are the contenders for long range networks for M2M [16]. The cellular networks are convenient for long-range connection for their availability and ubiquity over the installation of a new private radio network however; adjustments are to be made to optimize the legacy cellular network for the coexistence of M2M and H2H communication.

3.1.1 M2M and H2H scenario with new applications and requirements

Conventional cellular networks are mainly designed for H2H services namely voice/video calls, web browsing, video streaming, social networking etc. High data rates, mobility, decent QoS for human satisfaction are some of the basic requirements for H2H communications [17].

On the contrary, M2M communication initiate a very different set of requirements from that of regular H2H communication. M2M have separate QoS requests with different service features including huge device density, small amounts of data, low traffic volume per device, periodic and bursty traffic, majority of uplink traffic, low or no mobility, low power consumption, priority based transmission etc. [18] . In [19] MTC devices are classified into five categories based on traffic shown in Table -1.

Table 1: Classification of MTC traffic

Class Name	Application examples	QoS requirements
H2H	Voice call	Hardly effected by MTC
Low Priority	Consumer electronics	Strict delay
Scheduled	Smart meters	Delay tolerant
High Priority	E-care	Delay sensitive
Emergency	Seismic Alarms	Extreme short delay

With MTC, a diverse range of new services and applications can be offered. The potential MTC applications have very different features and requirements that add constraints on the network technology as well as on MTC devices. A comprehensive range of applications has been presented in [20] [21]. Table-2 provides some notable MTC service functions and application examples, including their major requirements on cellular systems.

2.1.2 RACH for M2M and H2H

M2M and H2H communication devices need to perform an initial access procedure called RA procedure for delivering resource request to the network. Both M2M and H2H are able to perform this RA procedure using Physical Random Access Channel (PRACH). Present procedures for RA in LTE standard system involves transmission of a limited number of preambles using slotted aloha [22] method to a base station without prior resource allocation. Because of performing conventional RA and signaling procedure in an environment containing huge number of devices, there will be resource shortage of RACH, which can introduce a high collision probability and massive access delay [23]. Therefore, the first priority improvement area is the overload control of RACH, which is the cellular uplink-signaling channel.

Table 2: M2M Communication Application and Service

Service functions	Application examples	Main requirements
Metering	Electric power, gas, and water metering	Support of a massive number of MTC devices with small data bursts and high coverage
Control systems and monitoring	Industrial and home automation, and real-time control	Low-latency data transmissions
Payment	Point of sale and vending machines	High level of security
Security and public safety	Surveillance systems, home security, and access control	High reliability, high security, and low latency

2.2 Random Access Procedure

2.2.1 Brief overview on LTE Network

LTE is a standard for wireless network with a high-speed data. It is an evolution from the standard GSM/HSPA established by the 3rd Generation Partnership Project (3GPP) and widely used for mobiles. The LTE brings more capacity in order to accommodate a vast number of future device and to provide high-speed data link with

a new radio interface. Some concepts of LTE will be explained in this section to be requirement for understanding the followings section.

3.2.2 LTE Channel and Frame Structure

A conventional LTE FDD frame structure is depicted in Fig. 5. In time domain, the duration of one radio frame is 10 ms where each frame is identified by a number known as System Frame Number (SFN).

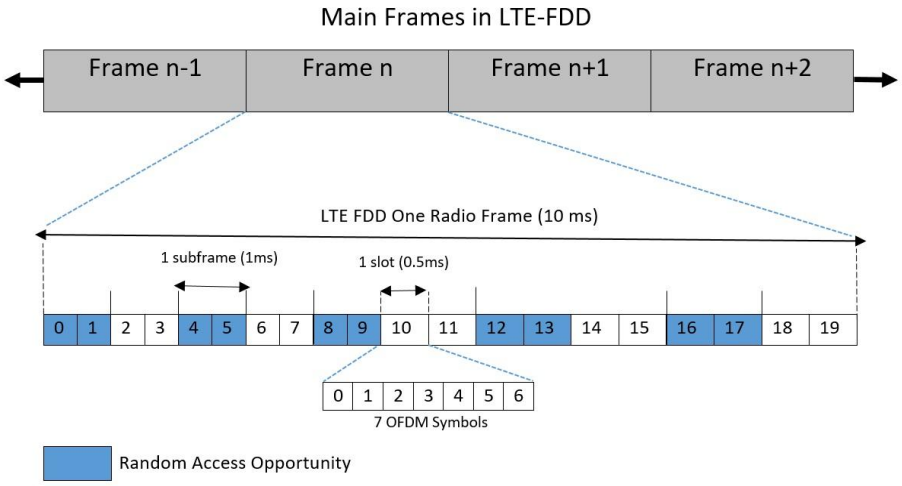

Figure 5: Type-1 LTE FDD Frame Structure

One frame is divided into 10 sub-frames of 1 ms each and each subframe is again divided into two equally sized slots of 0.5 ms, resulting a 20 slots per frame. Each slot contains either six or seven OFDM symbols, depending on the Cyclic Prefix (CP) length. Whereas symbol is the smallest modulation unit in LTE which is equal to one 15 kHz subcarrier in the frequency domain.

In LTE, three different types of channels are used to transport data across the LTE air interface. The channels are separated based on the types of information they carry and process. In LTE the channels are grouped into three channels,

i. Logical Channel
ii. Transport Channel and
iii. Physical Channel

The details function and the categories of the above channels will not be provided here as most of them are not relevant to this work, however the details can be found in [24]

[25] [26]. When UE does not have RRC connection in order to establishing the connection an uplink control channel of a logical channel called Common Control Channel (CCCH) is used for random access information.

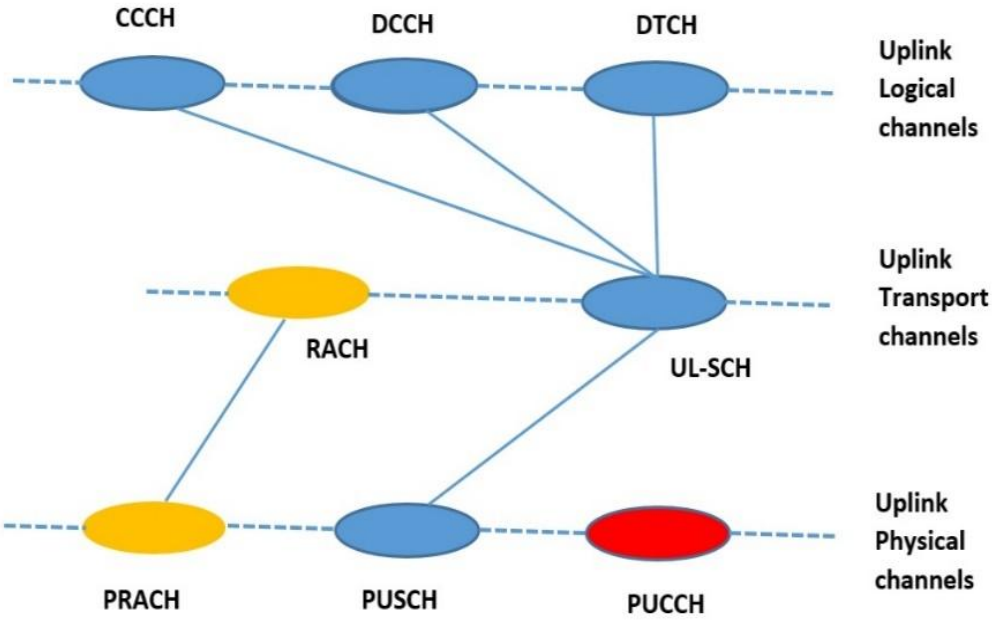

Figure 6: LTE Channel Mapping

Fig. 6 shows how CCCH is mapped to its uplink transport channel corresponds to RACH via Uplink Shared Channel (UL-SCH) and finally the RACH is directly mapped to PRACH to establish the connection request over air interface to the base station or eNodeB.

2.2.3 PRACH Configuration

In Fig. 5 it is shown that each System Frames (SF) contains Random Access attempts called RAO can only be transmitted in specific sub-frames, which are mentioned as Random Access slots (RA slots). One or more RA slots may be supported in each frame and the number of RA-slots depends on the PRACH configuration index, which also depends on the cell size. LTE defines up to 64 possible configurations [27] that varies between a minimum of 1 RA-slots in every 2 frames to a maximum of 1 RA slots per 1 subframe, i.e., every 1 ms. In Fig. 7 some PRACH configurations index are presented where colored squares represents RA slots [28].

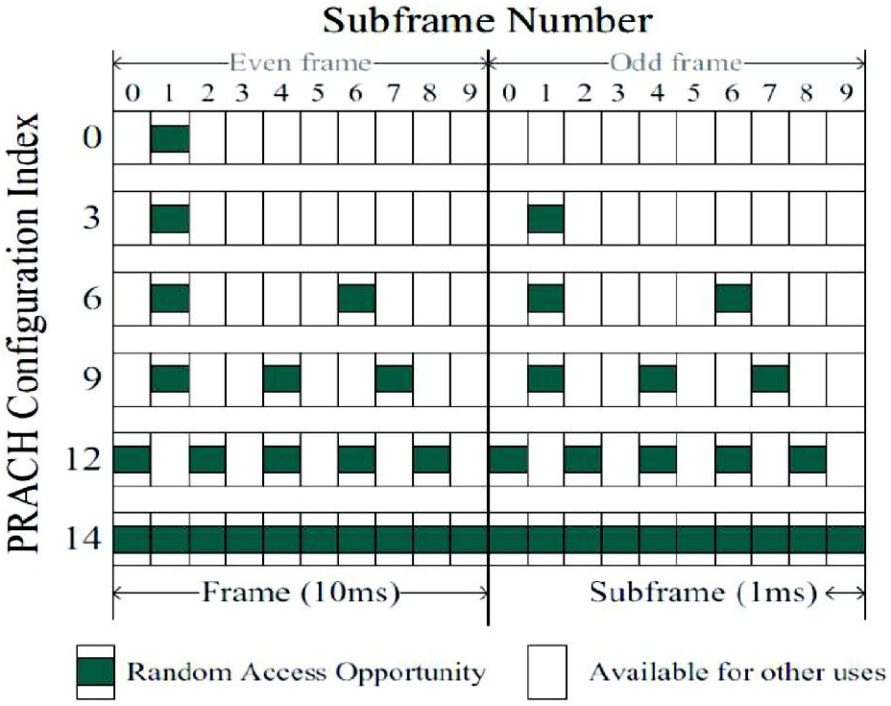

Figure 7: Representation of PRACH configuration index (directly reproduced from [28])

From Fig. 7 based on different PRACH configuration index it is clear that RACH request are restricted to RA-slots. Due to this nature of the PRACH configuration arrangement LTE adopts s-ALOHA protocol to control Random Access Procedure. Random Access Procedure is described in the next following section.

2.2.4 Random Access Mechanism

As mentioned earlier, the RACH is the initial access through which a user is connected with the network. A device (H2H and M2M) must initiate the access procedure to establish a connection to the BS/eNodeB/access point mainly in the following situations as mentioned in [28] [29]:

1. While UE is trying to establish an initial access to the network (RRC_IDLE to RRC_CONNECTED).
2. While uplink synchronization is lost and UE is trying to receive/transmit a new data.
3. To perform a seamless handover (change of associated eNodeB).
4. To re-connect to the network in case of radio link failure.

In order to handle all these situations, random-access procedure in LTE-based cellular system can be categorized into two different forms:

- Contention based (Support collision)
- Contention-free (No collision and applicable to handover only)

Due to the limitation in the number of available resources as compared to the massive number of access request to be supported, the contention-based scheme is the main focus of this deliverable. This section describe the random access scheme used by the conventional LTE network. Random access RA procedure of 3GPP LTE is briefly summarized in Fig. 8.

i. Random-access Preamble transmission:

In this step, UE sends its access request by transmitting one out of available preamble sequence via Msg-1. The preamble is selected in a random manner and carried in the PRACH, which is a part of an uplink resource of a LTE network. A Random Access Response (RAR) window is set up to wait for the RAR. If a UE does not receive the RAR in a RAR window, it means initial access has failed and UE shall randomly backoff for a period between 0 to Backoff Parameter value.

As shown in Fig-8, each random-access procedure consist of the following four steps:

ii. Random-access Response:

In this step, the eNB transmits the access response to the detected preamble sequence by sending an RA Response (RAR) via Msg-2 on the Physical Downlink Shared Channel (PDSCH).

iii. Scheduled Transmission:

After receiving the RAR at step-2, the UE transmit a connection request such as Radio Resource Control (RRC) connection request followed by Msg-3 in order to establish a connection using Physical Uplink Shared Channel (PUSCH).

iv. Contention Resolution:

In this step, when eNodeB receives Msg-3 it replies Msg-4 to confirm that the connection is successfully established and the status changes to RRC_CONNECTED. Otherwise, if the Msg-4 is not received by the UE the RA is declared as a failed and UE needs to restarts the RA process all over again until the allowed preamble retransmissions are reached. For further details on RACH procedure in LTE [30].

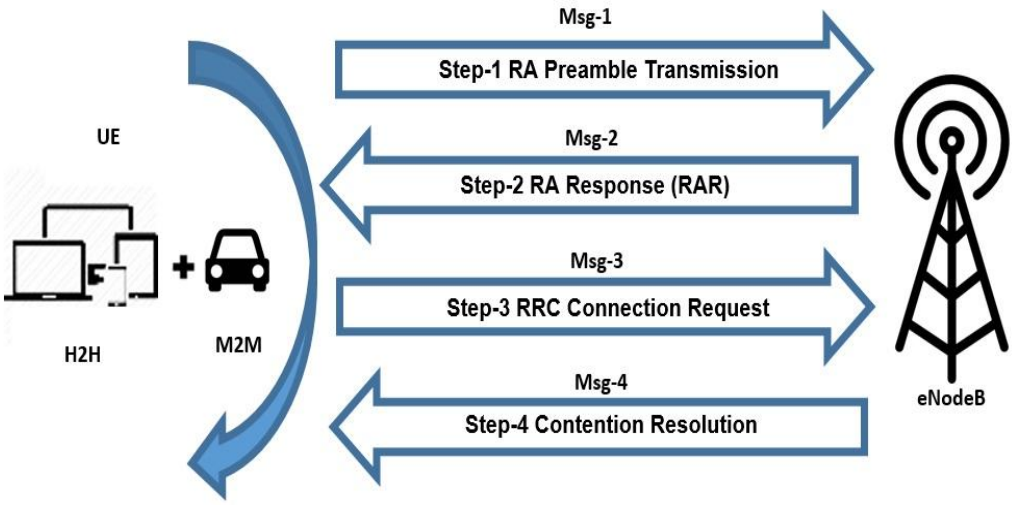

Figure 8: Random-Access Procedure

In each LTE cell there are up to 64 orthogonal preambles available these preambles are created by Zadoff-Chu sequence [31]. However, the eNodeB reserves some of them for contention-free access while the remaining ones are used for contention-based RA.

3.2.5 Conventional RACH Limitations and Overload Control Mechanism

As the number of M2M devices has been rapidly growing, the load on the random access channel is also increasing. Consequently, devices are attempting RA within a small time interval which is causing access problem referred to as the “massive access problem” as mentioned by 3GPP. The worst-case scenario according to 3GPP is that thousands of devices may attempt to perform RACH within a 10 ms time window [31]. However, at a peak traffic load where the number of access request is maximum standard LTE random access mechanism suffers congestion due to the high probability of collision and this cause excessive time delay [32] [33]. In [34], results showed that using standard current LTE medium access control system, the access delay may be intolerable when the number of devices exceeds 30000 per cell. As a result, for some delay critical M2M applications that requires ultra-low latency (e.g., e-

health, intelligent transportation system) the standard LTE will not be adequate, which may cause a sharp degradation of QoS.

One possible way to mitigate the overload problem is to increase the RA opportunities per frame, but this cause a fall in existing amount of resource for data transmission and therefore, it reduces the data transport capacity of the uplink channel. To diminish this problem besides improving the legacy system, it is significant to provide an effective approach for managing the massive access in the radio access network to reduce the network overload as well as to minimize the latency.

In this section, we discuss RA congestion solution proposals to control the RACH overload problem caused by M2M traffic in LTE system. The proposals are categorized under two classes, i.e. 3GPP and non-3GPP specified solutions. In [35], 3GPP has proposed six basic mechanism for RA overload control and in the following, we briefly describe the principle of these techniques.

3.2.5.1 3GPP Specified Solutions

Access Class Barring (ACB)

ACB is a renowned mechanism in controlling RA congestion by decreasing the access arrival rate. ACB can define 16 access classes [36], each class operates on two factors: a set of barring access classes (ACs) in which devices are classified, and a barring time duration (Tb) [35]. First the eNB broadcasts the ACB parameter, p (ranging from 0 to 1) to the MTC device and each MTC device also generate a random number, r (between 0 and 1) uniformly. If r<p, the device is permitted to transmit their RA preamble otherwise the access is barred and the device has to wait for a random backoff time based on the barring time duration (Tb). However, in peak congestion condition when massive number of devices try to connect in a very short time the value of p might be set to a very low value which causes chaotically high time delay.

In [37] a Dynamic ACB scheme is approached and a Prioritized RA jointly with dynamic ACB is proposed in [38] to improve the performance of RACH channel.

MTC-Specific Backoff

In this scheme, when a device faces a collision, it waits for a fixed backoff time and then retransmits a connection request. Although the network performance under low congestion levels increases using this scheme, however, in high-congestion level network performance is reduced [39]. In [40] it is suggested a separate backoff scheme for separate user group like delay sensitive M2M with H2H as group 1, on the other hand delay-insensitive M2M as group 2. Devices in-group 2 has a longer backoff time compare to group 1. Although this scheme can provide some enhancement for low congestion [41], it is not satisfactory to handle peak congestion level.

Dynamic Resource allocation

In this scheme, the BS allocates additional RACH resources dynamically in the time domain or frequency domain or both by predicting the congestion level of the access network overload caused by MTC devices [42] [43]. In [41], a simulation result is presented by 3GPP showed that additional allocation can solve most of the congestion problem. However, allocating more resource for RACH will reduce the available resources for data traffic, which in turn causes problem in the network performance.

Slotted Random access

In slotted aloha method, each MTC device is provided with a dedicated RA opportunity using only the slot allocated to the device [43]. In effect, the access delay becomes very high in ultra-dense scenarios, as the RA cycle for each device will be excessively large.

Separate Resources Allocation

In this approach, the MTC and HTC devices are delivered with different RACHs as an attempt to avoid the effect of RA congestion on HTC devices. The separation can be possible by assigning different RA slots for MTC and HTC devices or by splitting the

available preambles into MTC and HTC subsets [42] [43]. This separation technique might help dropping the negative impact on non-M2M devices.

Pull based RA

All the schemes mentioned earlier section are categorised as push-based approach in which RA attempts are random and started by device autonomously. On the other hand, in pull-based method [35], the RA procedure is started by eNB. Therefore, the eNB can control the number of requests and mitigate congestion problem. The devices perform RA attempts only after getting paging messages from the eNB. However, the scheme needs of additional control channel resource to page massive device. To reduce the number of paging load in this approach, a number of MTC devices can be paged together by following a group paging method [43]. In [44] an analytical model is developed for performance evaluation of group paging in LTE.

3.2.5.2 Non-3GPP RACH solutions for supporting M2M services

In addition to the solutions specified by 3GPP, several academic, industrial and governmental institutions have also proposed various RA congestion solutions to support the huge MTC in LTE networks. Some of the proposals offer better performance than 3GPP solutions. Vital proposals are discussed in the subsequent paragraphs.

Group-based RA Scheme

Group-based RA mechanism is an addition of pull-based group RA model. Based on some specific criterion like similar QoS/delay requirement MTC devices can be grouped in a particular region and RA procedure can be assigned on group-basis in order to minimize the network congestion. In [45], a two-layer device segregating technique to reduce congestion is proposed where in first layer devices are grouped into several paging group. Device within a paging group are then partitioned into different access groups. For each group, a group head is assigned who is responsible for communicating with the eNB. Another group based approach is proposed in [28]

where dividing cell coverage area into different spatial groups is mentioned. The idea behind the cell division approach is to permit the use of same preambles at the same RA slot by different groups of MTC user if the distance is not larger than the multi path delay spread. In [46] a cluster-based approach is proposed for mitigating the inefficiencies of the ACB algorithm. In [47] a technique is proposed based on groups in which M2M devices are grouped according to their characteristics (access speed) and requirements (maximum tolerable delay). Hence, a decision making step is taken upon reception on the data about the characteristics after the third step of RACH procedure in LTE.

Code-Expanded RA Scheme

In this scheme, a codeword (set of preambles) is transmitted to execute RA process instead of a single preamble. A virtual RA frame is considered which contains a group of RA slots, or a set of preambles in each slot. This allows expanding the number of contention resources, and reduction the collision [48].

Self-optimization overload control RA

This self-optimizing mechanism is proposed in [49] which configures the RA resources depending on the load condition. It encompasses a combination of other solutions specifically Separation of RACH Resources, ACB Schemes, and Slotted-Access scheme. The LTE-A ACB scheme is modified by adding two classes of M2M devices, i.e. high priority and low priority. The ACB scheme is applied to the next attempt when a device is not granted access in the first attempt and. After receiving a RAR, a device sends the number of retransmitted preambles to the eNodeB within message 3, which is used for overload monitoring and adjusting the RA resources according to the congestion level of the RACH. If the RA slot usage reaches the maximum accessible limit, the lowest priority M2M class devices are temporarily restricted from accessing the network until overload conditions recover.

Prioritized Random Access

In this scheme, applications are divided into five categories: HTC, high-priority MTC, low-priority MTC, scheduled MTC and emergency service [38]. Also, virtual separation of the RACH channel is applied into three classes i.e. HTC, random MTC, and scheduled MTC & emergency services [38]. A prioritization is accomplished by applying distinct backoff window sizes to guarantee QoS using a prioritized access algorithm based on the mentioned classes. It has been reported that this type of scheme is better than other EAB methods in terms of access delay and probability of success; but it still needs prohibition a M2M device for an amount of time [50].

In the study above, we have discussed existing methods for controlling the RA procedure in corporation with M2M from 3GPP and non-3GPP perspective. We have highlighted some key issues related to the enormous number of devices trying to connect to the network at the same time. We also noticed that some existing RA congestion control methods are not appropriate for solving the RACH overload problem in massive M2M scenario. ACB based methods are targeted to lessen collisions, preamble-splitting methods perform decently to protect the H2H QoS but both schemes result in intolerable delay. Resource allocation methods are suitable for both counts; however, like back-off based schemes, it decreases the general throughput. In contrast to the ACB and Backoff schemes [51], slotted access approach is advantageous in several ways such as better access rate, complexity, and has minimum influence on H2H traffic. The simulation results presented in [51], demonstrate that slotted access scheme out-performs others in case of access rate even for randomly assigned slot. The slotted access uses the already provided paging mechanism whereas ACB imposes signalling overloads adding more complexity. [52] shows a comparison of ACB, Backoff and slotted access approach for overcoming the LTE RACH overload problem, summarizing the minimal effect of slotted access method on H2H access traffic in contrast to the other methods. Nevertheless, some works show that the combination of two or more methods could result in better performance aimed for overload controlling. In our work we are interested in slotted access based RA and overview of slotted access based RA is given in following subsections.

2.3 RACH in delay Perspective

3.3.1 SA-RACH scheme with retransmission and RACH instability

In RACH access scheme transmissions are restricted to slot in order to avoid overlap of user, traditional slotted aloha scheme is effectively used in all standard cellular RACH [53]. In our work, RACH using slotted-aloha protocol is called SA-RACH. Collision is unescapable due to the nature of this protocol and it cause poor throughput performance at high traffic loads. In order to maximize the chance of the request getting through, cellular system allows system to retransmits the request again after a collision. Retransmission help to reduce the possibility of blocking the user by giving more chances. However, it is essential to control the retransmission to make a system stable because at higher load higher retransmission will generate more retransmitted traffic so the aggregated traffic will be high and the system will be unstable. A retransmission cut-off strategy with fixed a back-off window is presented in [54] [55].

As shown in Fig. 9 a user-generate a RACH request and need to wait for the next time slot and will send the request if that particular slot is used by another user at the same time collision occur. After the collision, a user checks its retransmission value and if it is minimum than max retransmission number then it will generate a random backoff window and retransmit again. If a user exceeds the maximum retransmission number the request will drop and needs to start from the beginning.

For slotted aloha RACH, we consider some assumption this assumption represents a scenario that allows us to analyse RACH throughput of conventional slotted aloha according to the standards. These assumptions are:

- All packets have the same length equal to the length of the slot.
- There is a huge number of user generating request in a minor interval.
- RACH request arrival should follow Poisson arrival process.
- The system is perfectly synchronised with every user and can transmit only at the beginning of a slot.

2.3.2 RACH request process diagram

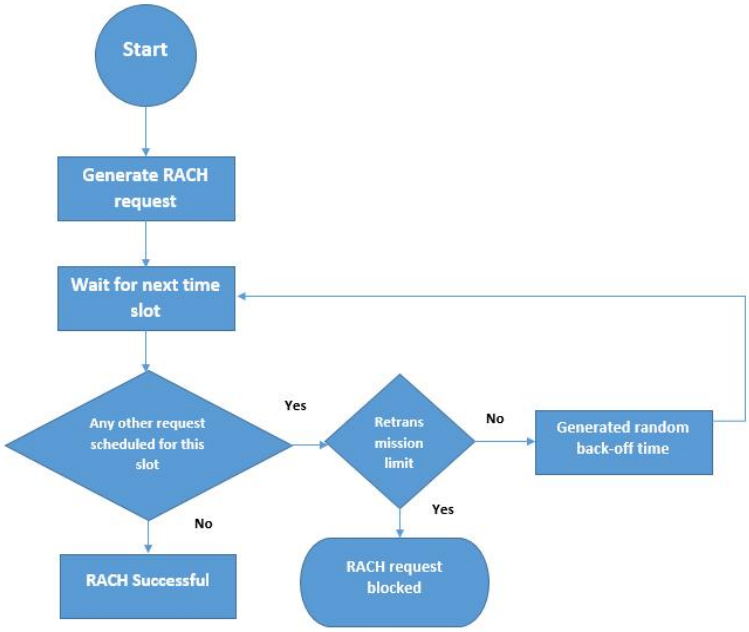

Figure 9: RACH Request Process of a Cellular System

- All users share a single RACH mean only one preamble available.

2.3.3 SA-RACH Delay Study

To evaluate the performance of the Slotted Aloha RACH (SA-RACH), we consider OPNET based simulation scenario, we have used the standard parameter as shown in Table 3. For result validation, we compare our result with [55] and the result presented in this deliverable shows almost the same result as in [55].

Table 3: Simulation parameters for SA-RACH

Parameter	Value
Transmission data rate	271 kbps
Data Packet length	157 bits
Slot period	5.7657e-004 s

Frame duration	0.0046
Retransmission limit	7
Max Backoff window value	14
PRACH configuration index	{8,10,12,14}
Number of preamble sequence	1

Fig. 10 shows the RACH throughput performance at different retransmission value [1, 2, 4 and 7] with a maximum backoff interval of 14 slots within which a user selects a slot at random. In another effort to validate our model we compare this result with those presented in [54] [55]. The results show that at low generated traffic the highest maximum number of retransmission produce better throughput.

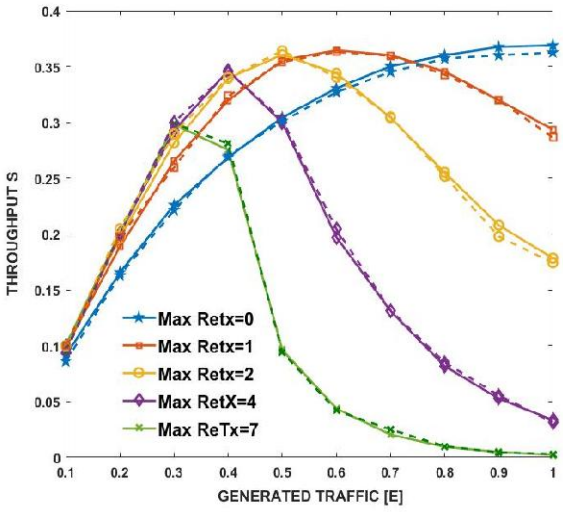

Figure 11: RACH throughput against generated traffic for different retransmission number with a random retransmission interval of max backoff value fixed at 14 RACH slots (Solid line represents our result and dashed line represent the result in [55])

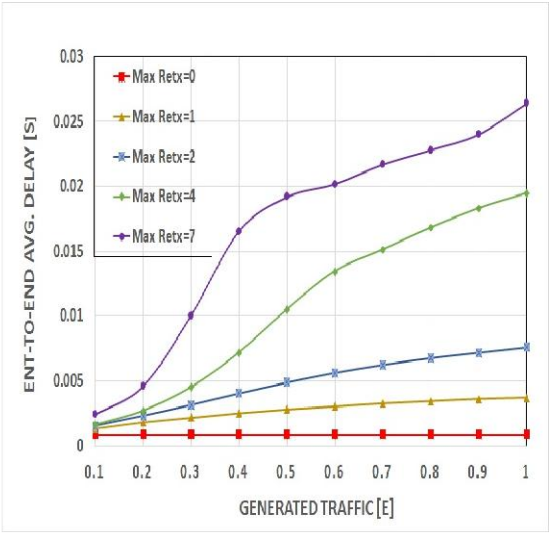

Figure 10: Average end-to-end against generated traffic for different retransmission number with a random retransmission interval of max backoff value fixed at 14 RACH slots.

Although with the growth of the generated traffic the throughput increases up, after some point it starts to fall. For example, for a maximum retransmission value of 4, it

shows an increasing trend in throughput up to a value of 0.4E. Nevertheless, at higher generated traffic it starts to fall as the retransmission traffic injects more traffic to the system causing the traffic to exceed s-ALOHA capacity.

Average end-to-end delay is another parameter used in our work to describe the behaviour of RACH access at different number of transmissions. In general, Fig. 11 showed that average end-to-end delay is increasing in nature with respect to generated traffic growth. Fig. 11 showed that when there is no retransmission (max retx=0) the delay is almost the same throughout the range of generated traffic but the delay increases with the increase in maximum number of retransmission, where at maximum retransmission value 7 the delay is also maximum. Therefore, we can see that there is a trade-off between RACH throughput and average end-to-end delay with regards to maximum number of retransmission.

The impact of RACH Configuration Index (CI) on the average access delay is illustrated in Fig. 12 In particular, as shown in Fig. 12 the average access delay increases with increasing traffic load. Besides this, it also shows that at higher configuration index, corresponding to higher number of RA Opportunities (RAO) per frame; the access delay reduces compared to lower number of configuration index.

The retransmission interval width is the second parameter used to determine the retransmission behaviour in our work. A high interval reduce the probability of more than one user retransmitting at the same time slot that could cause another collision. As shown in Fig. 13 the retransmission interval has an impact on average end-to-end delay. In general, for all interval values delay increases with the increase of generated traffic level. For example, retransmission interval width value 3 (3 RACH slot) shows an increasing trend in delay and becomes stable at higher load with a value of about 0.01s. On the other hand, retransmission interval width value 50 (50 RACH slot) shows a different characteristic, it shows that the delay increases with respect to generated traffic growth and becomes large compared to interval width 3. However, it is clear from the above Fig. 13 that at low generated traffic level, the delay introduced for the high retransmission interval is accepted for some delay critical applications but at higher traffic load, the delay is intolerable.

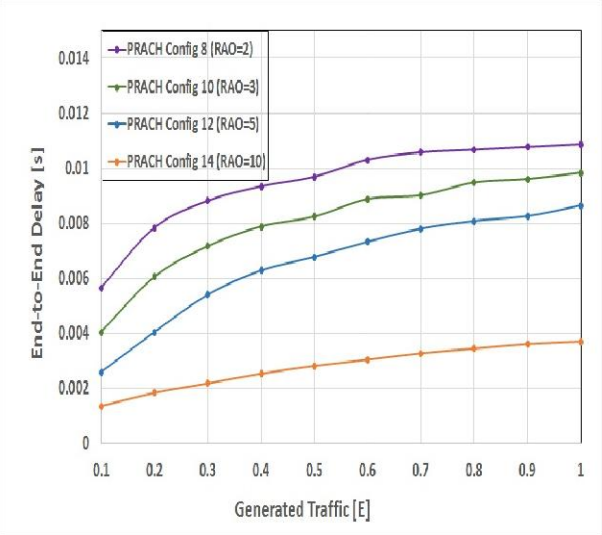

Figure 13: Average end-to-end against generated traffic for different PRACH configuration values with a random retransmission interval of max backoff value fixed at 14 RACH slots

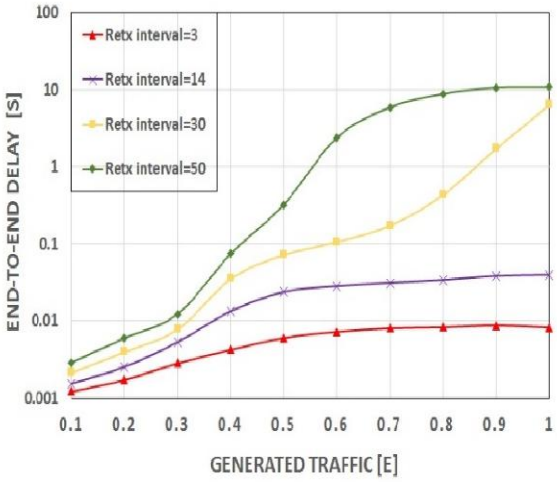

Figure 12: 9 Average end-to-end delay vs generated traffic for different retransmission interval values with a fixed retransmission value

As can be seen from Fig 5, the throughput performance of s-ALOHA is inadequate due to the limited channel capacity, which aggregates with the retransmission of the contended traffic. Consequently, the traffic surpasses the channel capacity causing the system to become unstable. Although the s-ALOHA for RACH access proves to be unstable as stated above, it is efficient enough in terms of H2H communication in current cellular networks. The reason lies in the dimensioning of the system and regularity of the H2H traffic as the RACH request falls within the s-ALOHA throughput capacity. On the contrary, if M2M traffic is allowed to be supported by the system provisioned with s-ALOHA protocol, the M2M traffic will cause extra load on the system, as M2M systems will have huge number of devices. Therefore, the traffic has the potential capability to cause the RACH overload affecting s-ALOHA to be unproductive for practical use. To conclude, in order to support M2M communication in cellular network effectively, an agreeable RACH congestion control mechanism development is necessary.

2.3.4 Dynamic RACH-Configuration Scheme for Delay sensitive M2M applications

As we investigate the RACH delay profile in previous sections, it is clear that RACH becomes unstable in high load scenario and cause an excessive delay, which is intolerable for some delay-sensitive application. In this section, we proposed a dynamic RACH configuration technique for a specific set of users and for this; we divided the user group into two

i. Delay tolerance group (H2H)
ii. Delay sensitive group (M2M)

The motivation behind the approach is to mitigate the delay issue for M2M delay-sensitive applications. We proposed a dynamic RACH configuration model rather than a fixed configuration. We use the same set of parameters for a dual user group in low load scenario because both groups perform well on the contrary, in the high low scenario we change the parameter set in a dynamic manner for the M2M user group in order to reduce the delay over priority. The method will adjust the PRACH configure index, backoff Interval value and max Retx value according to the load to satisfy the delay profile for user group ii. Two different sets of parameters shown in Table 4. For the low-priority UEs, that are tolerant to delay, so there is a relatively large BI value with higher max Retx number. On the contrary, the BI value of low-priority UEs is small and using minimum max Retx value.

Table 4: Simulation parameters for Dual user group

Parameter	H2H Value	M2M Value
RAO	5 per frame	5 per frame
Slot period	1 ms	1 ms
Frame duration	10 ms	10 ms
Retransmission limit	7	7 & 3
Max Backoff window value	20	20 & 5
PRACH configuration index	12	12 & 13
Number of preamble sequence	1	1

The simulation is performed in the LTE scenario of a single cell, with an eNB and many UE devices. Where the ratio of H2H and M2M is 10:1. The basic parameters setup for random access procedure are defined in [55] and some important simulation parameters are set as shown in Table 4.

Fig.10 and Fig. 11 show the avg. end to end delay for the different user group, when the load is less than 0.4 E it adopt same parameters for both user group and when the load is higher than 0.4E, it dynamically change it's set of parameter for the only M2M user group. In higher load M2M users are using PRACH config 13 [1, 3, 5, 7, 9] whereas H2H users are using config index 12 [0, 2, 4, 6, 8]. The result shows that the delay for M2M user group become stable because they are avoiding collision with H2H and have more access in slots with different sets of BI and Retx value. We also investigate the scenario for two limit of BI and Retx value, in Fig.15, both user groups are using max BI and Retx value on the other hand Fig.14 both user groups are using min BI and Retx value. In both scenario M2M user group maintain a minimum end-to-end delay, which is, much less than other group H2H.

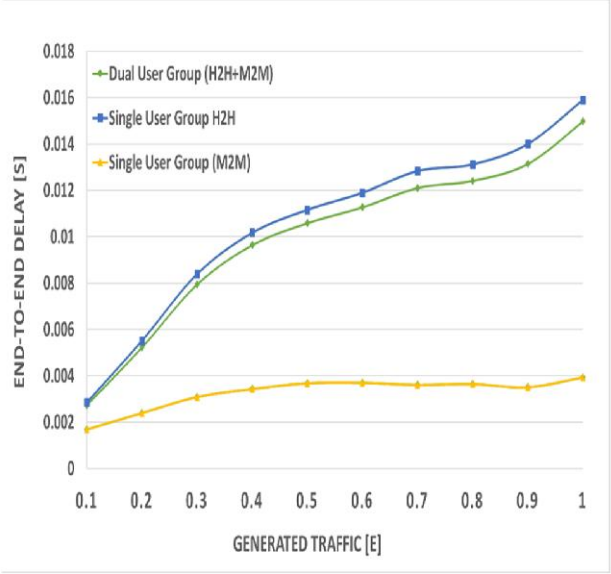

Figure 15: Average end-to-end against generated traffic for different user group scenario for maximum value of Retx and max backoff 20 RACH slots

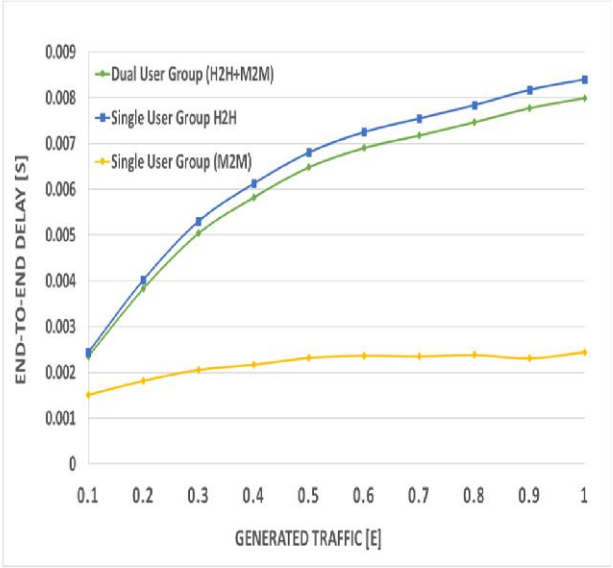

Figure 14: Average end-to-end against generated traffic for different user group scenario for minimum value of Retx and min backoff 5 RACH slots

In the above study, we present a dynamic RACH configuration approach for LTE that reduces the impact over the H2H devices and establishes priorities among M2M devices by dynamically setting the backoff interval, RACH configuration index and max Retx value for the different classes of device. Based on the simulations results, we observe that this approach reduces the access delay of M2M devices with high priority without affecting in the total number of accesses. Thus, our approach is able to handle IoT applications that present different access delay requirement.

2.4 Q- Learning Assisted Random Access

2.4.1 Learning based techniques

The learning techniques mainly aim to acquire the system variations/parameters uncertainties, to classify the associated cases/issues, to predict the future results, and to explore possible solutions [56]. Machine learning is characteristically categorized into three broad classes depending on the nature of the learning objects and signals [57] [58]: supervised learning, unsupervised learning and reinforcement learning. In supervised learning, example inputs and their anticipated outputs are provided to the learning agent that targets to determine a general rule mapping inputs to outputs. On the other hand, for unsupervised learning, instead of feeding prior input data to the learning agent, it turns to its own ability to find the embedded structure or pattern in its input making it suitable for application in the AI category of cellular networks. Lastly, in reinforcement learning, the agent interacts with a dynamic environment in order to obtain its goal.

2.4.2 Advantage of learning in wireless communication

As the cellular system moves on from one generation to the next, the number of reconfigurable system parameters also increase considerably. For instance, the number of configurable parameters are about 200 in a 2G node, which increases to about 1000 in a 3G node, further rising to about 1500 in a 4G node. The prediction for the number of system parameters in a 5G node is around 2000 [59] [60] and this number is expected to soar up with each upgrade of the cellular system. Therefore, carrying out self-configuration, self-optimization and self-healing operations will be tremendously challenging. In addition, the evolving ultra-dense networks will need to

observe environmental variations, learn uncertainties, plan response actions and configure the network parameters effectively to handle these operations . Possible paybacks in effective handling of these operations can be achieved using emerging ML techniques.

2.4.3 Learning technique for RACH congestion

Similar to other application scenarios, learning techniques are expected to provide substantial paybacks in the case of solving RACH overload problem by adaptively learning various parameters. Some of the cases for RACH include [61]:

I. Learning to exploit a unique RA slot for each MTC device within the considered transmission frame in a way that concurrent transmissions in the same RACH opportunity can be avoided [62].
II. Learning to adapt an access control parameter, i.e., access barring factor for the RACH congestion [37].
III. Learning to associate MTC devices with suitable BSs/eNodeBs with the objective minimizing overall access network congestion [54] [25].

2.4.4 Q learning in Slotted-Aloha RACH

Q-Learning is a basic model of reinforcement learning with a simple algorithm that allows early system convergence [37]. In general, it is a trial-and-error technique, which decides its action through learning a system behavior of a given environment. As we mentioned in the earlier section, standard RACH has become unstable in presence of massive M2M traffic. In this regards, [43] proposed a Q-learning based solution to increase the overall throughput of RACH and guard H2H traffic against the performance degradation that can be triggered by massive M2M request. According to the authors, the traffic should be divided into two groups one called H2H and another M2M (containing MTDs) and apply learning technique for M2M group whereas H2H group will use the conventional s-ALOHA technique without learning. M2M communication uses a virtual frame of RA slot called M2M frame and the length of the frame (number of slots) is equal to the M2M user. Every slot in the frame has a Q-value that keeps the transmission history. Every M2M user has individual Q-values and at

initial, all Q-values is set to zero and updated after every RACH attempts using the following model:

$$Q \leftarrow (1-\alpha)Q+\alpha r \qquad \text{(i)}$$

Where α is the learning rate and r is the reward (+1) or punishment (-1) depending on the status of the request.

In transmission time, each user will transmit in the slot with the highest Q value. At convergence, M2M RACH becomes contention free because every M2M user will select a dedicated slot since the number of overall collisions will reduce and there will be no collision among M2M traffic. Hence the performance evaluation in single user group scenario (where only M2M users using Q-learning) offers up to 100% throughput. On the other hand in dual user group scenario in case of the high load from H2H, Q-learning stabilized the total throughput at 35% (approximately the max efficiency of slotted aloha). In low load case for H2H, proposed solution shows a significant enhancement, raising the total RACH throughput to 55% without negatively affecting the delay performance.

2.4.5 Group-based separate Q-Learning for delay sensitive M2M applications

QL technique in RACH has been described previously, here group based separate Q-learning technique is applied for M2M users group called delay sensitive group. Therefore we implement the slot learning here by considering the existence of H2H using the conventional SA-RACH scheme. The intuition is that combining QL-RACH with SA-RACH in the RACH access reduces the overall number of collision since there will be no collision between the M2M users after convergence. Similar to the description of the single user QL-RACH scheme, the M2M users also learn their individual dedicated slots in a virtual frame called the M2M-Frame of size equal to the number of M2M users. The slot timing and length is mapped directly on to the control frame for the PRACH frame for the LTE standard. The information on frame timing and the number of active M2M users are broadcasted by the central entity eNB via downlink channel. The M2M-frame keeps repeating and is only considered by M2M users in

which user transmissions are restricted to only one per M2M-frame. Similarly, the transmission history of each M2M user is recorded using the Q-value in each slot of the M2M-frame and is updated at every successful or failed transmission attempt using equation (i). Also the transmission decision is made (by an M2M user) based on the slot with the highest Q-value. At convergence, M2M RACH access becomes contention free amongst the M2M users, with H2H using SA-RACH and not being aware of the M2M-frame. Fig.16 shows the combined QI and SA- scheme. Where the global frame is repeating with Q-learning for M2M and SA-RACH is for H2H.

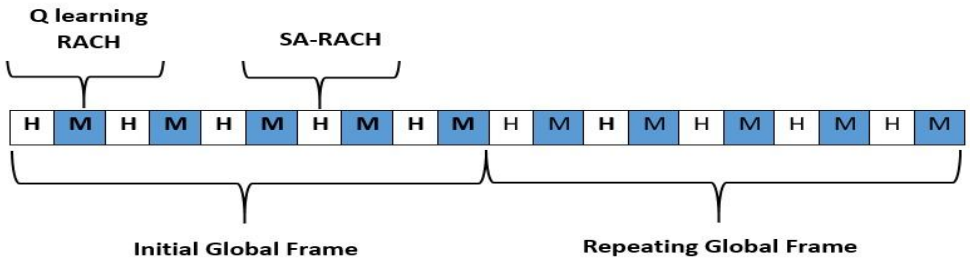

Figure 16: Representation of combined SA-RACH and QL-RACH scheme

The simulation is performed in the LTE scenario of a single cell, with an eNB and many UE devices. Where the ratio of H2H and M2M is 10:1. The basic parameters setup for random access procedure are defined in [43] and some important simulation parameters are set as shown in Table 5.

Table 5: Simulation parameters for Q-learning

Parameter	Value
Slot period	1 ms
Frame duration	10 ms
Retransmission limit	5
Max Backoff window value	14
PRACH configuration index	13
Number of preamble sequence	1
Learning rate	0.01

RAO	5 per frame
Preamble format period	1 ms

Fig.17 compares the single user group RACH-throughput performance of the SA-RACH with retransmission scheme and the steady state RACH-throughput of the QL-RACH scheme. It can be seen that the SA-RACH throughput increases with the increase in the generated traffic. However, immediately after the channel throughput limit (~36% which is the maximum capacity of the s-ALOHA) is reached, the aggregated traffic increases to the point that the s-ALOHA scheme can no longer support the traffic. This is why we can notice the throughput dropping with an increase in the traffic, to the extent that the channel becomes unstable. On the other hand, with steady state of the Q-learning, the QL-RACH scheme offers up to 100% throughput. This is because there are no collisions since the scheme is contention free.

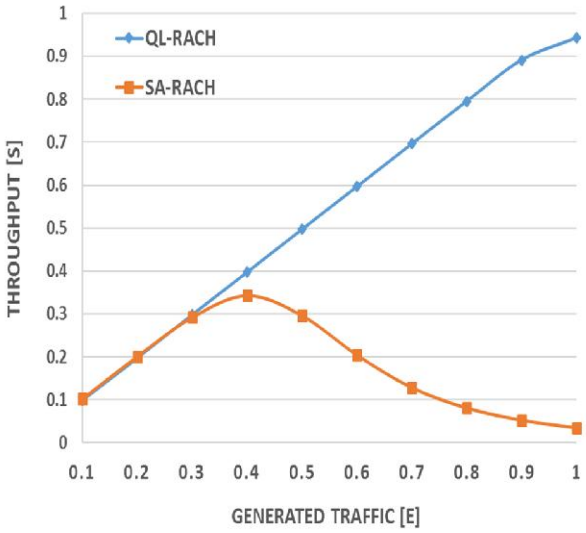

Figure 17: RACH Throughput against generated traffic for single user group with and without Q-learning

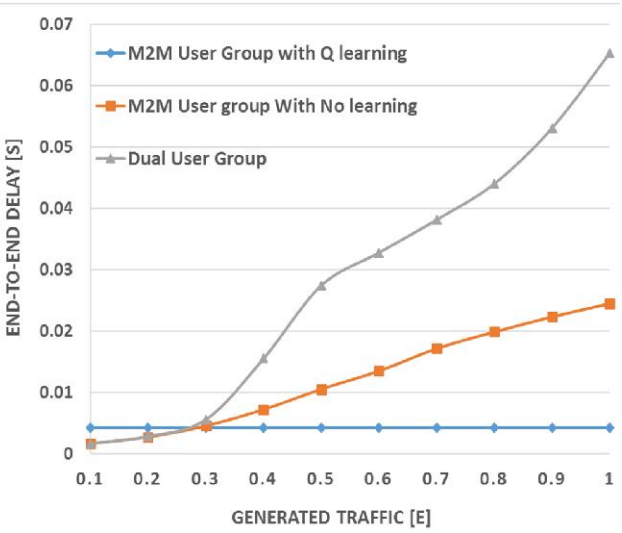

Figure 18: Average end-to-end against generated traffic for different user group with and without Q-learning

Fig. 18 shows the delay performance for dual user group where in dual mode SA-RACH (both H2H and M2M) and in single user mode M2m user use QL based RACH on the other hand H2H user use conventional SA-RACH. It can be seen that using QL technique shows a minimum delay profile respect to non-learning and dual user group

scenario. In convergence mode there will be no collision between M2M user and the scheme is become contention free so the delay is less compare to non-learning mode.

2.5 Conclusion and Future Work

The upcoming cellular networks require to be designed in such a way that can provision the massive number of MTC devices fulfilling their various QoS requirements along with enriching the access latency, scalability and network throughput. As per the prediction of 3GPP and other organization members, simultaneous access attempt of the massive number of MTC devices will result in congestion of the RACH channel of the LTE. As an attempt for advancement of the design of random access operation to suite the M2M operations, extensive research has been and are still being carried out by researchers throughout the world. In this deliverable, we explore the situation inflicted upon the cellular network with the inclusion of M2M communication. Furthermore, a brief review of LTE networks has been provided namely frame structure, uplink channels, and random access request mechanisms. Afterwards, we inspect M2M traffic characteristics including the problems inflicted upon massively accessing the cellular network. In addition, we investigate different proposed solution mechanisms with their advantages and disadvantages, and provided comparisons among these schemes. We analyze slotted aloha scheme for RACH access thoroughly and deliver simulation results presenting the effect of various parameters of SA-RACH technique.

We approach a Dynamic RACH-Configuration Scheme for Delay sensitive M2M applications and based on the simulations results, we observe that this approach reduces the access delay of M2M devices with high priority. Thus, our approach is able to handle IoT applications that present different access delay requirement. Additionally, we explore the benefits of machine learning as an effort to minimize the RACH overload/congestion of the cellular network. We also approach a Group-based separate Q-Learning technique for delay sensitive M2M applications and simulation results shows a positive impact on delay. Using Q-Learning in a M2M scenario enhanced the throughput as well as delay for M2M devices. As a part of our future work, we will use a learning-based preamble separation model among H2H and M2M

for prioritization based access. Finally, we will introduce MEC with Q learning RACH protocol to get the end-to-end performance.

3. Traffic Management Mechanism for Edge-Assisted RAN Architectures under Flash Crowd

3.1 Introduction

The vision of future 5G systems is to enable service delivery in ultra-dense networks. Particularly, always-connected devices, such as various types of smart phones, tablets, video-game consoles, Virtual/Augmented Reality (V/AR) devices and wearable electronics impose significant pressure on the backhaul and access networks. Moreover, the emerging IoT and massive mMTC are expected to introduce a huge number of machine connections [63]. In this context, serious performances degradation in terms of QoS and/or Quality of Experience (QoE) is inevitable especially for the services with strict QoS requirements. Nevertheless, in such challenging environments, traffic bottlenecks in the core and backhaul networks can be reduced by locally processing data intensive task at network edge in proximity to user devices.

Mobile Cloud Computing (MCC) was introduced to deal with challenges of diverse and complex mobile service and application in terms of processing and data storage constraints in addition to battery lifetime, memory limitation and computational power of end-devices [64]. MCC augmenting the resource capabilities of mobile devices by acting as an Infrastructure as a Service (IaaS) for data storage and processing. However, the MCC also imposes huge additional load both on radio and backhaul of mobile networks and introduces high latency since data is sent to powerful server that is far away from the users [65].

To address the problem of long latency, the cloud services should be moved to a close proximity of the end users, i.e., to the edge of mobile network as considered in newly emerged edge computing paradigm. The edge computing offers significantly lower latencies and jitter, mainly because the computing and storage resources are in proximity of the mobile users. Moreover, edge computing could exploit the contextual information for provisioning the network congestion states. This could indeed be achieved by combining MEC based application platform with the communication and context services that could be provided by potential 5G technologies [66].

The ETSI standard on MEC [67] plays an important role in this direction. MEC, as a key 5G network enabling technique, allows leveraging the cloud computing power by deploying application services at the edge of the mobile network. This can facilitate content dissemination within the access network. A key component for enabling MEC are servers integrated within the operator's RAN (e.g., 3GPP, Wi-Fi or small cells). MEC opens the door for authorized third parties, such as Content Providers (CP), to develop their own applications hosted in the MEC servers. These applications can add the flexibility to handle the traffic from/to mobile users. Besides, operators can expose their RAN edge Application Programming Interface (API) to authorized third parties to provide them with radio network information in a real-time manner.

The MEC framework consists of a hosting infrastructure and an application platform. The hosting infrastructure includes the MEC virtualization layer and the hardware components such as the computation, memory, and networking resources. The MEC application platform includes an IaaS controller together with the MEC virtualization manager, and provides multiple MEC application platform services. The MEC virtualization manager supports a hosting environment by providing IaaS facilities, while the IaaS controller provides a security and resource sandbox for both the applications and MEC platform. Four main categories of services are offered by MEC application platform including Traffic Offloading Function (TOF), RNIS, communication services and service registry.

In addition to MEC, SDN paradigm, which is an emerging enabling technology, is utilized to facilitate data plane redirection mechanism through applying intelligence and centralize control over heterogeneous infrastructure [68]. Since the SDN controller has an overall view of the network, it has the visibility over data redirection. Furthermore, there are two important flow management protocols, known as OpenFlow [69] and Simple network Management Protocol (SNMP). Openflow is used for datapath control while SNMP is in charge of device control [70]. In OpenFlow Wireless [71], [62], SDN can control network by adding protocol to BSs/APs software.

Additionally, Delay Tolerant (DT) traffic which accounts for a large portion of mobile data traffic is considered in this study. DT traffics are featured with relatively long latency in comparison with delay-sensitive traffics. For instance, e-mails, updates of

social networking portals and firmware updates which can tolerate delay ranging from few seconds up to few minutes [62]. However, DT traffic has its delay requirements or lifetime, which are much longer than delay sensitive traffic.

Significant research efforts have been invested on reducing the current overload of cellular networks. The most importantly, research works have analyzed the impact of traffic offloading and caching technique. Authors in [72] propose an offloading mechanism in which the content can be delivered through small cells or D2D communications. References [73], [74] and [75] investigate the performance of two type of WiFi offloading. The first one is on-the-spot offloading that is when there is WiFi available, all traffic is sent over the WiFi network; otherwise, all traffic is sent over the cellular interface. The second one is known as “delayed” offloading where the traffic is delayed until WiFi connectivity becomes available. The work presented in [76] considers SDN-based WiFi data offloading, in which SDN controller facilitate the coordination between cellular and WiFi networks. In [77] and [78], the role of proactive caching via small cells and D2D are investigated for 5G system. In particular, [78] study the social networking and D2D use cases in order to exploit proactive caching. There has been some research in offloading computation to MEC or MCC [79], [80],. Considering the fact that MCC imposes huge additional load both on the RAN and backhaul and introduces high latency since data is sent to remote server. Therefore, MEC is seen as a promising approach to address the aforementioned problems. Moreover, MEC can provides an IT service environment and cloud-computing capabilities at the edge of the mobile network in close proximity to the users. While all the aforementioned studies present very attractive solutions, there are still limitations.

In this deliverable, we propose a congestion control mechanism in the context of MEC to reduce RAN congestion. The key idea is to delay DT content from being delivered, until the congestions expire. This mechanism driven by the following context information: i) the characteristic of data traffic (i.e., delay-tolerant data traffics) and ii) the network conditions (i.e., sudden traffic peaks). More precisely, the proposed mechanism functions within the framework of MEC. It is aims at real time decision making for selectively buffering traffic, while taking account of the network condition and QoS. In order to support a MEC-assisted scheme, the MEC server is expected to

locally cache delay-tolerant data traffics during the congestions. This enables the network to have better control over the radio resource provisioning of higher priority data. To achieve this, we introduce a dedicated function known as CCE, which captures the RAN condition through the RNIS function, and uses this knowledge to make real time decision to selectively and intelligently offload traffics. Analytical evaluation results of our proposed mechanism confirms that it can alleviate network congestion more efficiently.

3.2 Overview of Content Caching

In this section, we overview the techniques for core network caching, RAN caching and D2D caching, and analyze their limitations.

3.2.1 Core Network Caching

Current widely deployed caching functions mostly take place within the core network. A Content Delivery Network (CDN) provides high availability and high performance by distributing the services spatially relative to end-consumers. The deployment of CDNs is known as a common approach to alleviate ever growing multimedia traffic.

Despite many advantages that CDNs bring, including increasing the number of concurrent users, decreasing content server load, etc, however, there are still some inherent limitations of utilizing CDN such as the ability to handle flash crowd traffic (a flash crowed occurs when there is an unexpectedly high amount of traffic during a short period of time). However, core network caching still has its natural limitations, no matter what kind of technique is adopted. First, the contents stored in the core network are still not “near” enough to the end consumers. In addition, core network caching just reduces the amount of duplicate contents transmitted in the core network; however, the traffic amount to the RAN still remains challenging, which poses high pressure on RANs’ backhaul.

3.2.2 RAN Caching

Deploying caches in the RAN is regarded as a promising way to break through the natural limitations left by core network caching. RAN caching improves mobile user experiences and alleviate the increasing pressure of traffic growth.

In this context, FemtoCaching [81] was proposed to cache popular contents at the very edge of a wireless networks (eg,. Base stations). FemtoCaching has many advantages, such as, reducing end-to-end latency which is mainly because of the distance between content and users are decreased. Furthermore, FemtoCaching utilizes the mobile backhaul more efficiently by reducing overall backhaul load. However, FemtoCaching also has its implicit problems such as limitation in caching size, content placement problem, cost of adding storage device, etc.

Furthermore, caching decisions are coupled not only because caches share backhaul links, but also because users might be in range of multiple cache-enabled base stations. These characteristics, together with the inherent volatility of the wireless medium, render caching decisions particularly difficult to optimize and, oftentimes, less effective e.g., in terms of the achieved cache hit ratio.

3.2.3 Edge Caching

Consider the novel cache-enabled MEC system shown in Fig. 19. Caching in the mobile edge network has been proved been beneficial. In edge caching, the MEC server can cache several application services and their related database and handle the offloaded computation from multiple users.

Thanks to the software defined environment, it is easy to incorporate more functionalities into the MEC server. In this study, we propose Caching as a Service (CaaS) as the functionality extension for MEC (Figure 19), which means that some popular contents can be cached in the MEC, and users can fetch these contents from an adjacent MEC servers as well.

Figure 19: Cache-Enabled MEC

3.2.4 D2D Caching

Another type of caching is D2D caching. In this approach, users collaborate by caching popular content and utilizing D2D communication.

Network with proximity-based D2D communication has emerged as a promising technology for enhancing current cellular network infrastructure as a means to improve throughput, latency, and energy efficiency. Moreover, in D2D-assited networks, the probability of establishing D2D connection is also constrained by the similarity of the content cached in each mobile device. D2D caching may bring some advantages such as

(i) Decreasing traffic load form fronthaul,
(ii) Decrease latency to access content,
(iii) Flexible reuse of radio resources, and
(iv) Increasing the number of users that can receive benefit from caching.

However, D2D caching introduces some challenges. for instance, it causes interference to cellular users and the main issue with D2D caching is that users may not be willing to participate in caching. Consequently, it is important to introduce an incentive to encourage users to participate in D2D caching.

3.3 Handling Flash Crowed Traffic

A flash crowed, refers to the sudden, large, and often unforeseen increases in request for a service during a short period of time. This phenomenon can significantly impede user QoS/QoE by exhausting the network bandwidth and processing capability so that requesters face large amount of delay, which in turn leads to user dissatisfaction.

A variety of techniques, such as Cloud-CDN [16] and proactive caching techniques [78], have been proposed to address the technical challenges introduced by the flash crowds. For example, the use of CDNs, allows the content to be closer to the end consumer. However, because of their inherent architectural limitations, existing CDN solutions are inadequate to deal with the exponential growth of multimedia traffic.

As for proactive caching mechanism, files are proactively cached during off-peak periods based on file popularity and correlations among user and file patterns based on the predicting the set of influential users to cache strategic contents and disseminate them to their social ties via D2D communications.

To this end, cooperative edge caching is the key to handle flash crowds. In cooperative edge caching, virtual machines are carved out of an underlying distributed edge cloud, forming a content distribution overlay. With the rapid elasticity property of distributed edge computing, once a flash crowd is detected, the computation and storage resources can be easily increase in real time to handle flash crowd.

In the next section, we will introduce a case study of cooperative edge caching technique where the MEC-assisted RAN handles flash crowd (i.e., traffic congestion in peak traffic period).

3.4 Case Study: Peak Traffic Congestion Control Mechanism for Delay Tolerant Traffics

In this case study, we propose an edge-assisted congestion control scheme which aims to alleviate network congestion in emerging 5G network environment. Supported by the MEC, the system is able to harvest context information for real-time RAN

condition. Such knowledge is then translated into the dedicated function known as CCE to make decision for selectively buffering traffic.

3.4.1 System Model

We consider a heterogeneous mobile network orchestrated by the SDN framework which is fully integrated with MEC [82]. This network is composed of Macro-cell Base Station (MBS), Small Cells (SC) and Mobile Node (MN). The MBS provides full coverage to subscribed MNs. The SCs are distributed within the MBS area to provide ample capacity to the few MNs within range. The system overview of this network is illustrated in Fig. 19.

In this work, the ETSI MEC [67] is considered as reference framework. It is assumed that there is a tight integration between SC and MEC in a way that a group of SCs are equipped with MEC server. Accordingly, MEC acts as an intermediate server so that DT contents can be temporarily stored and forwarded at later times. This significantly mitigates RAN load and improves resource utilization. Besides, the MEC server actively interacts with SDN through an API interface to facilitate traffic redirection.

Additionally, we assume that the core network entities have some capabilities, which would enable them to classify traffics and then, based on the QoS requirements, assign a deadline (i.e., a maximum delay it can wait for) to each DT traffics [83]. This can be achieved by leveraging Deep Packet Inspection (DPI) techniques. However, it should be noted that the traffic classification or DPI technique is not within the scope of this study.

Finally, it is assumed that MN traffics consist of two generic types of traffics namely, DT and delay sensitive traffics. Particularly, DT traffics refer to the type of traffics which are featured with long latency in comparison with that of delay sensitive. For instance, e-mails, updates of social networking portals and firmware updates [84] can tolerate delay with range from few seconds up to few hours. Note that such DT traffic also has the delay constraints or lifetime. The only difference is that its tolerant delay is much higher than delay sensitive traffic.

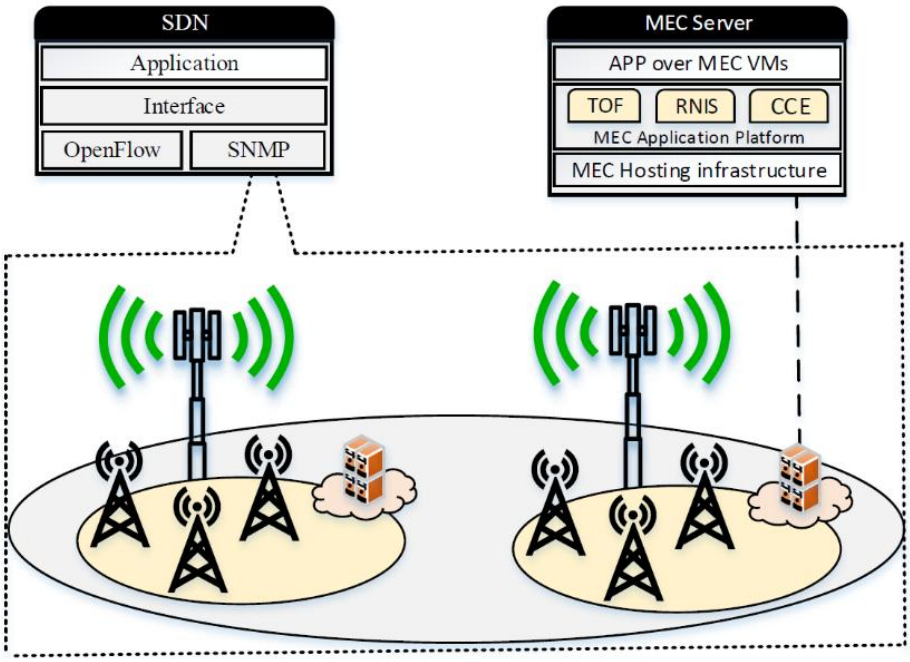

Figure 20: System Overview and module framework.

3.4.2 Congestion Control Mechanism

The goal of the proposed congestion control mechanism is to alleviate network congestion while makes better use of available network resources. A distinctive characteristic of our approach is that the MEC is playing an active role in this mechanism. The key idea is to intentionally delay DT content from being delivered and buffer it through an intermediate cloud server, with the goal of reducing RAN congestion, particularly during traffic peak hours.

We take advantage of RNIS cloud service introduced by ETSI, which is responsible for capturing real-time RAN condition. In addition, a dedicated function known as CCE is proposed, which takes the RAN context information into account and perpetually monitoring the deadline of DT contents.

With the proposed mechanism, a DT content is delivered depending on the network condition and their associated deadline. To illustrate, consider a situation that the network is overloaded, an operator can deliver DT content to an interested MN by temporarily storing content in MEC. In this context, the content will be transmitted to

intended MEC over the backhaul and buffer it there until the congestion ratio is reduced under acceptable threshold or before the deadline expires.

The proposed algorithm consists of the following sequential steps (Algorithm 1):

- **Packet inspection**: the network traffics are classified and then, each DT content assigned with a deadline based on their delay constraint.
- **Congestion detection**: in order to identify RAN congestion, CCE constantly monitors RAN condition and in the case of congestion, it provide feedback to SDN.
- **Redirection & buffering**: for each successive time the network is found to be congested, SDN redirect the DT content to MEC storage where the content will be stored.
- **Content delivery**: to capture the fact that buffered content may have a different deadline, CCE is monitoring the deadline of the content perpetually. If the deadline of a DT content is approaching, then the contents will abandon the storage and transmit to encountered requesters immediately. Otherwise, the content is kept until the RAN congestion is reduced to an acceptable level.

Finally, note that mobile users prefer to have data immediately, However, they will be willing to accept delay for DT traffic (e.g. Email, software update, mobile backup, etc) if the mobile operator provides appropriate incentive in form of instantaneous price reductions [83].

Algorithm 1 Congestion Control Algorithm

Input $deadline$ ▷ Assigned Deadline
Input RA_{status} ▷ RAN Congestion Condition
procedure CONGESTIONCONTROL(RA_{status},$deadline$)
    $RA_{congested} = isRadioAccessCongested()$
    while ($RA_{congested}$ = true) **do**
        SDN redirect DT traffics → MEC.
        if ($deadline$ → expire) **then**
            deliver content immediately → requester.
        else
            keep content until deadline expires.
            OR
            keep content until congestion is relaxed.
        end if
    end while
end procedure

3.5 Analytical Evaluation

It is assumed that each MN is interested in different content over time. The content can deliver to an interested MN either by direct transmission from the MBS or transmitting the content to SCs over the backhaul.

Two types of nodes are involved in this mechanism, requester of a content $R(t)$ and holder of content $H(t)$. The $R(t)$ is a MN that is interested in the content and not received it yet and the $H(t)$ is an MEC-assisted SC. The number of requesters, $r(t)$, shows how many users still need to be served at a given time. The number of holders, $h(t)$, represents the amount of resources used for serving user requests.

Concerning the holders of a content, we assume that MEC server stores the contents before their deadline expire, and during this time interval MECs always deliver them to encountered requesters through SCs. If a MN has been waiting for an amount of time, then the operator is obliged to deliver content before the expiration of their deadline. This is a reasonable assumption, since MECs are under the control of SDN, which knows the operating state of each MEC, and thus content discards (e.g., due to MEC overloads) can be avoided [72].

In order to analyses the performance of the proposed scheme, two key performance metrics are used. The first metric is the content delivery probability, which represent how much traffic can be buffered in the edge server. The second metric is content delivery delay, which indicates that how fast content can be delivered [85].

The number of holders and requesters can be approximated over time through a mean filed approximation and a resulting system of ordinary differential equation. According to [85], the fluid-limit deterministic approximation for the expected number of holders $h(t)$ and requesters $r(t)$ at time t, is

$$h(t) = h(0) \cdot \frac{(r_0 + h_0) \cdot e^{M_\lambda \cdot (r_0 + h_0) \cdot t}}{r_0 + h_0 \cdot e^{M_\lambda \cdot (r_0 + h_0) \cdot t}} \quad (1)$$

$$r(t) = r(0) \cdot \frac{r_0 + h_0}{r_0 + h_0 \cdot e^{M_\lambda \cdot (r_0 + h_0) \cdot t}} \quad (2)$$

where $h_0 = h(0^+)$ and $r = r(0^+)$ at $t = 0^+$, just after the initial placement of the content. M_λ denotes the meeting rates (i.e., the edge nodes can exchange data only when they come within transmission range) between two nodes i, j where nodes $i \in MN$ and $j \in SCN$. Furthermore, the meeting rates λ_{ij} are drawn from an (arbitrary) probability distribution $f_\lambda(\lambda)$ with mean value M_λ. Meeting duration is negligible compared to the time intervals between nodes, but long enough for a content exchange.

Based on (1) and (2), the desired performance can be calculated. Let us consider a requester $i \in r(0^+)$, and denote as T_i the time it receives the content. The probability that this (random) requester receives the content by a time t, i.e. $P\{T_i \leq t\}$, is equal to the percentage of offloaded contents by time t. Hence, we can write

$$P\{T_i \leq t\} = \frac{r_0 - r(t)}{r_0} = 1 - \frac{r(t)}{r_0} \quad (3)$$

which can be written as fellow:

$$P_{dlv} = 1 - \frac{r_0 + h_0}{r_0 + h_0 \,.\, e^{M_\lambda \cdot (r_0 + h_0) \cdot t}} \qquad (4)$$

where h_0 and r_0 are number of content holders and requesters, respectively.

Finally, the expected content delivery delay, which represents the MN's experienced delay until it receives the content, is given by

$$E[T_d|TTL] = \frac{1}{M_\lambda \cdot h_0} \cdot \left(1 - exp^{(-M_\gamma \cdot h_0 \cdot TTL)}\right) \qquad (5)$$

where TTL denotes the assigned deadline.

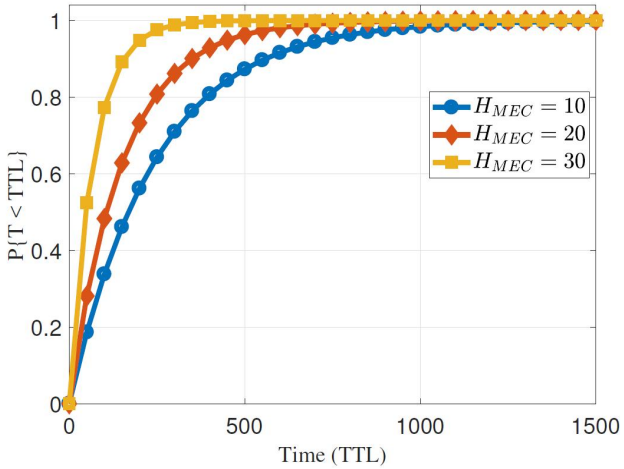

Figure 21: Delivery Probability P{T_d≤TTL} over time TTL (R(0)=50).

3.6 Results and Discussion

In this section, we present analytical results to illustrate the performance of our proposed mechanism. In this respect, we consider two performance metrics, namely probability of content delivery and content delivery delay. We consider 100 MNs reside in the network with an average meeting rate $M_\lambda = 3.3 * 10^{-5}$ meeting/sec [85]. The cellular network has to deliver DT contents to the MNs within deadline with range of 10 minutes, 30 minutes and 60 minutes.

Fig. 21 shows the delivery probability $\mathrm{P}\{\mathrm{T_d} \leq deadline\}$ of the content for the increasing number of the content holders (MEC server). Different density ratio of 10, 20 and 30 cloud servers are considered. It can be seen that increasing the number of content holders i.e. deploying more edge server, contribute to higher probability of

content delivery for the short deadlines. This is more evident in a case that $H_{MEC} = 30$, where the RAN able to alleviate the content delivery within short deadlines.

Moreover, Fig. 22 indicates the average delay a MN experiences until it receives the content in terms of the density of the edge servers for 10 servers, 20 servers and 30 servers. We compare the performance of the network with different deadlines of 10 minutes, 30 minutes and 60 minutes. It is clear from the figure that increasing the deadline of DT content yields lower content delivery delay. This is due to the fact that deploying more edge server can expedite content delivery by buffering more DT content especially during RAN congestion period, which result in lowering the content delivery delay.

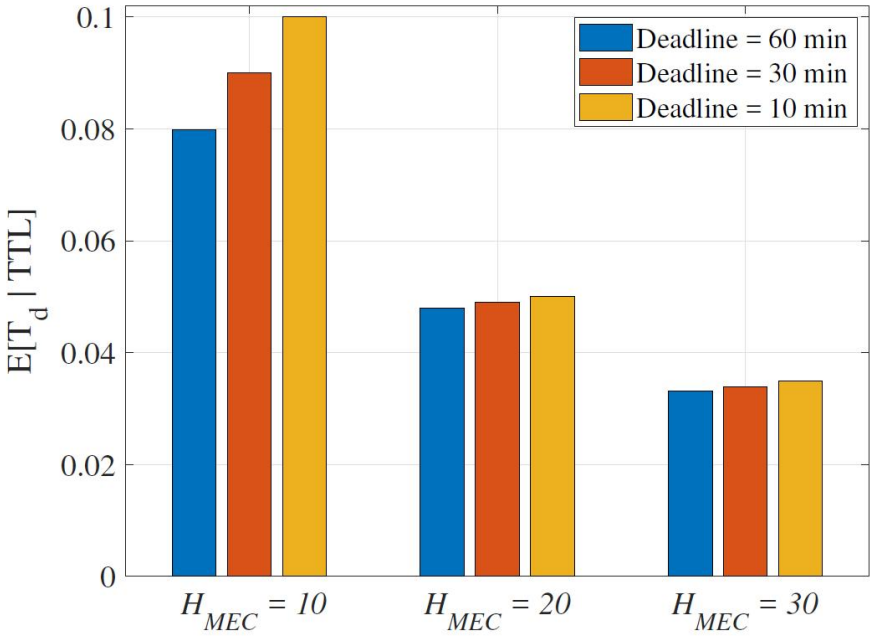

Figure 22: Expected delivery delay E{T_d ┤|TTL} for different deadlines.

3.7 Conclusion

The explosion of data traffic has posed great challenges in terms of congestion and delay to the current networks. To cope with these two challenges, we have proposed an edge-assisted congestion control scheme which aims to alleviate network congestion in emerging 5G network environment. Supported by the MEC, the system

is able to harvest context information for real-time RAN condition. Such knowledge is then translated into the dedicated function known as CCE to make decision for selectively buffering traffic. Performance evaluation results were presented to demonstrate the performance improvement of the proposed scheme.

5. Active Queue Management implementation and evaluation in 5G. Testbed and benchmarking

5.1 Introduction

The new 5G standard is emerging with the aim to support new use cases and business models where predictability and determinism will play a major role. The new 5G architecture is emerging as a Service Based Architecture design where the functionalities will be split between different entities in contrast with the monolithic approach that has been used until now. From the Fig. 23, extracted from the TS 23.501, it can be noticed that most of the new entities will focus on the control plane.

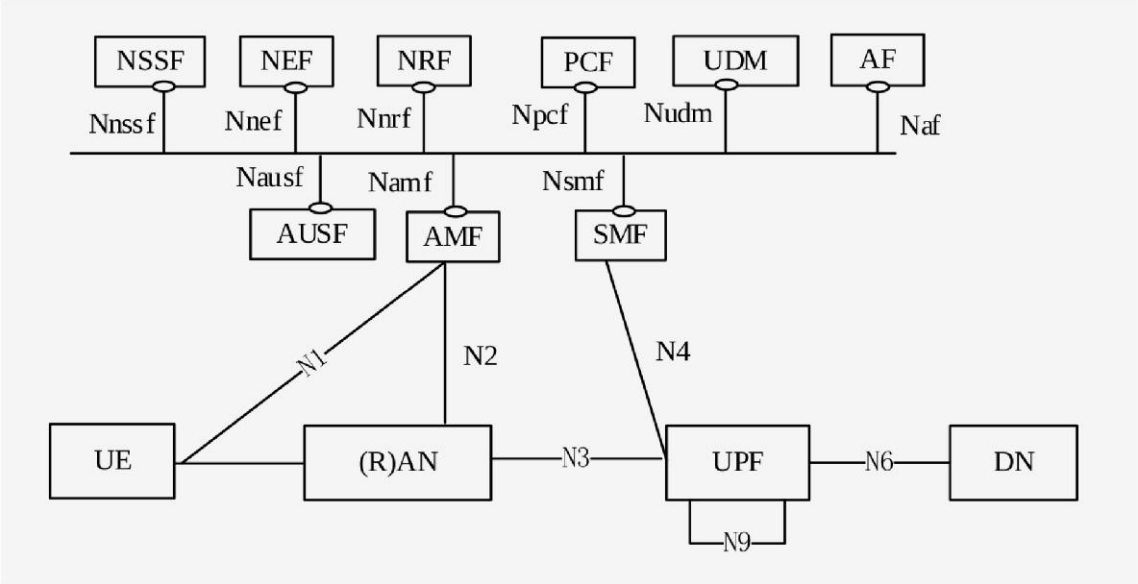

Figure 23: 5G Service Based Architecture (extracted from TS 23.501 [86])

Equally important, the new standard brings a new QoS model, as observed in Fig. 24 that will enable new business revenue models. Data packets will arrive to the UPF through the N6 interface, where the Packet Detection Rules (PDR) established by the SMF and mapped into QoS Flow Indicators (QFI) will classify them. QFI is a scalar that is used as a reference to a specific QoS forwarding behavior (e.g., packet loss rate, packet delay budget) to be provided to a 5G QoS ow in [86]. As described by 3GPP [86] "All traffic mapped to the same 5G QoS Flow receive the same forwarding

treatment (e.g. scheduling policy, queue management policy, rate shaping policy, RLC configuration, etc.). Providing different QoS forwarding treatment requires separate 5G QoS Flow".

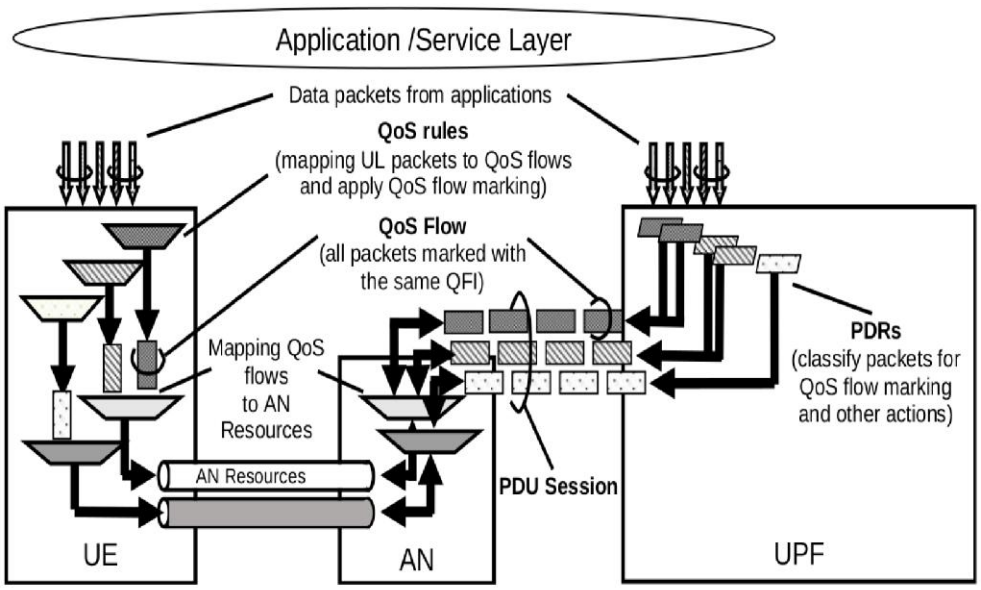

Figure 24: 5G QoS (extracted from TS 23.501 [86] 5G QoS model)

Then, the Service Data Adaptation Protocol (SDAP) [87] handles the flows, which is responsible for mapping the QFI flows into Data Radio Bearer (DRB) flows. The DRB flows will go through the PDCP entity, which is responsible for header compression, ciphering and in-sequence delivery among other tasks.

5.1.1 General outline and scope

A crucial challenge for achieving a deterministic delay is the undesirable latency that occurs when network buffers accumulate a significant amount of data. This problem, known as buffer bloat [88], happens in 5G since the 5G Access Network (5G-AN) equipment is deployed with large buffer sizes in order to minimize the possible throughput lost due to the physical channel capacity variability in the radio access. This conservative but usual approach, creates large unnecessary delays for traffic flows that share the same buffer. However, since there will be services mapping to the same

QoS class, it is critical to have a method that ensures the required delay, while achieving fairness in the individual egress rate. Although there are Active Queue Management (AQM) network algorithms such as CAKE, FQ-CoDel or CoDel that target to reduce the delay on bottleneck links, their applicability in 5G networks has not been deeply studied before.

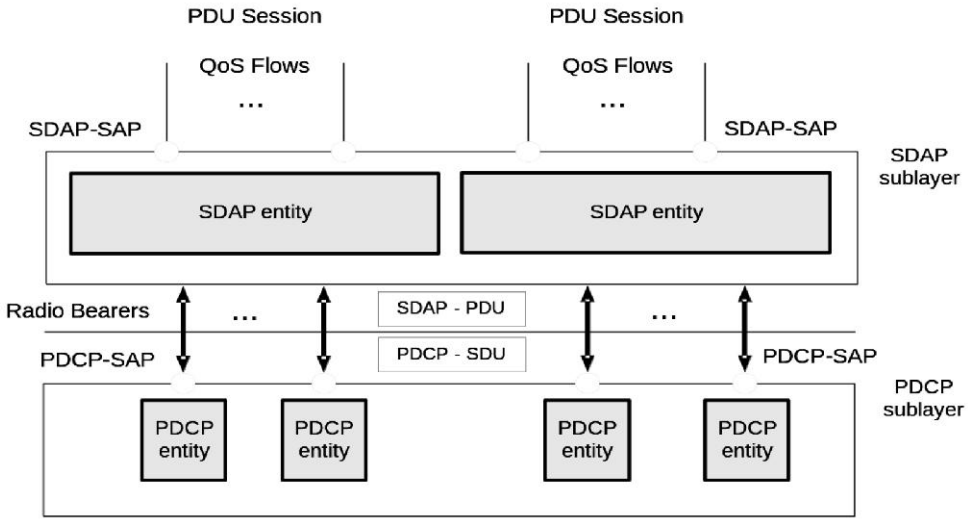

Figure 25: SDAP mapping QFI flows (extracted from TS 37.324 [86].)

5.1.2 5G QoS Model

Some of the real time services that are predicted to emerge from the 5G will require time constraints in data delivery. With this aim, the 5G standard introduces a detailed QoS model, which defines QoS criteria for many use cases [86] and business. It is envisioned that the new standard will support a very heterogeneous range of different services with very different characteristics from Real Time Gaming, IP Multimedia Core Network Subsystem (IMS) signaling or Video Streaming.

The QoS is basically defined with 6 characteristics. The resource type, the Priority Level, the Packet Delay Budget, the Default Maximum Data Burst Volume, the Default Averaging Window and the Maximum Data Burst Volume.

The Resource Type determines if dedicated network resources related to a QFI Guaranteed Flow Bit Rate (GFBR) values are permanently allocated. Guaranteed Bit Rate (GBR) QFI can typically be allocated dynamically. The definition of Packet Delay Budget (PDB) and Packet Error Rate (PER) are different for GBR and Delay-critical GBR resource types, and the Maximum Data Burst Delay (MDBD) parameter applies only to the Delay-critical GBR resource type.

The Priority Level indicates the priority in scheduling resources among the QFI. This value should be passed into the (R)AN so that the SDAP can map the QFIs correctly into the DRBs. It is to be taken into account that there will be less DRBs than QFIs and therefore, different QFIs will have to be mapped into one DRB. In a congested scenario, the priority level should be used to share the scarce resources according to the priority, while in a non-congested scenario; the priority level should be used as guideline in order to distribute the resources.

The PDB defines an upper bound for the time that a packet can be delayed between the UE and the N6 interface termination, just before the UPF. The PDB is also used as a scheduling parameter for priority weights and HARQ operation points. For GBR QoS Flows using Delay-critical resource type, a packet that surpasses the PDB is considered lost if the data burst is not exceeding the MDBV and the GFBR. For GBR QoS Flows with GBR resource type, the PDB is interpreted as the maximum delay with a confidence of 98% if the QFI is not exceeding the GFBR. Services using Non-GBR QFIs will be the first to suffer from congestion-related packet drops and delays. In uncongested scenarios, 98% of the packets should not suffer from larger delays. A packet that suffers more delay than the one assigned by the PDB, should not be discarded or either added to the PER for Non-GBR and non-Delay Critical GBR. On the other hand, Delay Critical GBR packets that exceed the PDB are added to the PER and may be discarded or forwarded.

The PER is defined as the ratio of the packets that have been processed by the sender of a link layer protocol (eg., RLC in (R)AN) but were never received by the upper layers in the receiver (eg., PDCP in (R)AN).

The Averaging Window is the duration over which GFBR and MFBR are calculated.

The Maximum Data Burst Volume is the maximum data that the (R)AN has to serve, respecting the PDB in an averaging window.

\As it can be noticed, some of the fields remain unspecified. This clearly shows the problem of merging all the different characteristics while remaining efficient in all the entities that transport the data due to the variability on the radio channel. In fact, due to the non-predictable resource model nature of wireless communications, no data delivery can be guaranteed in adverse conditions. The standard reminds that it may also happen that a GBR flow must be degraded to a non-GBR flow. This constraint unfortunately cannot be avoided and will remain as a non-trivial problem for every wireless network.

5.1.3 Traffic data transport protocols

3GPP defines the PER for the 5G standalone scenario in terms of packet delivery and packet receive rate from the RLC and PDCP entities, respectively. On the other side, the PDB is defined between the N6 interface and the UE. In any case, it tries to maintain the specifications in an abstract mode without tightening to any implementation specification. However, the vast majority of the data network abstractions used by 3GPP will be materialized in the Internet. Therefore, according to the OSI model, the layer 4 is responsible for packet delivery. A real scenario cannot exist without the Internet constraints to the data packets delivery. In the next following section we analyze the most popular data transport protocols.

5.1.3.1 Transmission Control Protocol

Transmission Control Protocol (TCP) is an OSI layer 4 transport protocol. It is a stream based protocol, rather than packet based protocol, that guarantees data delivery. If one packet is lost, TCP will take care and deliver it again until the receiver acknowledges that has received the corresponding packet. TCP has four interwinded [89] algorithms. Slow start, congestion avoidance, fast retransmition and fast recovery. At slow start, TCP will increase its transmitting window for every ACK received until a limit is reached. At that moment, it will pass to congestion avoidance, where segments delivery rate decreases. If three or more unordered ACKs are received, TCP interprets

it as an indication that a segment has been lost and performs a retransmission of the unacknowledged segment. After a fast retransmission, the fast recovery takes place. As TCP is aware that there are still some packets in flight and that going to a slow start would mean an abrupt throughput lost, fast recovery is an improvement that allows maintaining the throughput high under moderate congestion. It is crucial to understand correctly its mechanisms as TCP will play a major role with more than 90\% of total flows based on it. In the last years, new variants were successfully deployed.

TCP CUBIC is “the facto“ TCP implementation enabled on most servers in the world. It is also the implementation used by default in Linux kernels 2.6.19 and above, as well as in Windows 10.1709 Fall Creators Update, and Windows Server 2016 1709 update [90]. It is a packet lost based algorithm.

TCP BBR is a new version of TCP implementation that emerged from Google at 2016 with the aim to avoid saturating the queues in the Internet. It is actually used in production by some Google services and is delay based, meaning that its state does change based on the Round Trip Time of the packets. Unfortunately, it has been proven to have some weakness and is not massively deployed [91].

5.1.3.2 User Datagram Protocol

User Datagram Protocol (UDP) is a packet based protocol instead of a stream-based protocol. It has not intrinsic mechanisms to assure that the data has been delivered. Normally used in real time traffic, many routers do discard its packets since it can be used for DoS attacks.

5.2 Different traffic flow constraints

Different applications will present very different traffic flows characteristics according to their needs. Some may need to transmit very few bytes rapidly, while others will have to transfer huge amounts of data. This heterogeneous nature of the flows, makes it difficult to optimally satisfy the access to the share data links successfully. It seems reasonable to wait several more seconds for the last software update, while a mobile

multi-gaming experience will be ruined if the scenario is refreshed every 250 ms. Since all the data will share the radio access as well as some intermediate queues, a solution for satisfying both constraints while maintaining a good user experience is desirable.

5.2.1 Elephant vs mice flows

On the modern Internet, there are some few flows that last for big periods of time that tend to disproportionately occupy the bandwidth greedily. This phenomenon punishes short flows (mice flows) that normally do not last for long period of time, but have to suffer long latencies if they share the data link with the elephant flows. If these mice flows have some time constraints, it will be very difficult to fulfill them since the queues will be already occupied and the packet will have to suffer a big sojourn time. Therefore, some restrictions to the elephant flows are normally imposed by the schedulers with the aim to favor mice flows. A very special but very common case is the HTTP traffic. There, many parallel flows are opened concurrently with the aim to reduce the latency. This bursty traffic is a common traffic pattern that needs to be studied carefully if a good QoS in the 5G network is desired.

5.2.2 3GPPP Technical Specification absences

In the Section 2.1.2, the 5G QoS model from the TS 23.501 has been explained. 3GPP specification does not, however, specify the profile of the traffic for Guaranteed non Critical Bit Rates. This can have negative consequences if a queue in the system cannot accept a bursty traffic profile and discards packets. Another too optimistic value is the Default Averaging Window. It is fixed at 2 seconds for Guaranteed Bit Rates. This value does not sufficiently constraint many applications where such a delay may not be acceptable.

3GPP does not concretize the transport layer. Therefore, and even though most of the flows will be composed of TCP traffic, the standard avoids the explicit term. This is a good approach since the system is maintained independent from the transport layer protocol, but since most of the services will run on TCP, it deserves a more careful analysis in this work.

3GPP does not define some constraints in the amount of packets that the buffers should accept. Deploying buffers with excessive size can have very bad consequences from the delay perspective.

Many congestion control algorithms in TCP rely on lost packets to adjust its transmission rate. Therefore, the packet accumulation is an unavoidable phenomenon that will take place due to its design nature. Packet drop rate is used by TCP to try to guess the available bandwidth between two endpoints of a connection. If the buffer capacity is too large, TCP will not be able to correctly measure the available bandwidth, will deliver more packets than the egress rate and packets will start accumulating at the bottleneck link forming a queue and affecting the latency.

5.2.3 Real Time Applications

There are many business models foreseen that will need to deliver some amount of data in a restricted amount of time with some guarantees. Due to the nature of the radio in the 5G, such assumptions can be theoretically impossible while practically feasible. Another important aspect that has not catch enough attention in the 5G research community is the inevitable bufferbloat problem that arises from any packet based network. Virtualization, and with it, slicing, which has been advertised as the future candidate to resolve prioritization problems, will suffer from the same constraints from the moment that has to share different flows. Moreover, the abstraction of the hardware does not mean that the hardware architecture will change. Therefore, the virtualized systems will need an entity that orchestrates the hardware access.

Real time constraint is defined as the deadline in which an event has to occur. Therefore, the systems that run within real time constraints cannot afford many abstraction layers as every layer of abstraction makes it more difficult to assure that the event has occurred in the time frame. Predictability is prioritized over performance. A complex calculation will have to abandon the CPU to other processes, with all the context switching burden involved, in order to assure that every process receives the agreed CPU time in a sliding window. It should be noticed that real time applications do not automatically mean low-latency, it is just focused on the fact that a process will receive the agreed CPU time in an agreed time period. However, when we move to

the 5G, the time periods remain restricted. A typical radio frame lasts 10 ms and the PHY layer will have different Transmission Time Interval (TTI) according to the mode that they are serving, but it will last less than 1 ms in every case.

There are 3 different types of real time criteria: i) Hard: Systems must deliver the action within the time window, ii) Firm: Systems can infrequently fail but do degrade the QoS and the results cannot be used anymore, and iii) Soft: Systems can infrequently fail but do degrade the QoS and the results can be used.

As it can be seen in TS 23.501 [86], the delay is measured between the N6 interface at UPF and the UE. It is therefore from vital importance to know the exact delay that a packet has already suffered until it arrives to a new entity, especially since the Service Base Architecture promotes the separation of the functionalities in different entities. 3GPP unfortunately does not describe how such a method can be achieved. If this entities are distributed among different hardware, they will inevitably have different clocks. Therefore, to measure the time that a packet has already spent in the system remains challenging. The Precision Time Protocol described in [92] does assume the same upload and download path time. Moreover, it assumes a non-congested scenario. With this assumptions, it is really hard to be able to serve real time applications efficiently in congested scenarios, precisely at the moment where managing priorities and delays becomes more important.

5.2.3.1 Types of Services

From the TS 23.501 [86], 3 different types of traffic can be distinguished for different types of services with different meanings.

Delayed-critical Guaranteed Bit Rate: This kind of traffic has the most restrictive characteristics. This is the only kind of traffic where a Maximum Data Burst Volume is defined in a time period with a Packet Delay Budget. There is a GFBR and a Maximum Flow Guaranteed Rate (MFGR). The Maximum Data Burst Volume remains small in order to assure the delivery without degrading the bandwidth of the whole system. This kind of resource type should be used with caution as can easily starve

the other kind of traffic and does not respect any fairness. Typical services inside this category are used in discrete automation.

Guaranteed Bit Rate: This kind of traffic is the second most restrictive.

3GPP does not provide a Maximum Data Burst Volume. It is also assumed in general, that if the service is sending at a smaller or equal rate than the GFBR, congestion related packet drops will not occur. Typical services that lay inside this category are Real Time Gaming, some V2X messages or video live streaming.

Non-GBR: This resource type is the most generic. It also has some Packet Delay Budget, but it is the resource type that will be firstly dropped in a resource scarce situation. It is also the resource type that most applications will be tied to. The 5G QoS entities will try to respect the packet delay budget and the packet error rate, but in case of scarce resources, they will be the ones that are first discarded. Typical services are TCP based streaming or IMS Signalling.

5.3 5G QoS enablers

In order to assure the QoS in the 5G, the mentioned models with each particularities have to be taken into account. In this section, the mechanisms that will be used to achieve the required PDB are presented.

5.3.1 Active Queue Management in the 5G Networks

Active Queue Management (AQM) has emerged as the correct tool in order to maintain the queue occupancy of the buffers low. This will play a major role in 5G, since many new services have delivery restrictions that need to be fulfilled.

5.3.1.1 Bufferbloat problem

Due to the low memory prices, routers are deployed with large buffers that can hold several megabytes of data, which can introduce delays in the order of tens of seconds [88]. This completely distorts TCP's congestion control algorithm feedback and, thus,

nullifies its ability to quickly adapt the transmission rate to the data link capacity. Therefore, TCP creates large buffers that cause important packet sojourn times. TCP's congestion control algorithm does achieve a steady state but the buffers are unnecessarily overloaded.

5.3.1.2 Benefits of AQM

Active Queue Management was designed with the aim of maintaining the buffers at the optimal size while achieving the maximum possible throughput. When the queue starts getting overloaded, it discards packets as a measure to notify the sender that it should reduce the transmission rate. Therefore, the queue is maintained relatively empty, and greedy flows cannot saturate the link. If another flow shares the same queue, it will not suffer big sojourn times. As a consequence, time constraint low latency traffic will be able to be delivered successfully even if the queues are shared among different flows.

5.3.1.3 Types of AQM

AQM mechanisms can be separated in two different groups. In the first group, AQM mechanisms that rely on the queue growth can be classified. In this first group Random Early Detection (RED) [93] appeared as its pioneer. Even though the first results were very promising, it never really got a wide implementation in the consumer devices. RED considers the growing rate of the queue as a congestion symptom, and increases the probability of discarding a packet accordingly. While persistent queues indicate congestion, the growing rate of a queue does not. The bursty traffic nature of concurrent TCP sources can grow and shrink the queues before RED can effectively react accordingly [94]. Therefore, the RED algorithm was never widely adopted in consumer electronics.

Some years later, the Controlled Delayed (CoDel) algorithm appeared. This time, the packet sojourn time was taken as the principal measurement method. Due to the bursty nature of TCP, queues are formed. But these queues disappear after a Round Trip Time (RTT). Hence, CoDel classifies the type of queues into "good" queues that emptied in a time interval, and "bad" queues that are persistent in the time [95]. A

packet is dropped from bad queues and the interval time is reduced. The bad queue is monitored to ascertain that it has been emptied in an interval. In case that the queue is still persistent, CoDel will drop another packet, reduce the interval time and continue recursively.

5.3.2 Scheduling

Scheduling is the other important pillar when it comes to QoS. Higher priority traffic should access first the resources. This is a crucial aspect for low-latency delay sensitive traffic. In any way, a balance has to be reached to avoid starving some flows, while prioritizing others, especially for Non-GBR traffic.

5.3.2.1 State of the art in scheduling techniques

One of the first network algorithms that addressed such a problem is the Stochastic Fair Queuing (SFQ) [96]. Flows are hashed and assigned to different queues. Every active queue is assigned an equal egress rate in a Round Robin manner. However, due to the hashing nature, two flows can end sharing a queue, splitting each flow's theoretical corresponding share of bandwidth. This situation is partly alleviated by periodically adding a perturbing value to the hash function that rehashes the flows, thus reducing the possibility of different flows sharing the same queues for large periods.

One improvement over SFQ is the Deficit Round Robin (DRR) [97]. In SFQ, different flows could have different packet size and, therefore, the fairness would be packet-wise but not bit-wise. Traffic sources that send packets with smaller size, would get less than its corresponding bit-wise bandwidth. DRR adds a quantum value that measures how much bandwidth corresponds to each active queue. If the packet at the queue is smaller than the quantum value, the packet is subtracted and the quantum value is reduced by the packet size. If, on the contrary, the packet size surpasses the quantum size, the quantum value is accumulated for the next round. In this way, a bit-wise fairness is assured.

One more modern approach is the DRR++ [98]. In this scheduler, the latency sensitive traffic demands are handled. The sender agrees to send less than one quantum during a round of the scheduler. As long as the sender does not surpass this rate, the scheduler guarantees that only high priority traffic will delay this flow. If the sender surpasses this rate, these new packets will not be taken into account for the current round. In this way, the high priority traffic is not lost, even if the traffic is transported through a bursty protocol.

5.3.3 Implementation challenges in the 5G

The 5G network presents several peculiarities that deserve special attention. The data network has its own independent protocol stack that must fit with the 5G specification. The data network is prepared to deal with the loss of IP packets, but it may be inefficient to deal with packets already processed by the 5G network. Therefore, if AQM mechanisms are to be implemented, it looks like a natural approach to implement them before the 5G stack does process any data. This happens at the UPF that is responsible for mapping the IP flows to QFI flows. However, in a normal scenario, the bottleneck of the system will be certainly formed at the radio access. The bandwidth of the radio communications is certainly more restricted in today's networks than the wired bandwidth. Therefore, even though there exist some preliminary studies [99] of limiting the bandwidth at the UPF, in order to artificially generate the bottleneck at the UPF, its wide implementation seems unrealistic. At [99], the Round Trip Time (RTT) of the packet is measured and the egress rate of the UPF entity is adapted accordingly. The egress rate is constraint to the maximum bandwidth of the link. This ensures that the packet accumulation will happen at the UPF queues rather than at the 5G-AN entities. This approach presents several problems. In the first place, the 5G networks do dynamically and abruptly change their bandwidth due to its dependence with the radio channel conditions. If more bandwidth is available, bandwidth will be squandered as the egress rate control mechanism depends on the feedback from the RTT and needs some time to adapt correctly. Moreover, the UPF can reside relatively far from the 5G access network, which will increase the response time due to bandwidth vari- ability. Secondly, this approach relies on protocols that send some feedback to the sender. While most of the 5G traffic

will certainly be implemented in such a manner, low-latency time constraint traffic may not rely on a feedback from the transport layer (e.g. QUIC, SCTP).

Once the UPF forwards the data packets, the first entity at the 5G-AN that receives the QFI flows is the SDAP layer, which does the mapping of QoS classes to Data Radio Bearers. According to [86], the SDAP entity is not able to schedule the packets or accumulate them. It seems like a good candidate to enhance its capabilities to store the QFI flows packets and schedule them, since the flows will be reduced from a maximum of 64 QFI to a maximum of 11 DRBs. In any case, if the buffers at the following layers are not limited, the bottleneck will not be generated at the SDAP, and the AQM mechanism will have no effect.

The last entity where the packets may be accumulated is at the RLC buffer. In the 5G, the capability of the RLC entity to aggregate packets was deprecated, enabling the possibility of dropping packets without affecting two different flows. This new approach facilitates the segregation of packets, which is crucial for prioritization. Priority traffic can be firstly scheduled avoiding the large sojourn time that may occur if priority traffic has to share the queue with bulky traffic.

The use of a SFQ at the Packet Data Convergence Protocol (PDCP) entity has been explored by [100] with a SFQ mechanism implemented at the Packet Data Convergence Protocol (PDCP) entity. The PDCP entity, resides just before the RLC and is responsible for header compression, ciphering and in sequence delivery among other tasks. This approach segregates the traffic that has already been aggregated into a QoS Flow Indicator (QFI) in order to fairly distribute the egress rate between different 5-tuple flows. QFI is a scalar that is used as the finest granularity reference to a specific QoS forwarding behavior (e.g., scheduling prioritization, queue management, packet loss rate, packet delay budget). All the traffic mapped into a given QFI must experience the same forwarding treatment according to [86]. Therefore, segregating the traffic from a QFI is a non-3GPP compliant technique. At [100], the possibility of implementing a communication mechanism between the RLC and the PDCP is also explored, in order to maintain the buffers at RLC in an optimal size.

A natural deployment for AQM mechanisms in 5G is the Radio Link Control (RLC) layer where data is buffered, segmented, reordered and transmitted to the following layers [101]. At [102], a modified version of the RED algorithm [93] at RAN's Layer 2 RLC entity is proposed. RED considers the growing rate of the queue as a congestion symptom and increases the probability of discarding a packet accordingly. While persistent queues indicate congestion, the growing rate of a queue does not. The bursty traffic nature of concurrent TCP sources can grow and shrink the queues before RED can effectively react accordingly [94]. Thus, the RED algorithm needs some tuning and can conceivably cause problems if it is implemented without a tedious study of the traffic patterns. Therefore, the RED algorithm was never widely implemented [103].

Another particularity in the wireless communication systems resides on the bandwidth variability. The bandwidth can substantially change due to the physical channel condition. This unpredictable fact makes the wireless domain a special use case scenario [104] if predictable low-latency is required.

The promising development of millimeter Wave [105] will make the bandwidth even more unstable, which will make setting the optimal number of packets in the queues very challenging.

5.4 Proposed Solution and Implementation

In order to tackle the QoS problem in 5G networks, we consider a full 5G QoS Scenario as the one shown in Fig. 26.

Although the presented QoS scenario describes a downlink scenario, similar SDAP and DRB mappings are also present in the uplink scenario. In this scenario, we model the entities that play a central role in the 5G QoS download scenario.

The data packets arrive from the Data Network (DN) to the UPF. These packets are firstly enqueued and then mapped to QFI flows according to PDR [86]. Once they arrive to the 5G-AN, these data packets are handled by the SDAP [87], which is responsible for mapping the QFI flows into DRB flows. Finally, the MAC

scheduler is responsible to deliver every TTI the data quantity requested by the Physical Layer (PHY), through the Downlink Shared Channel (DL-SCH) transport channel.

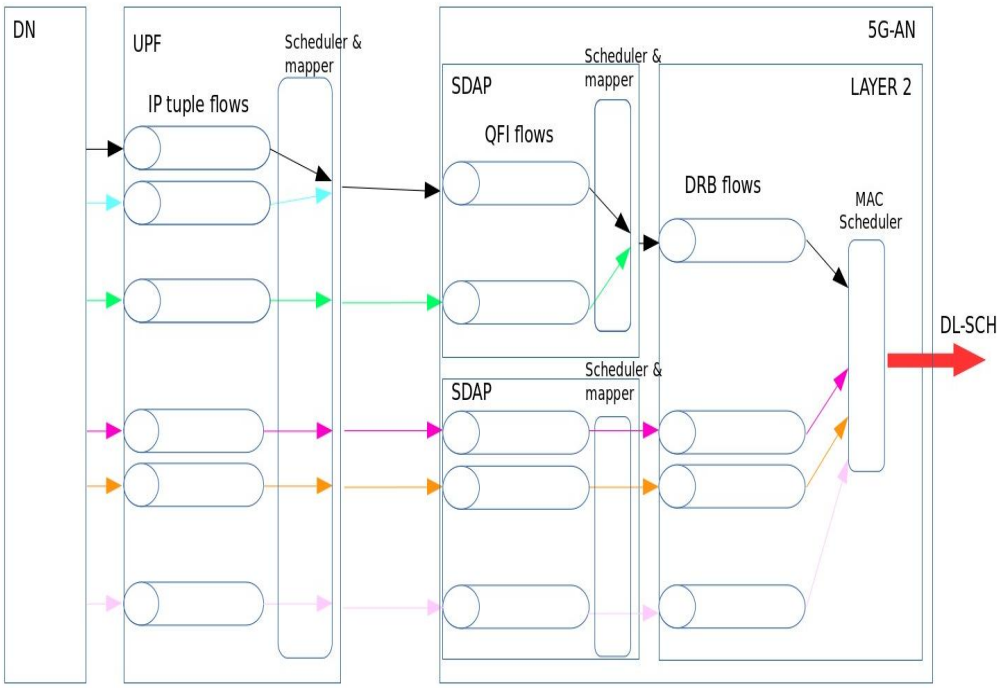

Figure 26: 5G QoS Senario

Maximizing the throughput while prioritizing the packets and reducing the latency is a complex task. On the one hand, if traffic with high priority arrives, it is desirable to forward it as soon as possible to the DL-SCH transport channel. Once packets are aggregated into a flow, they cannot be segregated again [86]. Therefore, if a high priority packet is forwarded to a congested queue, the packet will suffer a big sojourn time until the queue is emptied. Hence, it would be advisable to maintain the buffers as empty as possible. On the other hand, for each TTI, the MAC scheduler should send as many data through the DL-SCH as requested by the PHY entity in order not to squander any transmission possibility. Otherwise, the throughput will be reduced. Hence, it would be advisable to maintain the buffers as full as possible. In addition to the problem described above, the number of packets required by the PHY entity

changes dynamically due to diverse factors (e.g., radio channel conditions, HARQ retransmissions).

Unfortunately, many congestion control algorithms in TCP rely on lost packets to adjust its transmission rate. Therefore, the packet accumulation is an unavoidable phenomenon that will take place due to its design nature. Packet drop rate is used by TCP to try to guess the available bandwidth between two endpoints of a connection. If the buffer capacity is too large, TCP will not be able to correctly measure the available bandwidth, will deliver more packets than the egress rate, and packets will start accumulating at the bottleneck link forming a queue.

In order to tackle the aforementioned problems, we explore the following solutions. In the first place, we implement the CoDel AQM algorithm [95]. CoDel operates over an interval time. During this interval time, it measures the sojourn time of the packets and the lowest sojourn time is saved. When the last packet of the interval is dequeued, if the lowest saved value exceeds a target time, this last packet is dropped as a measure to notify the sender that excessive buffering is happening, and the interval time is reduced. Hence, CoDel adapts efficiently to abrupt changes in the egress rate, which makes it a good candidate for 5G networks.

In the second place, we propose to maintain DRB queues on 5G QoS scenario limited to values slightly above the order of magnitude of the maximum possible egress rate from the MAC scheduler. We do not study the values below that rate, as it would just sacrifice throughput. This principle is well known in other disciplines that have to deal with queues that are formed in the lower layers. Network Interface Controller (NIC) software developers vary the queue limits according to the egress rate in order to avoid large sojourn times at thenetwork card without squandering transmission possibilities [106].

We combine both of the aspects and heuristically find the best combination that maintains the throughput high and the latency of delay-sensitive traffic low.

5.4.1 Implementation

In order to evaluate our proposed AQM based solution and compare it with the baseline solutions, we implement a queue system that emulates different 5G entities and their queues presented in Fig. 26. As per QoS traffic, we define a delay-sensitive traffic flow, taking gaming application as a reference [107]. For this, we configure the well-known ping tool with a realistic gaming traffic packet size of 100 bytes and an interval that varies from 10 to 70 ms in increments of 10 ms in line with [107]. As background traffic, we use a second flow of TCP, generated by the iperf3 software. We run our experiments for 30 seconds for each ping interval.

To implement the evaluated queue management solutions realistically, we forward the IP packets from the kernel space traffic to the user space, where they are processed with these queue management solutions. The forwarding of the packets from the kernel space to the user space is achieved through iptables, by applying the NFQUEUE traffic control netfilter queue binding.

We use two PCs as the sender and the receiver of these flows. The sender PC acts as the Data Network (DN) that generates the different traffic flows, and the receiver PC implements all the 5G QoS queuing scenario. Note that the sender uses the TCP CUBIC congestion control algorithm. The receiver PC has an Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz, while the sender PC has an Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz. A TP-LINK TL-WR841N router with Ethernet cables is used to connect both PCs.

We classify the flows according to their source IP address/port number, destination IP address/port number and the protocol in use, known as the 5- tuple. These values are hashed with the Jenkins hash function and classified into IP tuple flows. A mapper at UPF, multiplexes the IP tuple flows into QFI flows. We implement a SFQ [96] as the UPF scheduler where 10 IP packets are egressed every 1 ms. We enhance the SDAP [86] capabilities from mapping to scheduling and mapping. We implement the SDAP scheduler as a Round Robin scheduler where 10 packets are egressed fairly among active queues every 1 ms. The QFI flows are mapped into DRB flows by the SDAP entity. Finally, the MAC scheduler egresses 10 packets fairly among the active DRB

flows every 10 ms for a theoretical maximum throughput of 11.68 Mbps considering a MTU of 1500 bytes and excluding the TCP/IP headers. Once a packet is egressed from the MAC scheduler, it is forwarded to the kernel space with a forward verdict. When an AQM mechanism decides to drop a packet, the discard verdict is passed to the NFQUEUE that informs the kernel space to drop the packet.

CoDel is well known as a knob-less QoS solution. It is governed by two variables, the interval time and the target time. The target time defines the maximum sojourn time that a packet should experience under no congestion. The interval time defines the duration in which the queue should overcome the congestion state. In [104] the target time is recommended to be set at around 5% of the proposed interval time of 100 ms. Under our test conditions, CoDel would classify all the packets into the dropping state with direct consequences for the bandwidth [108], since the MAC scheduler forwards 10 packets every 10 ms in discrete time. With the default CoDel parameter values, all the packets would be dropped in our scenario. Hence, we increased the target time to 15 ms and the interval time to 300 ms, while meeting the requirements of setting the target time close to the RTT. This value has been heuristically proven to be correct for the current scenario.

We implement and evaluate two scenarios. In the first scenario, two queues at the UPF entity are formed according to their hashed 5-tuple. The scheduler at UPF maps both flows (e.g., TCP bulky flow and the ping flow) into a single QFI flow. The newly implemented SDAP scheduler maps this flow into a DRB flow. This corresponds to the scenario, where different services are mapped to the same QFI class. Since there are 64 QFI classes and many types of services, this is an expected scenario in 5G. In the second scenario, the UPF scheduler maps the two flows into two different QFI flows, and the SDAP scheduler maps both of the flows into a single DRB flow. The two flows maintain an independent path until the DRB queue, where they are aggregated.

We evaluate four different solutions within these scenarios. In the first solution, which is similar to the default one used in the current cellular systems, buffers are unlimited and no AQM mechanism is implemented. In the second solution, the DRB buffer capacity is limited. The SDAP scheduler does not forward any packet that would surpass the DRB limited buffer capacity and no AQM mechanism is implemented. In

the third solution, CoDel is implemented at the DRB queue without any buffer limitation. Finally, our proposal of using CoDel AQM for the QFI queue and limiting the DRB queue capacity is evaluated.

We measure the buffers load status, the TCP throughput and the ping delay metrics in order to evaluate the scenarios and extract a conclusion.

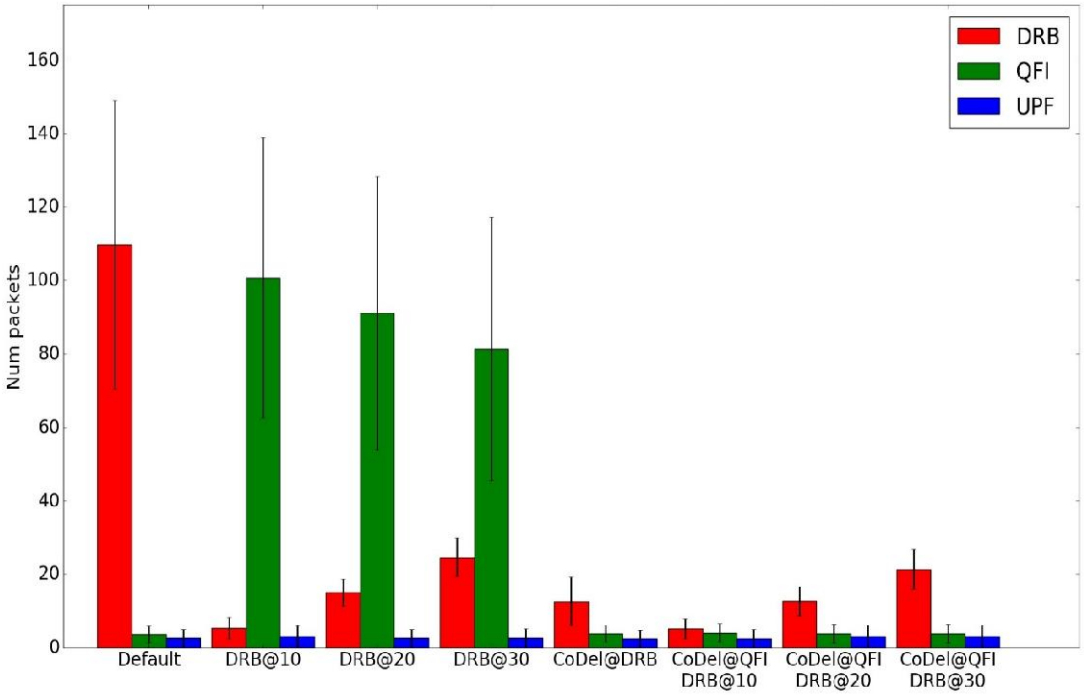

Figure 27: 1st scenario: Average queue occupancy, ping interval of 10 ms.

5.5 Results and Benchmark

In this section we present the experimental results, where the queue occupancy average and its standard deviation, the ping RTT average and the TCP throughput average are plotted. We run the experiment for 30 seconds for every ping interval. The average of queue occupancy and its standard deviation is given for ping interval of 10 ms, while the average TCP throughput and the average low-latency traffic delay are shown for the ping interval from the range of [10 ms,70 ms] with increments of 10 ms.

The experimental results corresponding to the first scenario can be seen in Figs. 27 – 29. The first case corresponds to the conventional solution of not limiting the buffers. Since the buffers are not limited, the packets are forwarded to the DRB buffer, where they accumulate. There are always enough packets at the DRB to fulfill the maximum egress rate and, therefore, no bandwidth is squandered. However, the delay-sensitive traffic suffers from important delays, since the DRB queue presents a large occupancy when the delay-sensitive traffic packet is enqueued.

The second, third and fourth cases correspond to the solution of only limiting the DRB buffer size. With this aim, the DRB buffer is limited to 10, 20 and 30 packets, respectively. As it can be seen from Fig. 27, the packets accumulate at the QFI queue since the SDAP entity does not forward more packets to the DRB queue once its buffer limit has been reached. However, the total number of packets in the system remains constant in the three cases. The throughput is maintained as well as the delay, as observed from Figs. 28 and 29. The system continues to be congested, and shrinking the DRB queue does not have any effect on the delay or the number of packets in the system.

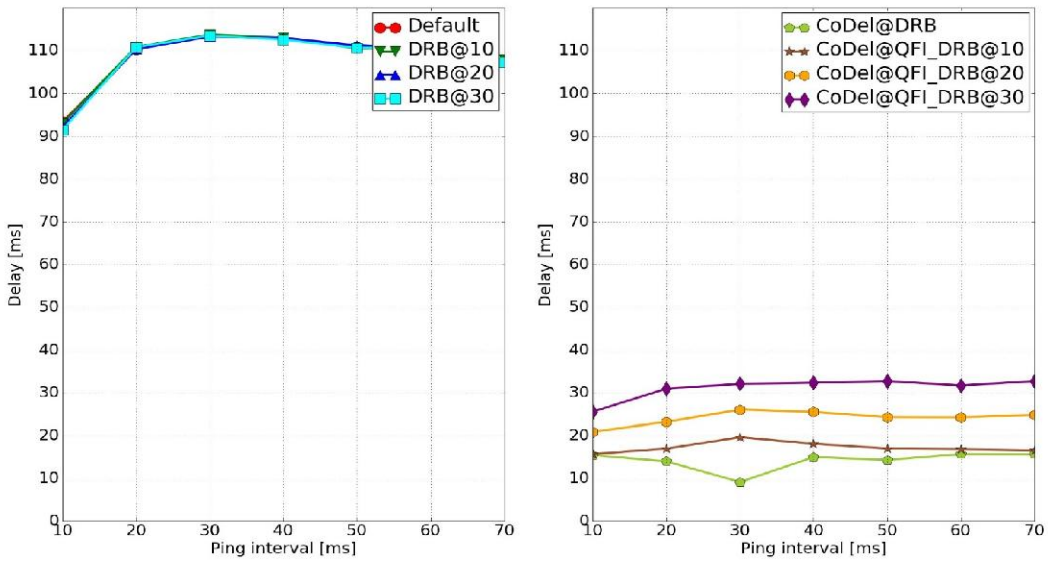

Figure 28: 1st scenario: Average RTT for delay-sensitive flow.

As an alternative solution, in the fifth case, CoDel is implemented at the DRB queue. It shows a clear advantage on the way to reduce the congestion of the system. The total number of packets in the system is significantly reduced as can be observed from the queues' occupancy in Fig. 27. CoDel discards packets if the lowest sojourn time exceeds the target packet delay time in an interval, effectively dropping the TCP transmitting rate, and avoiding the creation of persistent queues. Since the occupancy level of the buffers is low, the delay- sensitive traffic can avoid large sojourn time in queues, and thus, it is delivered faster as observed in Fig. 28. Unfortunately, CoDel also introduces an important variation at the DRB queue occupancy as observed by the standard deviation in Fig. 27, which also translates to the throughput (Fig. 29) and to the delay (Fig. 28) performance. The variation at the DRB queue occupancy leads to TTIs where the DRB queue does not have enough packets to fulfill the maximum egress rate, and therefore, the total TCP throughput is reduced since not all the transmission opportunities are exploited (Fig. 29).

The sixth, seventh and eight cases correspond to our proposed solution of limiting DRB buffer size and using CoDel for the QFI queue. Again, the DRB buffer is limited to 10, 20 and 30 packets, respectively. In this solution, the DRB queue's standard deviation is reduced (Fig. 27) as CoDel acts in the QFI, and therefore, the TTIs where the DRB does not have enough packets to fulfill the maximum egress rate are reduced. Augmenting the size of the DRB buffer, reduces the possibilities of squandering transmit opportunities. However, there exists a limit where augmenting the buffer will not augment the throughput, as all the transmission opportunities are already exploited. From Fig. 29, it can be observed that augmenting the buffer from 20 to 30, does not lead to a throughput growth in the full interval range (Fig. 29).

Moreover, as it can be seen from Fig. 28, incrementing the DRB queue capacity increases the delay. CoDel manages to maintain the buffer occupancy at the QFI queue low, but the RTT augments as the ping packet's sojourn time increases according to DRB's buffer capacity.

One of the effects observed is the TCP throughput rise as the ping interval increases. As there are less delay-sensitive traffic packets in the system, a larger amount of

packets from the TCP flow can be forwarded and, therefore, the throughput increases.

If CoDel is in the congested state after an interval time, it discards the next packet from the queue without distinguishing the packet type. 3GPP states that all the packets that are aggregated to one flow must be treated equally and, therefore, discarding packets with delay-sensitive requirements hap- pens. However, in this scenario, just 0.79% of all the packets emitted are dis- carded by CoDel. From Figs. 28 and 29, it can be extracted that, a queue size limit of 20 packets at DRB in conjunction with CoDel at QFI substantially reduces the delay while keeping the throughput high, leading to an appropriate balance between both metrics.

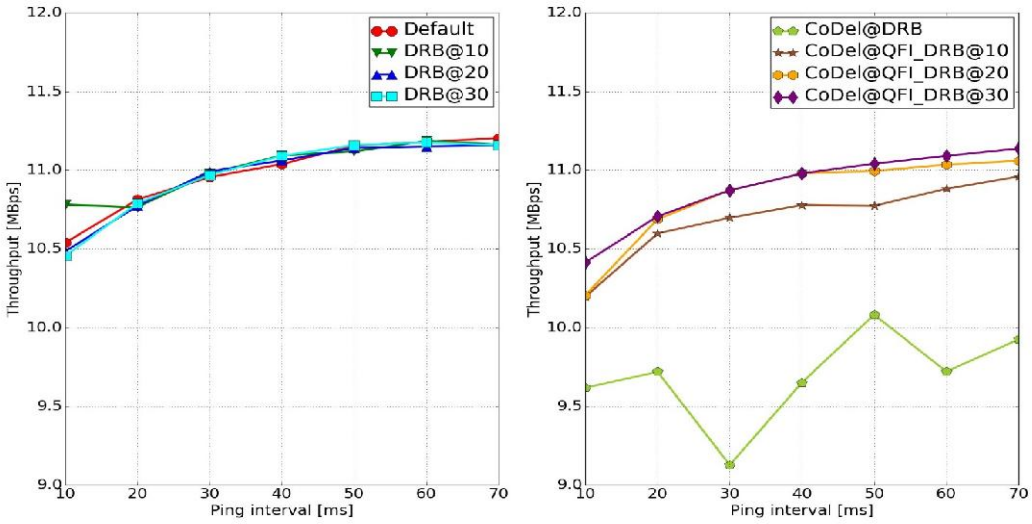

Figure 29: 1st scenario: Average throughput of TCP flow.

The experimental results from the second scenario are shown in Figs. 30, 31 and 32. In the first case, all the packets accumulate at the DRB queue, following the same trend as in the first scenario. No significant reduction in the delay can be obtained from the segregation of the flows in different QFI flows, as observed in Fig. 31, if the DRB queue is not limited. However, the throughput remains fully utilized as observed in Fig. 32.

In the second, third and fourth cases, the queue at the DRB is limited to 10, 20 and 30 packets, respectively. In these cases, the packets corresponding to the delay-sensitive

traffic benefit from the flow segregation and are enqueued into the DRB queue in a Round Robin manner without suffering the delay associated to the TCP flow in the congested QFI buffer. This approach reduces the latency drastically as can be seen from Fig. 31.

Moreover, the latency is directly proportional to the DRB queue size, since the delay-sensitive traffic will suffer bigger sojourn time as the number of packets in the queue increases.

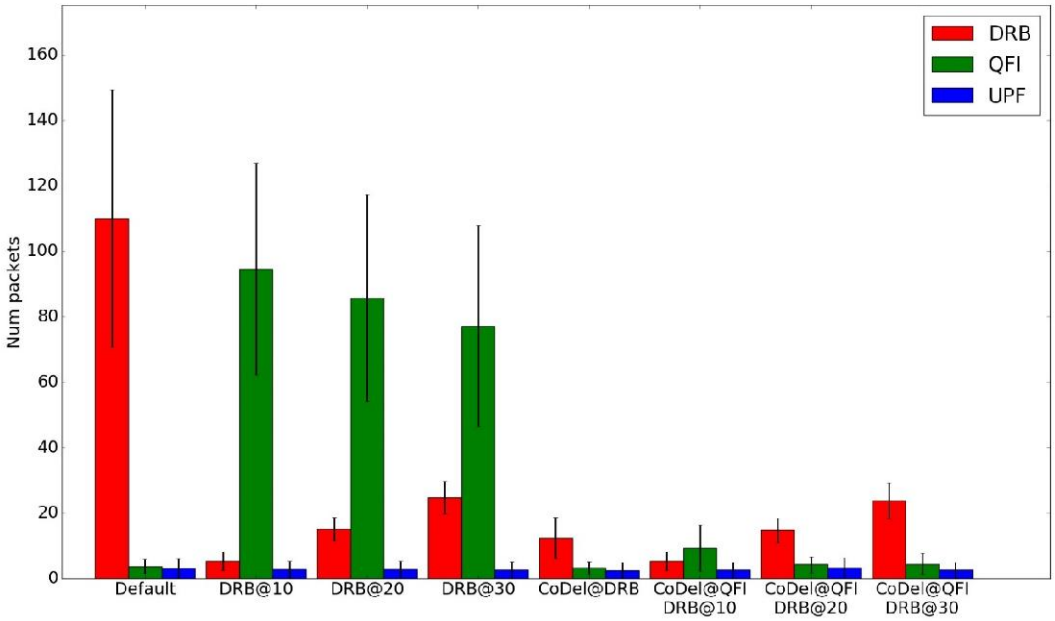

Figure 30: 2nd scenario: Average queue occupancy, ping interval of 10 ms.

This case is comparable to the scenario at [100], where the traffic is segregated in two different flows before being forwarded to the lower layers for prioritization purposes. The throughput is kept high as all the transmission opportunities are used (Fig. 32).

Another solution is shown at the fifth case, where CoDel is implemented at the DRB. The CoDel mechanism maintains the DRB buffer occupancy low as observed in Fig. 30. However, Fig. 32 shows that, in this case, throughput cannot be maximized for the same reasons aforementioned for the same case.

For the last solution and the sixth, seventh and eighth cases, our proposals are evaluated, where CoDel is implemented at both QFI queues, while the DRB queue is limited to 10, 20 and 30 packets, respectively. CoDel successfully maintains the QFI queue occupancy level low, discarding some packets, while all the packets from the delay-sensitive flow are forwarded as they do not exceed the target time. From Fig. 32, it can be observed that a 10 packet queue at DRB decrements the throughput, while the limited queues of 20 and 30 packets are close to the maximum achievable throughput. The delay increases as the DRB queue limit rises as observed in Fig. 31.

Maintaining the bulky and delay-sensitive traffic segregated in different QFIs leads to good TCP throughput and reduced delay as shown in Figs. 31 and 32. However, the number of QFIs per UE and DN are limited, thus, some services will inevitably

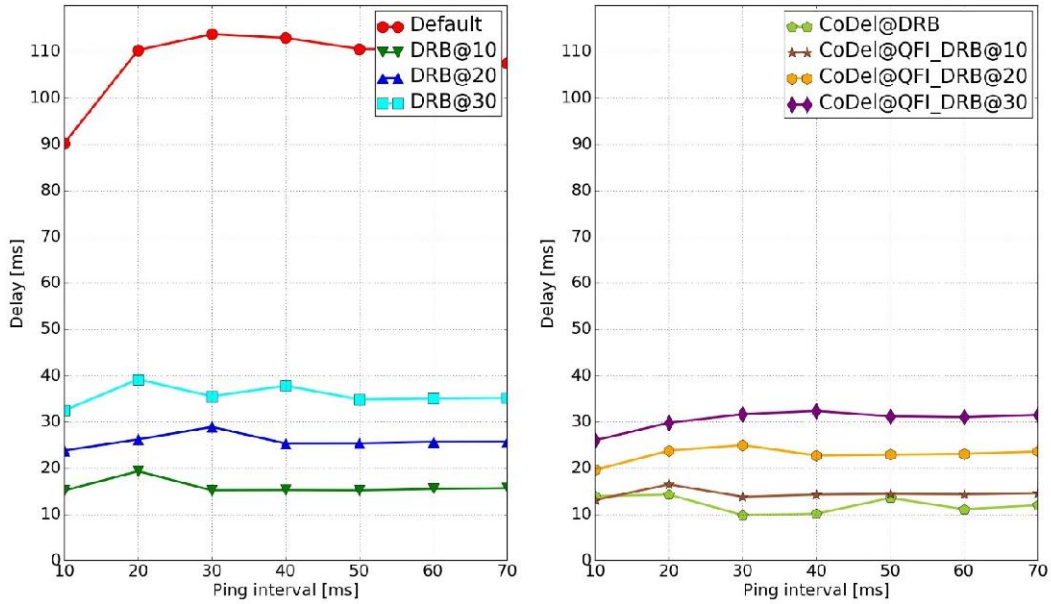

Figure 31: 2nd scenario: Average RTT for delay-sensitive flow.

share QFIs in real deployments. Therefore, the second scenario presented in this section is not scalable. Hence, a good solution for the first scenario is also critical for 5G systems. Moreover, due to 5G's channel capacity variability in the radio access,

determining the optimal limited queue size can be challenging, and overdimensioning the queue will inevitably lead to larger sojourn times than necessary. Hence, an adaptive approach such as the AQM method proposed in this section is needed for 5G. While achieving such dynamic, CoDel has only discarded 0.5% of the delay-sensitive flow packets in the evaluated scenarios. Hence, if deployed at the correct entity, our proposed solution is not detrimental to the throughput, while achieving low delays.

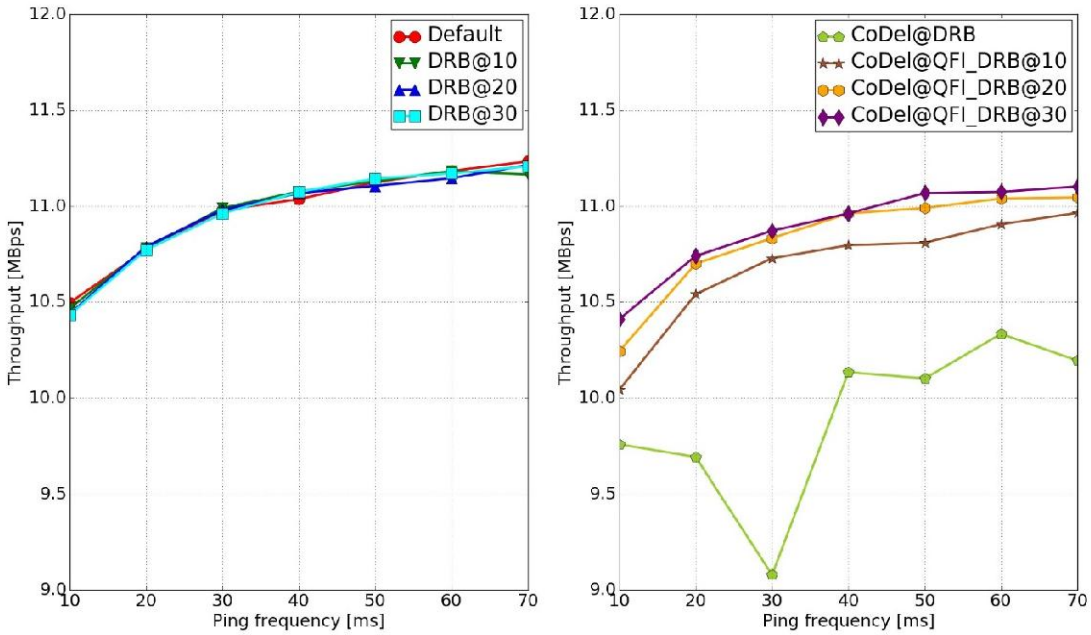

Figure 32: 2nd scenario: Average throughput of TCP flow.

5.6 Conclusion and Future work

Sharing of queues by different services with QoS criteria is an unavoidable phenomenon in 5G networks, for which an exponential increase of traffic is expected. A congested system will be challenging for low-latency services that have to guarantee time constraints. We show the benefits that AQM can bring to the 5G network, exploring the new QoS scenario with the recently included SDAP entity. In this work, non-3GPP compliant solutions have been avoided. We evaluated CoDel

with limited buffer sizes at different layers and entities. Through physical experiments, we show that AQM mechanisms and limited queues can reduce the low-latency traffic delay by a factor of 4 by reducing the queue occupancy, while maintaining the competing TCP flow's throughput close to the achievable maximum.

In any case, the problem itself remains tough due to the dichotomy of trying to achieve two objectives that may seem contradictory. On the one hand, for prioritization purposes, the buffers must remain as empty as possible. On the other hand, in order to achieve high throughput rates, the buffers have to hold enough data not to waste a transmission opportunity.

To understand this divergence objectives and manage them correctly will be crucial to successfully deploy the 5G QoS system.

Some of the related work mentioned in this deliverable [100], has been prove at OpenAirInterface, an open source 5G project. We also plan to integrate our solution at OpenAirInterface [109] in the near future. Moreover, in this work, most QoS parts of the packet delivery network have remained unaware of each other. The TCP congestion control algorithm, the RLC AM, the AQM, the TCP Explicit Congestion Notification (ECN), the Channel Quality Indicator (CQI) or the scheduler are some independent QoS enablers or mechanisms. It looks like a natural approach to aggregate all this information in an entity and orchestrate it according to the priority of the packets and the status of the network. This is also planned to be carefully studied and implemented in the next works.

4. Conclusion

This deliverable has described the enhanced 5G RAN architecture for MTC and proximity services using MEC, C-RAN, NFV, and SDN. Meanwhile, the important roles of QoS enhancement and end user QoE improvement have also discussed.

We have also discussed the MTC services using MEC in the RAN architecture of 5G. Every single node along with all interfaces between devices are thoroughly described. The proposed MTC architecture copes with all types of MTCDs by means of MEC in order to decrease delay, increase throughput, and furthermore to bring computation and communication devices to the edge of the network.

In this deliverable, a new MEC-assisted RAN architectural model for different proximity-based services is proposed. The aim is to enable cloud computing capabilities and IT services in close proximity to end user, by pushing abundant computational and storage resources towards the network edges. Employing MEC between mobile devices and servers brings the possibility of supporting applications with ultra-low latency requirement, prolonging device battery lives and facilitating highly efficient network operations.

A novel congestion control mechanism has been proposed to handle sudden traffic peaks and provide efficient network utilization, by caching latency-tolerant data traffics during congestions. Context information of both traffic characteristics and network conditions are exploit to handle sudden traffic peaks and to efficiently utilize the network capacity. Performance analyses with respect to different specifications are also provided.

Moreover, a comprehensive end-to-end structure of SLA between tenant and service provider of slice based 5G network, which balances the interests of both sides has been proposed, which defines reliability, availability, and performance of delivered telecommunication services in order to ensure a high QoS. We have also discussed business metrics of slice-based network SLA which are critical to the deployment of multi-tenancy services.

5. Reference

[1] M. A. Habibi, B. Han and H. D. Schotten, "Network Slicing in 5G Mobile Communication: Architecture, Profit Modeling, and Challenges," in *Proceedings of the 14th International Symposium on Wireless Communication Systems*, Bologna, 2017.

[2] ITU, "E.860: Framework of a Service Level Agreement," 29 June 2002. [Online]. Available: https://www.itu.int/rec/T REC E.860- 200206-I/en. [Accessed 09 January 2018].

[3] ETSI, "ETSI EG 202 V1.1.1, Quality of Telecom Services, Part 3: Template for Service Level Agreements (SLA)," February 2002. [Online]. Available: http://www.etsi.org/deliver/etsi eg/202000 202099/20200903/01.01.01 60/eg 2020090.

[4] ETSI, "ETSI EG 202 V1.2.1, Quality of Telecom Services, Part 3: Template for Service Level Agreements (SLA)," November 2006. [Online]. Available: http://www.etsi.org/deliver/etsi eg/202000 202099/20200903/01.02.01 50/eg 2020090.

[5] ETSI, "ETSI EG 202 V1.3.1, Quality of Telecom Services, Part 3: Template for Service Level Agreements (SLA)," November 2015. [Online]. Available: https://cedric.cnam.fr/fichiers/art 3101.pdf. [Accessed 17 January 2017].

[6] E. Bouillet, D. Mitra and K. G. Ramakrishnan, "The Structure and Management of Service Level Agreements in Networks," *IEEE Journal on Selected Areas in Communications,* p. 691–699, May 2002.

[7] ISO, "ISO/IEC 2382-14:1997," 1997. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:-14:ed-2:v1:en. [Accessed 08 January 2018].

[8] Gartner Research, "Outsourcing Incentive and Penalty Best Practices," 2003. [Online]. Available: http://www.bus.umich.edu/KresgePublic/Journals/Gartner/research/119100/119134/119134.pdf. [Accessed 03 February 2018].

[9] D. R. K. Z. S. Z. G. P. a. P. N. J. Kosinski, "Definition and evaluation of penalty functions in sla management framework," in *Fourth International Conference on Networking and Services*, 176–181, 2008.

[10] B. Han, S. Tayade and H. D. Schotten, "Modeling profit of sliced 5G networks for advanced network resource management and slice implementation," in *IEEE Symposium on Computers and Communications (ISCC)*, 576–581, 2017.

[11] Huawei, "5G: A Technology Vision," 2013. [Online]. Available: http://www.huawei.com/ilink/en/download/HW-314849. [Accessed 14 December 2017].

[12] K.-R. Jung, A. Park and S. Lee, "Machine-type-communication (MTC) device grouping algorithm for congestion avoidence of MTC oriented LTE network," *International Conference Security Enriched Urban Computer Smart Grid,* vol. 78, pp. 167-178, 2010.

[13] S. Lien, K. Chen and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Commun Mag.,* pp. 66-74, April 2011.

[14] ETSI, "ETSI TS 102 690 v1.1.1, Machine-to-Machine communications (M2M); Functional architecture," 2011.

[15] D. S. Waston, M. A. Piette, O. Sezgen, N. Motegi and L. Hope, "Machine to machine (M2M) technology in demand responsive commercial buildings," *Access Summer Study Energy Efficiency Buildings,* pp. 1-14, 2004.

[16] M. Chen, J. Wan and F. Li, "Machine-to-Machine Communications: Architectures, Standards and Applications," *KSII Transactions on Internet and Information Systems,* pp. 480-497, 2012.

[17] A. Lo, Y. W. Law, M. Jacobsson and M. Kucharzak, "Enhanced LTE-advanced random-access mechanism for massive machine-to-machine (M2M) communications," in *27th World Wireless Research Forum (WWF) Meeting*, 2011.

[18] 3GPP, "Physical channels and modulation. TS 36.211, 3rd Generation Partnership Project (3GPP)," 2014.

[19] J.-P. Cheng, C.-h. Lee and T.-M. Lin, "Prioritized Random Access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *IEEE GLOBECOM Workshops (GC Wkshps)*, 2011.

[20] H. Tian, L. Xu, Y. Pei, Z. Liu and Y. Yang, "Power rampingschemes for M2M and H2H Co-existing scenario," *Communication China,* vol. 10, pp. 100-113, 2013.

[21] J. Metzner, "On Improving Utilization in ALOHA Networks," *IEEE Transactions on Communication,* vol. 24, no. 4, pp. 447-448, 1976.

[22] 3GPP, "TR 37.868 RAN Improvements for Machine-type Communications," 2011.

[23] A. Larmo, M. Lindstrom, M. Meyer, G. Pelletier, J. Torsner and H. Wiemann, "The LTE Link Layer Design," *IEEE Communication Magazine,* vol. 47, pp. 52-59, 2009.

[24] "Understanding LTE," 02 Feburary 2012. [Online]. Available: http://www.sharetechnote.com/Docs/anritsu_understanding_lte6.pdf . [Accessed 13 January 2019].

[25] J. Zyren and W. McCoy, "Overview of the 3GPP Long Term Evolution Physical Layer," Freescale Semiconductor, Inc., white paper, 2007.

[26] "Evolved Universal Terrestrial Radio Access (E-UTRA): Physical Channels and Modulation," ETSI TS 136 211 V10.0.0 , 2011.

[27] A. Laya, L. Alonso and J. Alonso-Zarate, "Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 1, pp. 1-16, 2014.

[28] F. Amirijoo, F. Gunnarsson and F. Andren, "3GPP LTE Random Access Channel Self-Optimization," *IEEE Transactions on Vehicular Technology,* vol. 63, no. 6, pp. 2784-2793, 2014.

[29] "Evolved universal terrestrial radio access (E-UTRA) medium access control (MAC) protocol specification 3GPP TS 36.321 V10.0.0," 3GPP, 2010.

[30] H. Mashud and K. Mahata, "Sequence Design for Random Access Initial Uplink Synchronization in LTE-like Systems," *IEEE Transactions on Wireless Communications,* vol. 16, no. 1, pp. 503-511, 2017.

[31] X. Yang, A. Fapojuwo and E. Egbogah, "Performance analysis and parameter optimization of random access backoff algorithm in LTE," *IEEE Vehicular Technology Conference,* pp. 1-5, 2012.

[32] "Further Performance Evaluation of EAB Information Update Mechanisms 3GPP TSG RAN WG2 R2-120270," Intel, 2012.

[33] "Evolved Universal Terrestrial Radio Access (E-UTRA): Radio Resource Control (RRC) Protocol Specification ETSI TS 136 331," 2015.

[34] S. Duan, V. Shah-Mansouri and V. W. S. Wong., "Dynamic access class barring for M2M communications in LTE networks," *IEEE Global Communication Conference ,* pp. 4747-4652, 2013.

[35] T. A. Kaouther, S. B. Rajeb and Z. Choukair, "A congestion control approach based on dynamic ACB of differentiated M2M services in 5G/HetNet," *13th International Wireless Communications and Mobile Computing Conference (IWCMC) ,* pp. 1126-1131, 2017.

[36] H. W. e. al, "FASA: Accelerated S-ALOHA using access history for event-driven M2M communications," *IEEE/ACM Transaction on Network,* vol. 21, no. 6, pp. 1904-1917, 2013.

[37] "3GPP TSG-RAN2 R2-102978 Separate backoff scheme for MTC," Proc. 70bis, 2010.

[38] "3GPP TSG-RAN2 R2-104662 MTC Simulation Results with Specific Solutions," Madrid, Spain, 2010.

[39] M. S. Ali, E. Hossain and D. I. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Communication Magazine,* vol. 55, no. 1, pp. 76-83, 2017.

[40] C.-H. Wei, R.-G. Cheng and S.-L. Tsao, "Performance analysis of group paging for machine-type communications in LTE networks," *IEEE Transaction Vehicular Technology,* vol. 62, no. 7, pp. 3371-3382, 2013.

[41] G. Farhadi and A. Ito, "Group-based signaling and access control for cellular machine-to-machine communication," *IEEE 78th Vehicular Technology Conference,* pp. 1-6, 2013.

[42] J.-S. Kim, S. Lee and M. Y. Chung, "Efficient random-access scheme for massive connectivity in 3GPP low-cost machine-type communications," *IEEE Transaction on Vehicular Technology,* vol. 66, no. 7, pp. 6280-6290, 2017.

[43] A. M. AlAdwani, A. Gawanmeh and S. Nicolas., "A demand side management traffic shaping and scheduling algorithm," *Proceding Sixth Asia Modelling Symposuim ,* pp. 205-210, 2012.

[44] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul and A. Ghosh, "LTEM evolution towards 5G massive MTC," *IEEE Globecom Workshops (GC Wkshps),* pp. 1-6, 2017.

[45] N.-K. Pratas, H. Thomsen, C. Stefanovi and P. Popovski, "Code expanded random access for machine-type communications," *IEEE Globecom Workshops,* pp. 1681-1686, 2012.

[46] T. Kim, K. S. Ko and D. K. Sung, "Prioritized random access for machine-to-machine communications in OFDMA based systems," *IEEE Internatinal Conference on Communicactions (ICC),* pp. 2967-2972, 2015.

[47] "Merits of the slotted access methods for MTC, 3GPP: R2-112247," 3GPP TSG RAN WG2, 2011.

[48] "3GPP: R2-105623 Comparison on RAN loading control schemes for MTC 3GPP TSG RAN WG2," Alcatel-Lucent Shanghai Bell, 2010.

[49] C. Luders and R. Haferbeck, "The Performance of the GSM Random Access Procedure," *in IEEE Vehicular Technology Conference,* pp. 1165-1169, 1994.

[50] L. M. Bello, P. D. Mitchell and D. Grace, "Intelligent RACH Access Techniques to Support M2M Traffic in Cellular Networks," *IEEE Transactions on Vehicular Technology ,* vol. 67, pp. 8905-8919, 2018.

[51] J. H. Sarker and S. J. Halme, "The prudence transmission method I (PTM I): a retransmission cut-off method for contention based multiple-access communication systems," *IEEE 47th Vehicular Technology Conference. Technology in Motion,* vol. 1, pp. 397-401, 1997.

[52] R. Li, "Intelligentet al., 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Communication Magazine,* vol. 24, no. 5, pp. 175-183, 2017.

[53] X. Wang, X. Li and V. C. M. Leung, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges," *IEEE Access,* vol. 3, pp. 1379-1391, 2015.

[54] N. K. e. al., "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Communication,* vol. 24, no. 3, pp. 146-153, 2017.

[55] A. Imran, A. Zoha and A. Abu-Dayya, "Challenges in 5G: how to empower SoN with big data for enabling 5G," *IEEE Network,* vol. 28, no. 6, pp. 27-33, 2014.

[56] S. Shree-Krishna and X. Wang, "Towards Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions," *arXiv preprint: 1808 02924,* 2018.

[57] L. M. Bello, P. Mitchell and D. Grace, "Application of Q-learning for RACH access to support M2M traffic over a cellular network," *20th European Wireless Conference,* pp. 1-6, 2014.

[58] J. Moon and Y. Lim, "Access control of MTC devices using reinforcement learning approach," *International Conference on Information Networking,* pp. 641-643, 2017.

[59] M. Hasan, E. Hossain and D. Niyato, "Random access for machineto-machine communication in LTE-advanced networks: issues and approaches," *IEEE Communication Magazine,* vol. 51, no. 6, pp. 86-93, 2013.

[60] A. H. Mohammed, A. S. Khwaja, A. Anpalagan and I. Woungang, "Base station selection in M2M communication using Q-learning algorithm in LTE-A networks," *IEEE Conference in Advanced Information Networking and Applications,* pp. 17-22, 2015.

[61] M. Amirijoo, P. Frenger, F. Gunnarsson, J. Moe and K. Zetterberg, "On Self-Optimization of the Random Access Procedure in 3G Long Term Evolution," *IEEE International Symposium on Integrated Network Management-Workshops,* pp. 177-184, 2009.

[62] G. Araniti, J. Cosmas, A. Iera, A. Molinaro, R. Morabito and A. Orsino, "OpenFlow over Wireless Networks: Performace Analysis," *IEEE International Symosium on Broadband Multimedia System Broadcast,* pp. 1-5, 2014.

[63] Cisco, "Global Mobile Data Traffic Forecast Update 2013 - 2018," Cisco, 2014.

[64] N. Fernando, S. W. Loke and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Compution System,* vol. 29, pp. 84-108, 2013.

[65] H. Qi and A. Gani, "Research on Mobile Cloud Computing: Review, Trend and Perspectives," *2nd International Conference on Digital Information and Communication Technology and its Applications,* pp. 195-202, 2012.

[66] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communication Surveys and Tutorials,* vol. 19, no. 3, pp. 1657-1681, 2017.

[67] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young, "Mobile Edge Computing: A Key Technology Towards 5G," *ETSI White Paper,* no. 11, pp. 1-16, 2015.

[68] M. Y. Madhusanka and A. Gurtov, "Software Defined Mobile Networks," *John Wiley & Sons,* 2015.

[69] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka and T. Turletti, "A Survey of Software-defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communication Survey and Tutorials,* vol. 16, no. 3, pp. 1617-1634, 2014.

[70] K. K. Y. e. al., "Blueprint for Introducing Innovation into Wireless Mobile Networks," *Second ACM SIGCOMM Virtualized Infastructure System Architecture,* pp. 10-25, 2010.

[71] M. H. R. Jany, N. Islam, R. Khondoker and M. A. Habib, "Performance Analysis of OpenFlow based Software Defined Wired and Wireless Network," *International Conference of Computer and Inforamtion Technology (ICCIT),* pp. 1-6, 2017.

[72] P. Sermpezis, L. Vigneri and T. Spyropuulos, "Offloading on the Edge: Analysis and Optimization of Local Data Storage and Offloading in HetNets," no. i, 2015.

[73] F. Mehmeti and T. Spyropoulos, "Performance Modeling, Analysis and Optimization of Delayed Mobile Data Offloading under different Service Disciplines," *IEEE ACM Transaction Network,* vol. 25, no. 1, pp. 550-564, 2017.

[74] F. Mehmeti and T. Spyropoulos, "Performance Analysis of Mobile Data Offloading in Heterogeneous Networks," *IEEE Transaction on Mobile Computing,* vol. 16, no. 2, pp. 482-497, 2017.

[75] F. Mehmeti and T. Spyropoulos, "Is It Worth to be Patient? Analysis and Optimization of Delayed Mobile Data Offloading," *IEEE INFOCOM,* pp. 2364-2372, 2014.

[76] X. Duan, X. Wang and A. M. Akhtar, "Partial Mobile Data Offloading with Load Balancing in Heterogeneous Cellular Networks Using Software-Defined Networking," *IEEE Annual International Symposium on Personal Indoor, Mobile Radio Communication,* pp. 1348-1353, 2014.

[77] E. e. a. Zeydan, "Big Data Cashing for Networking: Moving from Cloud to Edge," *IEEE Communication Magazine ,* vol. 54, no. 9, pp. 36-42, 2016.

[78] E. Bastug, M. Bennis and M. Debbah, "Living on the Edge: The Role of Proactive Cashing in 5G Wireless Networks," *IEEE Communication Magazine ,* vol. 52, no. 8, pp. 82-89, 2014.

[79] M. Guan, B. Bai, L. Wang, S. Jin and Z. Han, "Joint Optimization for Computation Offloading and Resource Alloation in Internet of Things," *IEEE Vehicular Technology Conference,* pp. 1-5, 2017.

[80] M. Liu and Y. Liu, "Price-Based Distributed Offloading for Mobile-Edge Computing with Computation Capacity Constraints," *IEEE Wireless Communication Letter,* vol. 2337, no. c, pp. 1-4, 2017.

[81] N. Golrezaei, A. F. Molisch, A. G. Dimakis and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communication Magazine ,* vol. 51, no. 4, pp. 142-149, 2013.

[82] S. Ha, S. Sen, C. Joe-Wong, Y. Im and M. Chiang, "Tube: Time-dependent Pricing for Mobile Data," *ACMSIGCOMM Conference on Applied Technology Architecutre Protocol Computation and Communication,* 2017.

[83] P. Sermpezis and T. Spyropoulos, "Offloading on the edge: Performance and cost analysis of local data storage and offloading in HetNets," *Annual Conference on Wireless On-demand Network System and Service ,* no. i, pp. 49-56, 2017.

[84] P. Si, Y. He, H. Yao, R. Yang and Y. Zhang, "DaVe: Offloading Delay-Tolerant Data Traffic to Connected Vehicle Networks," *IEEE Transaction on Vehicular Technology ,* vol. 65, no. 6, pp. 3941-3953, 2016.

[85] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," pp. 244-251, 2005.

[86] 3GPP, "System architecture for the 5G System (5GS)," Technical Report (TR) 23.501, 3rd Generation Partnership Project (3GPP), 2018.

[87] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Service Data Adaptation Protocol (SDAP) specification," Technical Report (TR) 37. 324, 3rd Generation Partnership Project (3GPP), 2018.

[88] "Bufferbloat," 6 September 2018. [Online]. Available: https://www.bufferbloat.net/projects/. [Accessed 26 January 2019].

[89] E. Blanton, V. Paxson and M. Allman, " TCP Congestion," RFC 5681, 2009.

[90] "CUBIC TCP," 12 January 2018. [Online]. Available: https://en.wikipedia.org/wiki/CUBIC_TCP. [Accessed 26 January 2019].

[91] M. Hock, R. Bless and M. Zitterbart, "Experimental evaluation of BBR congestion control," *IEEE 25th International Conference on Network Protocols (ICNP),* pp. 1-10, October 2017.

[92] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," IEEE Std 1588 - 2008, 2008.

[93] F. S. and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking,* vol. 1, no. 4, pp. 397-413, August 1993.

[94] W. C. Feng, K. G. Shin, D. D. Kandlur and D. Saha, "The Blue Active Queue Management Algorithms," *IEEE/ACM Transactions on Networking,* vol. 10, no. 4, pp. 513-528, August 2002.

[95] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys and E. Dumazet, "The Flow Queue Code Packet Scheduler and Active Queue Management Algorithm," *RFC 8290,* January 2018.

[96] P. E. McKenney, "Stochastic fairness queueing," *In Proc. of IEEE Int. Conf. on Computer Communications,* vol. 2, pp. 733-740, June 1990.

[97] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round Robin," *In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication,* pp. 231-242, 1995.

[98] M. MacGregor and W. Shi, "Deficits for Bursty Latency-critical Flows: Drr++," vol. 02, pp. 287-293, 2000.

[99] M. Ihlar, A. Nazari and R. Skog, "Low Latency, High Flexibility - Virtual Aqm," 2018.

[100] R. Kumar, A. Francini, S. Panwar and S. Sharma, "Dynamic control of RLC buffer size for latency minimization in mobile RAN," *IEEE Wireless Communications and Networking Conference (WCNC),* pp. 1-6, April 2018.

[101] 3GPP, "NR, Radio Link Control (RLC) Specification," Technical Specification (TS) 38.322, 3rd Generation Partnership Project (3GPP), 2019.

[102] A. K. Paul, H. Kawakami, A. Tachibana and T. Hasegawa, "An aqm based congestion control for enb rlc in 4g/lte network," *In IEEE Canadian Conference on Electrical and Computer Engineering (CCECE),* pp. 1-5, May 2016.

[103] K. Nichols and V. Jacobson, "Controlling Queue Delay," *Queue,* vol. 10, no. 5, May 2012.

[104] T. Høiland-Jørgensen, M. Kazior, D. Taht, P. Hurtig and A. Brunstrom, "Ending the anomaly: Achieving low latency and airtime fairness in WiFi," *USENIX Annual Technical Conference,* pp. 139-151, 2017.

[105] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan and S. Panwar, "TCP Dynamics over mmWave Links," *In IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC),* pp. 1-6, July 2017.

[106] "Byte Queue Limits," 22 November 2011. [Online]. Available: https://lwn.net/Articles/469652/. [Accessed 26 January 2019].

[107] X. Che and B. Ip, "Packet-level traffic analysis of online games from the genre characteristics perspective," *Journal of Network and Computer Applications,* vol. 35, no. 1, pp. 240-252, January 2012.

[108] J. D. Beshay, A. T. Nasrabadi, R. Prakash and A. Francini, "On active queue management in cellular networks," *IEEE Conference on Com- puter Communications Workshops (INFOCOM WKSHPS),,* pp. 384-389, May 2017.

[109] "OpenAirInterface," 18 March 2018. [Online]. Available: https://www.openairinterface.org/. [Accessed 19 January 2019].

6. List of Acronyms

Abbreviation	Definition
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
ACB	Access Class Barring
API	Application Programming Interface
AQM	Active Queue Management
CAPEX	Capital Expenditure
CCCH	Common Control Channel
CCE	Congestion Control Engine
CDN	Content Delivery network
CN	Core Network
CP	Content Providers
C-MTC	Critical Machine Type Communication
CP	Cyclic Prefix
D2D	Device to Device
DPI	Deep Packet Inspection
DRB	Data Radio Bearer
DRR	Deficit Round Robin
DT	Delay Tolerant
eMBB	enhanced Mobile Broadband
ETSI	European Telecommunication Standards Institute
GFBR	Guaranteed Flow Bit Rate
GPRS	General Packet for Radio Service
GSM	Global System for Mobile Communiation
H2H	Human-to-Human
HD	High Definition
IaaS	Infrastructure as a Service
IMS	IP Multimedia Core Network Subsystem
IoT	Internet of Things
ISO	International Standardization Organization
ITU	International Telecommunication Union
KPI	Key Performance Indicator
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
M2M	Machine-to-Machine
MCC	Mobile Cloud Computing
MDBD	Maximum Data Burst Delay
MEC	Multi-Access Edge Computing
MFGR	Maximum Flow Guaranteed Rate
NF	Network Function
NFV	Network Function Virtualization

NFVI	Network Function Virtualization Infrustructure
NFVO	Network Function Virtualization Orchestrator
NGMN	Next Generation of Mobile Network
NIC	Network Interface Controller
NMO	Network Management Orchestration
OFDM	Orthogonal Frequency Division Multiplexing
OPEX	Operational Expenditure
PDCP	Packet Data Convergence Protocol
PDR	Packet Detection Rules
PDSCH	Physical Downlink Shared Channel
PER	Packet Error Rate
PRACH	Physical Random Access Channel
PUSCH	Physical Uplink Shared Channel
QFI	QoS Flow Indicators
QoE	Quality of Experience
QoS	Quality of Service
RA	Random Access
RACH	Random Access Channel
RAR	RA Response
RAN	Radio Access Network
RED	Random Early Detection
RFID	Radio Frequency Identification
RNIS	Radio Network Information Service
RRA	Random Access Response
RTT	Round Trip Time
SDAP	Service Data Adaptation Protocol
SDN	Software Defined Networking
SFN	System Frame Number
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TOF	Traffic Offloading Function
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
UHD	Ultra High Definition
UL-SCH	Uplink Shared Channel
URLLC	Ultra Reliable Low Latency Communication
VNF	Virtual Network Function
V/AR	Virtual/Augmented Reality
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network