UNIVERSITY OF MACEDONIA

SCHOOL OF INFORMATION SCIENCES

DEPARTMENT OF APPLIED INFORMATICS

Real-time Detection and Optimization Algorithms for Flexible Resource Allocation in 5G Networks and Beyond

Ph.D. Dissertation

of

Sotiris Skaperas

Thessaloniki, September 2020

Abstract

The roll-out of fifth-generation (5G) mobile networks and the forthcoming sixth-generation (6G) will bring about fundamental changes in the way we communicate, access services and entertainment. With respect to the latter, the multi-fold increase in the service data rates of enhanced mobile broadband (eMBB) services will provide users with ultra high resolution in video-streaming, multi-media and virtual reality, offering immersive experiences.

To this end, it is important for Edge content delivery infrastructures to rapidly detect and respond to changes in content popularity dynamics. For flexible and highly adaptive solutions, the capability for quick resource (re-) allocation should be driven by early and low-complexity content popularity detection schemes. In the present thesis, we study aspects of low-complexity detection of changes in video content popularity in real-time, addressed as a statistical change point (CP) detection problem, breaking completely new ground compared to earlier works that relied upon web content research topic.

Furthermore, novel exciting use cases were introduced in 5G in the context of ultra-reliable low latency communications (URLLC) and massive machine type communications (mMTC); the new industrial revolution, dubbed as Industry 4.0, along with emerging verticals in telemedicine, smart agriculture, etc., will bring about automation and intelligence to levels never seen before. As 5G is required to support a large variety of services, novel solutions to enable higher resource efficiency are sought; in this framework, in this thesis we study layer 2 scheduling of heterogeneous services, focusing in the case of URLLC and eMBB co-existence. We propose novel heuristic algorithms and further investigate solutions leveraging non-orthogonal multiple access (NOMA), because of its advantages over conventional orthogonal multiple access (OMA) schemes in terms of spectral efficiency, cell-edge throughput, and energy efficiency.

All of the proposed solutions are adapted to flexible resource allocation schemes, a cornerstone of 5G and of future 6G networks.

Keywords: Time-series Analysis, Video Content Popularity Detection, Change Point Analysis, Time-series Segmentation, On-line Change Point Detection, Load Balancing, Intrusion Detection, Time-frequency Analysis, Resource Management, Optimization

Acknowledgements

First of all, I would like to express my gratitude to my academic supervisors Prof. Lefteris Mamatas and Prof. Arsenia Chorti for their guidance and support, as, without them, I would not have been able to complete this endeavor. Their endless energy and commitment to hard work motivated me the most.

I would also like to thank my supervisors Prof. George Tsaklidis and Dimitris Kugiumtzis for their support and advices when I most needed them during the challenges I faced.

Most importantly, I am full of gratitude for my parents and my sister, who have always support me in every step of my live.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context

The roll-out of fifth-generation (5G) mobile networks and the forthcoming sixth-generation (6G) will bring about fundamental changes in the way we communicate, access services and entertain. Indeed, 5G will be a key asset impacting multiple sectors, supporting the Industry 4.0 paradigm, ultra high resolution in video-streaming, virtual and augmented reality, autonomous cars, smart city and factory services, remote medical diagnosis and telesurgery, unmanned aerial vehicles, to mention but a few emerging applications, besides the enhanced broadband services focusing on high speed mobility, densely populated areas and ultra-high-definition video. The "one size fits all architecture" currently in use in fourth-generation (4G) is no longer sufficient and 5G is designed to incorporate a diverse set of new services, which are known to have different needs in terms of network performance, such as low latency access, high communication reliability and the support of massive numbers of devices. According to their type, 5G classifies services into [2]:

- Enhanced mobile broadband (eMBB), emphasizing in very high data rates, and large bandwidth and throughput of the network.
- Ultra reliable low latency (URLLC), which provides ultra-responsive and reliable connections for real-time data transmission.
- Machine massive type communications (mMTC), supporting a high density of devices to interconnect billions of sensors and machines.

Consequently, the most innovative feature that 5G brings, is the integration of URLLC and mMTC services with the conventional eMBB services on the 5G new radio (NR) [3]. Especially the ultra reliable URLLC traffic is expected to introduce a broad set of applications, some of which are yet unknown [4].

Besides the above, as the 5G network landscape starts taking shape, new use case requirements arise, building on top of the emerging telecom paradigm. These advantages underpin a holistic network transformation, based also on the emerging network technologies and are expected to enable a societal shift to support a number of new services. Under this consideration, 5G imposes the rethinking of the network architecture at all system levels and the establishment of an overall 5G ecosystem [5] to support NR. This presents the need of an evolution in terms of capacity, performance and spectrum access in radio network segments, and of innate flexibility and programmability conversion in all non-radio 5G network segments. The future 5G and beyond (B5G) ecosystem will have to go well beyond the characteristics that have been identified for the core of the 5G so far, and will be exemplified by the design and implementation of new platforms, whose characteristics and capabilities are expected to surpass those described so far by the main standard developing organizations, e.g., the third generation partnership project (3GPP) [6], the international telecommunication union (ITU) [7] and the European telecommunications standard institute (ETSI) [8]. B5G will support a flexible and on-demand provision of network resources, network functions, and applications, even with short lifecycles considering both physical and virtual resources that stress across multiple administrative domains.

In this direction, another research challenge in 5G and B5G is the efficient synergy of the mobile network edge with nearby cloud deployments in order to achieve the requirements of 5G service classes. Nowadays, more and more services are pushed from the cloud to the edge of the network, as more and more data are produced at the latter [9]. Since it is more efficient to process the data at the edge of the network, this strategy enables shorter response times, better reliability and security. In this context, edge computing is a novel supporting technology to 5G, in the sense of bringing cloud capabilities near to the end users, in order to overcome the intrinsic problems of the traditional cloud, such as high latency and security issues [10]. Edge computing refers to the

technologies that enable edge servers in mini clouds (i.e., edge clouds) to allow computations to be performed at the edge of the network [11, 12]. Therefore, edge computing may be a vital infrastracture for challenging next-generation applications, which are interactive (e.g., augmented and virtual reality) or they assume hard real-time requirements (e.g., high quality video streaming, gaming, and eHealth).

Emerging network architectures, such as network function virtualization (NFV) [13] and software-defined networks (SDN) [14], provide a new path to the edge network, characterized by an increased level of flexibility. Their main goal is to be able to adapt to the dynamic requirements of users or applications and the constraints of resource availability. Such approaches are tightly coupled with cloud environments that provide a pool of resources, available on-demand to the network and service or application environment. However, traditional cloud technologies are typically far away from the user, making this approach unsuitable for real-time or delay-sensitive services. Furthermore, system-level virtual machines (VMs) are quite large and cannot be migrated, booted up or shut down instantly to match the dynamic network behaviour [15].

Lightweight cloud technologies, for instance containers [16] and unikernels [17] appeared to bridge this gap, offering programmable virtual resources that can boot up rapidly and as a consequence they can be manipulated at very low time-scales, at the range of seconds (even milliseconds, in the case of unikernels). Their low resource-demands makes them suitable for deployment near or at the network edge. By enabling these solutions, efficient network optimization strategies can be devised, e.g., a unikernel VM or container may be rapidly migrated or replicated to improve QoS or mitigate network traffic peaks.

The exponential growth of Internet content, in size, quantity, and network traffic demands, enabled new network architectures realizing efficient hosting, discovery and dissemination of content. Content delivery networks (CDNs) [18] are scalable architectures that target efficient Internet content delivery. However, CDNs are usually based on large data centers, typically far away from the content consumers and they are becoming inefficient for the next-generation services envisioned in 5G networks. This problem needs a fresh view, since new networking and cloud paradigms appeared lately (e.g., [19], [20]) addressing challenging issues, such as: (i) scalable and holistic resource

utilization, spanning from large data centers to the user device, including edge clouds; (ii) incorporation of heterogeneous physical and virtual resources; and (iii) adaptability to dynamic user requirements, server resources and network capacity constraints.

From this point of view, the above approaches offer new means to respond quickly to the changes in the content popularity dynamics, with the appropriate adaptations, e.g., efficient capacity and server resource allocation, load balancing or content caching. Consider, for example, an elastic content distribution platform that serves the Internet content using tiny unikernel VMs that host one or few video each, which appear rapidly in nearby cloud deployments, serve users and then disappear [19, 21].

On the other hand, flexible CDN approaches are not yet totally adept to address the needs of the emerging network infrastructures. There is a strong need for appropriate web content popularity monitoring and modeling, so as to drive such load-balancing strategies, e.g., to detect in real-time a video content that becomes viral and signal the event to the edge cloud for a rapid responce (i.e., reproduce a VM). These mechanisms should be capable of instant, accurate decisions, while consuming very low resources, matching the main characteristics of the tiny VMs technology, including their capability for rapid responses.

An important aspect of CDNs is the delivery of video content. Video content is projected to account for 82% of the global Internet traffic by 2020, significantly increased from 72% in 2016 [22]. To deal with the increased traffic, the network should become more intelligent, having capabilities, spanning from mathematical modeling to machine-learning and artificial intelligence (AI), and to be able to autonomously change its operation according to the users' feedback and experience. In order to ensure fast response, the network edge should incorporate intelligent features enabling real-time management of VMs, on a massive scale and in an "always connected" environment. This topic is studied in different facets of 5G networks [23, 24] and it is accurate to say that the required flexibility, adaptability and programmability can only be envisioned thanks to the use of such techniques [25]. Those techniques should be responsible for the implementation of management and orchestration modules, focusing on the execution of configuration and reconfiguration control

loops [26].

Along these lines, detecting or modeling content popularity dynamics is crucial for next-generation CDNs. For example, novel cache replacement methods that are "popularity-driven" have recently appeared, e.g., the algorithms proposed in [27], based on learning the popularity of content and using it to determine which content should be retained and which should be evicted from the cache.

Consequently, in the context of modern, flexible CDN architectures, emerges the need for real-time content popularity detection schemes that are that highly adaptable to short-term dynamics providing the infrastructure with tools for instant responses. Change point (CP) analysis is an ideal statistical tool for content popularity detection; note that it has been used extensively in anomaly detection [28], [29], e.g., for intrusions detection [30–33]. Hence, low-complexity, real-time and autonomous CP algorithms could be applied in flexible edge cloud environments.

In another research front, Internet of things (IoT) enable a wide-range of 5G and beyond network applications, however they face significant security challenges [34], e.g., the amount of IP data handled by wireless networks from under 3 exabytes in 2010 has increased to over 190 exabytes by 2018 and is on pace to exceed 500 exabytes by 2020 [35]. To tackle the relevant challenges, the 5G ecosystem is considered as a key enabler in meeting continuously increasing demands for novel wireless sensor networks (WSNs) and relevant IoT services.

In this context, edge cloud computing is a promising solution, providing resources closer to the IoT devices and enabling a series of emerging technologies, including SDN [36]. This combination is referred to as software-defined wireless sensor networks (SDWSN) and has been presented as an effective approach to address several of these issues in WSN, such as elasticity, processing cost offloading from devices to the edge and resource reuse [37].

With respect to security, SDN and SDWSN networks have shown to be vulnerable to serious threats, which could be grouped in three categories [38]: application plane attacks, control plane attacks, and data plane attacks. Among the three, the control plane attacks are pointed out as the most high impact and attractive to attackers [39], since the control plane is responsible for the overall management of the network [40]. This characteristic turns the control

plane prone to distributed denial of service (DDoS) attacks [41]. Therefore, the study of IoT security and of intrusion and DDoS detection in SDWSNs is an important topic in 5G research.

Furthermore, another key issue in 5G networks is the resource allocation scheduling problem for the efficient coexistence of eMBB with URLLC traffic. The multiplexing of eMBB and URLLC services has been shown to be a hard optimization problem, concerned with the fine-tuning of the trade-off between latency, reliability and spectral efficiency [42]. Hence, the joint scheduling of different 5G NR services in orthogonal multiple access (OMA) environments has attracted considerable attention recently, e.g., [1], [43], aiming at yielding higher throughput for the eMBB users. To this end, intuitively, an efficient strategy should allocate the URLLC services in low throughput resource blocks for the eMBB services locations, as long as URLLC performance is guaranteed. Another promising solution could be the use of the non-orthogonal allocation of the resources (i.e., non-orthogonal multiple access, NOMA) aiming at improving the quality of service (QoS) for the eMBB users, avoiding underutilization of resources due to URLLC (or mMTC) inactivity [43].

To sum up, this thesis addresses the following research topics and proposes novel algorithms, leveraging advanced statistics and optimization theory, that be applied within the scope of efficient resource allocation - or related fields, such as security aspects - in the 5G and B5G context:

- The development of lightweight statistical-based algorithms that rapidly detect changes driving novel flexible edge cloud facilities.

- The applicability of above algorithms on real network scenarios, including next-generation content-provisioning and secure software-defined Internet of things.

- Algorithms targeting the resource allocation optimization problem of joint scheduling URLLC and eMBB services, considering both OMA and NOMA schemes.

1.2 Research Contributions

The main purpose of this thesis is the development and implementation of change point and optimization algorithms for flexible resource allocation in 5G any beyond ecosystems. More precisely, the first part of this thesis is concerned with the development of CP algorithms for real time monitoring of content popularity distribution, within the framework of edge clouds and their real time resource scheduling demands, as well as of traffic anomalies in the IoT. The algorithms have been incorporated in (i) a novel next-generation content delivery infrastructure employing edge clouds with lightweight virtualization; (ii) an intrusion detection facility for software-defined Internet of things environments. In the second part, the 5G NR resource allocation scheduling is studied through formulating the respective optimization problem and providing accurate low-complexity heuristic solutions, targeting both orthogonal and non-orthogonal multiple access.

The main contributions of the thesis are summarized below:

1. The development of an integrated on-line algorithm, denoted as real time change point detector (RCPD), to estimate the existence, the number and the direction of changes on the mean value of video content popularity time series. The RCPD is real-time, lightweight, accurate and is parameterized autonomously by analyzing historical data. The early detection of changes is addressed with a non-parametric CP based methodology, consisting of a training phase, using historical data, and, an on-line phase. In the training phase, a modified off-line CP detection scheme is employed to configure the on-line algorithm's parameters. The off-line scheme is also complemented with a segmentation algorithm used for the detection of multiple CPs. The training phase is shown to greatly improve the accuracy of the on-line detector, as in essence, the algorithm parameterization is not arbitrary but rather extracted from corresponding historical data. Finally, the on-line algorithm is enriched with a modified exponential moving average filter that allows to detect the direction of changes and accordingly respond in terms of load balancing. The suitability of the algorithmic scheme is confirmed against a large number of synthetic as well as real YouTube video data sets. Last but not least, the RCPD is

compatible with modern, flexible networking and cloud approaches, that are highly adaptive and can respond to short-term network dynamics.

2. The extension of the RCPD algorithm for the detection of changes in the variance. A major difference concerned the assumptions for the underlying process, employing in the test statistics both parametric and non-parametric detectors. In the context of parametric models, linear dependence is considered in the form of autoregressive moving average (ARMA), and, nonlinear, in the form of generalized autoregressive conditional heteroskedasticity (GARCH) processes. The integrated algorithm is a combination of off-line and on-line CP schemes, with the off-line scheme used as atraining (learning) phase, similar to the RCPD. The algorithm is assessed with promising results over synthetic and real Youtube video content views time series.

3. An evaluation on the applicability (i.e., in terms of fast detection and computational simplicity) of the above CP schemes i) to drive tiny VMs allocation in elastic CDN platforms, e.g., based on unikernels and ii) to detect DDoS attacks in SDWSW environments. In the first case, such algorithms may signal whenever a significant content popularity change occurs, to trigger the platform to respond by replicating or displacing VMs. In the second case, the algorithms are shown to provide a high attack detection rate with a low computational cost, showcasing their suitability for such resource-constrained networks.

4. The development of novel multi numerology radio resource allocation algorithms to maximize scheduling efficiency for URLLC when coexisting with eMBB services. The novelty in this analysis is the re-formulation of the standard eMBB throughput maximization problem as an equivalent conflict minimization problem, in terms of resource allocation. The former is verified by a greedy heuristic algorithm, improving the performance of traditional approaches [1]. Regarding the running time, a bin packing approach is also considered by jointly minimizing the placements of URLLC services in the time-frequency resource grid and the aggregate conflict, showing with simulation results that the latter performs near

optimally.

5. Moreover, in order to further increase the efficiency of resource utilization, non-orthogonal multiple access (NOMA) is also investigated for URLLC and eMBB coexistence. The superior performance of NOMA, with superposition of services over the same resource blocks, is due to alleviating conflicts, i.e., by allowing for the efficient overlapping of different services in the two dimensional time-frequency grid. The relevant optimization problem is re-formulated and solved using linear programming techniques and solvers. The superiority of NOMA is shown by an extensive set of numerical results. This study indicates the increasing interest in employing NOMA when moving from a one dimensional domain (e.g., time), to two dimensions, i.e., time and frequency. Our conclusions point out that further enhancing of related gains could be expected by including the third dimension of space, i.e., by incorporating multiple input multiple output (MIMO).

1.3 Outline

The thesis is organized as follows. Each Chapter confronts a specific mechanism related to the problem of resource allocation and adheres to the following structure. First, we present a short introduction, the solution motivation and our contributions to to the considered problem, followed by a description of the current state of related work in the literature and a theoretical background, if necessary. Next, the system model is determined and the corresponding problem is formulated. Subsequently, the proposed algorithmic approach is presented along with the respective pseudocode for the practical implementation of the proposed solution. Finally, a series of simulations are provided to validate the operation and the performance gains of each approach. In further detail, the technical Chapters are structured as follows.

In **Chapter 2** a real-time detector of changes in the mean value of content popularity is discussed, referred to as the RCPD. The basic components of the proposed algorithm are described in detail, i.e., off-line and on-line CP procedures, indicators to reveal an existing change and our approach to the

detection of multiple changes. Particular emphasis is given on justifying the compatibility of the general requirements of the adopted methodology with modern, flexible networking and edge cloud approaches.

In **Chapter 3** an extension of the RCPD detection scheme is provided in order to detect also changes in the variance of content views time series. Besides that, the joint incorporation of parametric and non parametric CP detectors is considered. Additionally, the applicability of the RCPD is demonstrated in two different networking problems, i.e., more precisely: i) for resource allocation in an elastic content distribution platform, where VM orchestration utilizes RCPD; and ii) for network security, considering a SDWSN environment and applying RCPD for intrusion detection. In both cases, detailed aspects of the experimental methodology are presented.

In **Chapter 4** the notion of conflict in scheduling of radio resource allocation problems for URLLC when coexisting with eMBB services is introduced. The concept of "conflict" captures the fundamental constraint of OMA schemes that do not allow for superposition of different signals on the same resource block. Based on a novel explicit definition of conflicts heuristic algorithms based on the re-formulation of the standard eMBB throughput maximization problem as an equivalent conflict minimization with URLLC are proposed. Furthermore, in order to further increase the efficiency of resource utilization, NOMA is also investigated for URLLC and eMBB coexistence. The superior performance of NOMA, with superposition of services over the same resource blocks, is due to alleviating conflicts.

Finally, **Chapter 5** concludes the thesis providing a summary of our research results. Moreover, potential future research directions, stemming from this work, are presented.

1.4 Publications

The scientific findings of this thesis have been published in IEEE peer-reviewed journals and international conference proceedings. Below we present the complete list.

Refereed International Journals:

1. S. Skaperas, L. Mamatas and A. Chorti, "Real-Time Video Content Popularity Detection Based on Mean Change Point Analysis", *IEEE Access*, vol.7 pp: 142246-142260, Jul. 2019;

2. S. Skaperas, L. Mamatas and A. Chorti, "Real-Time Algorithms for the Detection of Changes in the Variance of Video Content Popularity", *IEEE Access*, vol. 8, pp: 30445-30457, Feb. 2020;

3. S. Skaperas, N. Fedosian, A. Chorti and L. Mamatas, "Scheduling Optimization of Heterogeneous Services for 5G NR", *IEEE Transactions on Wireless Communications*. **Submitted**.

Refereed International Conference Proceedings:

1. P. Valsamas, S. Skaperas and L. Mamatas, "Elastic Content Distribution Based on Unikernels and Change-Point Analysis", in Proc. *24th Eur. Wireless Conf. (EW)*, Catania, Italy, 2-4 May, 2018, pp. 1-7;

2. S. Skaperas, L. Mamatas and A. Chorti, "Early Video Content Popularity Detection with Change Point Analysis", in Proc. *IEEE Int. Global Commun. (GLOBECOM)*, Abu Dhabi, UAE, 6-11 Dec. 2018;

3. P. Valsamas, S. Skaperas, G. Violettas, T. Theodorou, S. Petridou, D. Vardalis, A. Tsioukas and L. Mamatas, "Experimenting with Cloud and Network Orchestration for Multi-Access Edge Computing", Demo Paper, *IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakech, Morocco, April 2019;

4. G.A. Nunez Segura, S. Skaperas, A. Chorti, L. Mamatas and C. Borges Magri, "Denial of Service Attacks Detection in Software-Defined Wireless Sensor Networks", in Proc. *IEEE Int. Conf. Commun. (ICC) Worskhop on SDN Security*, Dublin UK, 7-11 Jun. 2020;

5. N. Fedosian, S. Skaperas, A. Chorti, L. Mamatas, "Near Optimal Linear Complexity Scheduling of Heterogeneous Services by Resolving Conflicts", *IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2021. **Submitted**.

Awards:

1. We received the Best Demo award for our demostrator on the 5th FED4FIRE+ Engineering Conference, Apr. 19, Copenhagen, Denmark. The scope of the demo was the implementation of a novel Content Distribution Networking (CDN) paradigm deployed over three Fed4FIRE+ test-beds. It utilized lightweight Unikernel-based Virtual Machines and the change point analysis algorithms of this thesis.

Other presentations:

1. S. Skaperas and L. Mamatas, "Change point detection for load balancing based on content popularity," 16th Mathematics of Networks meeting, Sussex, England, 12 Sep. 2017;

2. S. Skaperas, "Change Point analysis for efficient Edge Resource Allocation," ETIS-ICI seminars, CY University, ENSEA, France, 25 Sep. 2018, invited talk.

Peer-reviewed papers outside the scope of the thesis:

1. O. Theodosiadou, S. Skaperas and G. Tsaklidis, "Change Point Detection and Estimation of the Two-Sided Jumps of Asset Returns Using a Modified Kalman Filter", *MDPI, Risks*, vol. 5, pp. 15, 2017.

Chapter 2

Real-time Change Point Detection for Efficient Edge Resource Allocation

2.1 Introduction

Video content is projected to account for 82% of the global Internet traffic by 2020, significantly increased from 72% in 2016 [22]. In parallel, novel emerging networking, cloud and edge computing paradigms with significant elasticity capabilities appeared recently, e.g., software-defined networks (SDNs) [14], Cloud orchestration proposals [44] and content distribution networks (CDNs) [19]. These advances offer the means to respond quickly to changes in content popularity dynamics with appropriate adaptations, e.g., in terms of efficient server resource allocation schemes, load balancing or content caching. As a result, the early detection of changes in content popularity [45], [46] is proving a highly important topic and can have a significant impact on content delivery performance, network traffic and utilization of servers.

In this work, to address the aforementioned issues, we assume a novel and flexible edge cloud environment exploiting lightweight virtualization technologies, such as unikernels [47] or containers, in the context of a video content distribution environment. The core idea in our proposal is that virtual servers (i.e., virtual machines or containers) could host individual video content and

could be "live" for as long as there is corresponding demand; in case of an increase in demand, more replicas of the virtual servers could be deployed, or alternatively shutdown when the demand dies off. We note that bringing up or down a unikernel is typically very fast, with reported numbers for the boot time as little as 20 milliseconds [48].

In this context, due to high volatility in the respective demand, it is important for video content delivery infrastructures to rapidly detect and respond to changes in content popularity dynamics. Here, we employ on-line CP analysis to implement real-time, autonomous and low-complexity video content popularity detection. Our proposal, denoted as *real-time change point detector (RCPD)*, estimates the existence, the number and the direction of changes on the average number of video visits by combining: (i) off-line and on-line CP detection algorithms; (ii) an improved time-series segmentation heuristic for the reliable detection of multiple CPs; and (iii) two algorithms for the identification of the direction of changes. The proposed detector is validated against synthetic data, as well as a large database of real YouTube video visits.

2.2 Contributions and Chapter Organization

So far, the vast majority of research efforts have focused on the *prediction* of content popularity dynamics, as opposed to *real time detection*, which is the focus of this study. There is a multitude of reasons as to why the precision of prediction algorithms may be impaired. Content popularity can be hard to predict, since a variety of factors - both from the digital and the physical world - can influence the users' Internet surfing behavior, e.g., [45]: (i) the quality, type (e.g., commercial or user-provided) and life-time of content; (ii) its relevance to users and physical events; (iii) the social interactions between users; and (iv) the content promotion strategies involved. Importantly, mid-term and long-term content popularity prediction [49] – and corresponding adaptations in the network or cloud environment – can prove highly inaccurate [50] and thus result in inefficient content delivery and utilization of server and network resources.

In this work, to address the aforementioned shortcomings of the commonly

employed prediction algorithms, we propose a corresponding detector, referred to as RCPD. The RCPD is compatible with modern, flexible networking and cloud approaches, including the considered edge cloud environment, i.e., being highly adaptive and responsive to short-term network dynamics. With accurate, on-line content popularity detection, discrepancies between inaccurate predictions and actual changes can be alleviated. The RCPD is real-time, lightweight, accurate and is parameterized autonomously by analyzing historical data.

In the RCPD, we employ the CP detection theory and algorithms. In this contribution, the early detection of changes in the average content popularity is addressed with a novel CP detection methodology, consisting of a training phase, using historical data, and, an on-line phase. In the training phase, we employ a modified off-line CP detection scheme to configure the on-line (sequential) algorithm's parameters. This approach is shown to greatly improve the accuracy of the on-line detector, as in essence, the algorithm parameterization is not arbitrary but rather extracted from corresponding historical data. To the best of our knowledge, this was the first proposal in the literature on combining retrospective (off-line) and sequential (on-line) CP detection schemes in a single algorithm operating autonomously (i.e., without manual configuration of parameters).

Besides that, our approach complements the off-line scheme with an improved time-series segmentation heuristic for the detection of multiple CPs. Furthermore, we propose two possible variations for the on-line CP algorithm, the first based on the standard cumulative sum (CUSUM) procedure [51] and the second on the ratio-type CUSUM procedure [52][1]. Additionally, we introduce two alternative indicators to detect the direction of changes: the first one is directly derived from the statistical test of the on-line CP procedure, while the second is based on a modified exponential moving average filter, extensively used in econometrics. As discussed in Sections 2.4 and 2.5, the RCPD combines all the above mentioned algorithmic elements, and is based on sufficiently general and convenient assumptions. Moreover, unlike other approaches e.g., [53], we employ methods that allow dependence between obser-

[1]The advantage of ratio-type CUSUM is that it does not require the estimation of long-run covariance (variance) matrices, which is the case for the standard CUSUM method.

vations (in the form of $t-$dependence), leading to more realistic assumptions for the statistical structure of the content visits.

We evaluate the proposed detector and its individual algorithmic components (i.e., the off-line / on-line test statistics, the time-series segmentation algorithm and the trend indicator), over synthetic and real YouTube content views data. Our experiments using synthetic data, generated by an autoregressive moving average (ARMA) filter, demonstrate:

- The superior performance of the proposed time-series segmentation heuristic over the standard approach, improving the true alarm rates by up to 43%.
- The ability of the two proposed trend indicators to identify the direction of estimated changes, with successful identification rates exceeding 99%, in all cases.
- The RCPD performance; the true alarm rates surpass 94% for medium / large changes in the mean number of content views, while the corresponding CP identification lag ranges between 10 to 20 instances, confirming the real-time operation of the detector. On the other hand, the RCPD achieves very small false alarm rates, well within the limits of the statistical error specified by the chosen significance level of the CP algorithms.

Furthermore, our tests on real YouTube content views datasets show that:

- YouTube video views match the underlying assumptions of the RCPD, i.e., the content popularity time-series datasets can be modeled as t-dependent.
- RCPD can detect CPs in more than 70% of the videos in our dataset, implying a sufficiently high number of content popularity changes and the suitability of the CP theory framework for content popularity detection.
- The successful CP direction identifications exceed 91%, i.e., the proposed trend indicators work for real data.
- The average dynamic time warping (DTW) distance [54], [55] between the identified CPs and a benchmark off-line algorithm was estimated to be 52 time instances on average, showcasing the rapid responsiveness of the RCPD.

- The overall processing cost of the RCPD is very low; notably, it took less than one second to process 882 videos on a typical personal computer (PC).

2.2.1 Chapter Organization

The rest of the Chapter is organized as follows. In Section 2.3 we provide a comprehensive background and literature review of related topics. In Section 2.4, we present the off-line (training) phase of the RCPD algorithm, while the on-line phase is discussed in Section 2.5. In Section 2.6, we present four experiments over synthetic video content data, providing an extensive validation of RCPD and its main components, Finally, in Section 2.7, we discuss corresponding experiments using a database of real YouTube video views.

2.3 Background and Related Works

2.3.1 Introductory Statistics

This Subsection provides a quick introduction to some basic concepts in statistics and time series analysis. For a more analytical introduction see [56], which inspired this Subsection. A time series is a sequence of observations x_t, each one being recorded at a specific time $t \in \mathbb{R}_+$, on a variable of interest. For example, Fig. 2.1 depicts the time series plot of the daily views of a specific YouTube video content. A discrete time series, used in this thesis is defined for time $t \in \mathbb{N}$.

To allow for the possibly unpredictable nature of future observations, it is natural to suppose that each observation x_t, is a realization of a specific random variables X_t. The time series $\{x_t, t \in \mathbb{N}\}$ is then a realization of the family of random variables $\{x_t, t \in \mathbb{N}\}$. Hence, time series analysis is based on the assumption that the time series is a realization of a stochastic process.

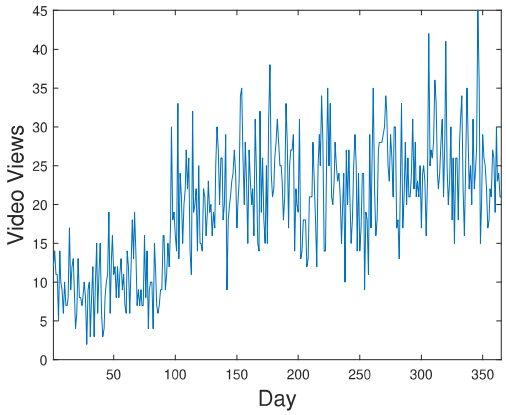


Figure 2.1: Content popularity measurements of a specific YouTube video, i.e., per day.

Stochastic Processes

A *stochastic process* is a family of random variables $\{X_t\}$, where $t \in \mathbb{R}$ or $t \in \mathbb{Z}$ or $t \in \mathbb{N}$. The *mean* μ of a stochastic process $\{X_t\}$ is given by,

$$\mu = E[X_t], \tag{2.1}$$

and the *variance* σ^2 of a stochastic process $\{X_t\}$ is given by,

$$\sigma^2 = Var\left(X_t\right) = E\left[(X_t - \mu)^2\right]. \tag{2.2}$$

The *autocovariance*, which is a metric of the memory of the stochastic process, is given by,

$$\gamma(t, t+h) = Cov(X_t, X_{t+h}) = E\left[(X_t - \mu)(X_{t+h} - \mu)\right], \tag{2.3}$$

and the *autocorrelation* function (ACF) is given by,

$$\rho(t, t+h) = \frac{\gamma(t, t+h)}{\sigma^2} = \frac{E\left[(X_t - \mu)(X_{t+h} - \mu)\right]}{\sigma^2}. \tag{2.4}$$

An important statistical property of stochastic processes, that characterize its behavior in time, is this of *stationarity*. Broadly speaking, a stochastic process $\{X_t\}$ is said to be stationary if it has statistical properties similar to those of the "time shifted" stochastic process $\{X_{t+h}\}$, for each h. Focusing on the

properties that depend only on the first and second-order moments of $\{X_t\}$, a stochastic process is *weak-stationary* (for the rest of the thesis we will refer to weak stationary processes simply as stationary) if and only if:

1. The mean exists, is finite and independent of t. That is $\mu = E[X_t]$.
2. The covariance is independent of t and depends only on the absolute value of the lag τ. That is, $\gamma(\tau) = \gamma(t, t+h)$.

An important statistical property of the stationary stochastic processes is the Wold's decomposition, which proves that every stationary stochastic process X_t can be written as the sum of two time series, one deterministic and one stochastic.

In time series analysis - also in this thesis - there are some well-known baseline stochastic processes that we briefly present below.

Identical and Independently Distributed Random Variables and Random Walk

We say that any $T \in \mathbb{Z}$ random variables $X_1, X_2, \ldots, X_T$ are identical and independently distributed (i.i.d.), if they are identical and independent, i.e., if and only if,

1. $P_1(X_1 \leqslant x) = \cdots = P_T(X_T \leqslant x)$, and,
2. $P(X_1 \leqslant x_1, \ldots, X_T \leqslant x_T) = P_1(X_1 \leqslant x_1) \cdots P_T(X_T \leqslant x_T)$

where P is the joint cumulative distribution function of the T random variables, $x_i \in \mathbb{R}$ and P_i is the cumulative distribution function of the random variable X_i, $i = 1, \ldots, T$. In this model there is no dependence (memory) between observations. In particular, for all $h \geqslant 1$ and all $x, x_1, \ldots, x_T$,

$$P(X_{t+h} \leqslant x | X_1 = x_1, \ldots, X_n = x_n) = P(X_{t+h} \leqslant x). \tag{2.5}$$

Let $Z_t, t \in \mathbb{N}^*$ be an i.i.d. sequence. Then, the sum of Z_t until time T is a stochastic process, referred to as random walk, and is denoted as follows,

$$X_t = \sum_{t=1}^{T} Z_t. \tag{2.6}$$

A random walk is a non stationary stochastic process with mean value $E[X_t] = 0$ and variance $\sigma_X^2 = T\sigma_Z^2$.

White Noise

The sequence $\{X_t\}$ is refered as white noise (WN) if $\{X_t\}$ is a sequence of uncorrelated random variables, each with mean $E[X_t] = \mu$ (or simply $E[X_t] = 0$) and variance $E[X_i X_j] = \delta_{ij}\sigma_X^2$, where δ_{ij} is the Kronecker delta. This is indicated by the notation $\{X_t\} \sim \mathcal{WN}(0, \sigma_X^2)$.

White noise is by definition a stationary process and if the elements of a white noise process follows the Gaussian distribution then the stochastic process is identified as an i.i.d. Gaussian.

Brownian Motion

Brownian motion is the increment process of independent normal variables with zero mean and variance σ^2 (Gausian WN). A standard (one-dimensional) Brownian motion is a stochastic process $\{W_t, t \geqslant 0\}$ with the following properties:

1. $W_0 = 0$, almost surely.
2. W_t is a continuous function in t.
3. The process $\{W_t\}$ has independent increments.
4. The process $\{W_t\}$ has Gaussian increments, $W_{t+s} - W_t \sim \mathcal{N}(0, s)$.

Also the mean value is $E[W_t] = 0$ and $Var(W_t) = 0$. In Fig. 2.2, we illustrate the realization of four independent standard Brownian motions, for $T = 1000$ time instances.

One of the many reasons that Brownian motions are important in probability theory and statistics is that they represent, in a sense, a limit of rescaled simple random walks, or other discrete-time stochastic processes with stationary independent increments.

More precisely, let us assume a sequence $z_i, i \in \mathbb{N}^*$, of i.i.d. random variables with mean 0 and variance 1. For each $n \leqslant 1$, define a continuous–time stochastic

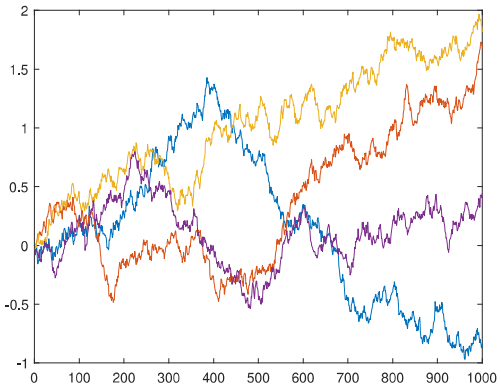


Figure 2.2: Four independent Brownian motions.

process $\{W_n(t), t \in [0,1]\}$ by,

$$W_n(t) = \frac{1}{\sqrt{n}} \sum_{i=1}^{\lfloor nt \rfloor} Z_i \tag{2.7}$$

where $\lfloor . \rfloor$ denotes the integer part. Then the functional central limit theorem (FCLT) asserts that as $n \to \infty$, W_n approaches a standard Brownian motion.

Brownian Bridge

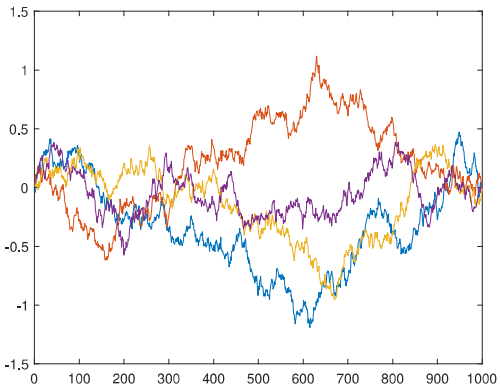


Figure 2.3: Four independent Brownian bridges.

Another important stochastic process is the Brownian bridge which is obtained by taking a standard Brownian motion process $\{W_t\}$, restricted to the interval $t \in [0,1]$, and conditioning on the event that $W_0 = 0$ and $W_1 = 0$.

In other words, if $\{W_t, t \leqslant 0\}$ is a standard Brownian motion then

$$B_t = W_t - tW_1 \tag{2.8}$$

is a standard Brownian bridge in $t \in [0, 1]$. A realization of four Brownian bridges, for $T = 1000$ time instances, is shown in Fig. 2.3.

Brownian Motion and CUSUM Detectors

Recent research efforts have extended the FCLT, in such a way, to provide that for several classes of dependent variables $\{Y_n, n \in \mathbb{Z}\}$, if $\{A_T(x), x \in [0, 1]\}$ is a stochastic process,

$$A_T(x) = \frac{1}{\sqrt{T}} \sum_{n=1}^{\lfloor Tx \rfloor} Y_n \tag{2.9}$$

then,

$$A_T \Longrightarrow \omega W \qquad (T \to \infty) \tag{2.10}$$

where $\Longrightarrow$ denotes weak convergence, $W = \{W_x, x \in [0, 1]\}$ is a standard Brownian motion and ω a functional parameter. For a more exhaustive study see the works [57–59].

Brownian motions and there functionals are proved to be limit distributions of several Cumulative Sum (CUSUM) approaches. Since, the CUSUM statistic is given by,

$$C_T = \max_{1 \leqslant k \leqslant T} \left| \sum_{i=1}^{k} Y_i - \frac{k}{T} \sum_{i=1}^{T} Y_i \right| \tag{2.11}$$

the FCLT, for $x = k/T$, may be used to show that,

$$\frac{1}{\sqrt{T}} \left(\sum_{i=1}^{k} Y_i - \frac{k}{T} \sum_{i=1}^{T} Y_i \right) \Longrightarrow \omega(W_x - xW_1) = \omega B \qquad (T \to \infty) \tag{2.12}$$

where $\omega^2 = \sum_{j=-\infty}^{\infty} Cov(Y_t, Y_{t+j}) = Var(Y_t) + 2\sum_{j=1}^{\infty} Cov(X_t, Y_{t+j})$ denotes the long-run variance and $B = \{B_x, x \in [0, 1]\}$. Finally, following the continuous mapping theorem [57],

$$\frac{C_T}{\omega\sqrt{T}} \xrightarrow{\mathcal{D}} \sup_{0 \leqslant x \leqslant 1} B_x \qquad (T \to \infty) \tag{2.13}$$

where $\xrightarrow{\mathcal{D}}$ indicates convergence in distribution.

The CUSUM processes are in the "epicentre" of the community of change point (CP) analysis, since they are based on a comprehensive mathematical background and are characterized by computational simplicity. Furthermore, they stem from the area of sequential change point detection, a branch of statistics concerned with the design and analysis of the fastest way to detect a change (i.e., an anomaly) in the state of a phenomenon (time process) of interest [60]. A brief literature review on CP analysis follows.

2.3.2 Basis of CP Analysis

Traditionally, CP problems have been phrased as hypothesis tests. The null is set up to describe structural stability of the process and the alternative contains one or multiple CP(s). The test statistics may be viewed as two-sample tests adjusted for the unknown break location, thus leading to max-type procedures. Often asymptotic relationships are derived to obtain critical values for the tests. After the null hypothesis is rejected, the location(s) of the break(s) need(s) to be estimated. This is the setting covered, for example, in [61] and [62].

With respect to CP methodologies that incorporate the serial dependence of the observations into the statistical analysis, two approaches have emerged. The first one aims at quantifying the effect of dependence on the test statistics developed for the independent setting and then to extend their reach to include also the second-order properties as given for example in the autocorrelation function. In this case, the fitting of a particular parametric time series model may be avoided. This appears to be advantageous whenever ambiguity arises at the model fitting stage and model misspecification becomes an issue. This approach then leads to establishing functional central limit theorems for the dependent case and, most crucially, to deriving appropriate estimators for the long-run variances. The second approach utilizes particular time series models and seeks to explicitly describe the dependence structure concurrently with potential structural breaks in the observations. Most popular are the classes of linear ARMA and nonlinear GARCH-type models. Since parametric assumptions are being made, likelihood methods are available and can be used to design relevant test statistics.

Focusing on non-parametric methods [61] CUSUM approaches are non-parametric by design and concurrently may be modified to work also for data exhibiting serial dependence. In general, CUSUM techniques follow the following structure: First we compare the successively estimated cumulated sums of a quantity of interest, with the corresponding quantities estimated from the whole sample; subsequently we reject the null of no-change if the difference exceeds a predefined critical value. This approach is nonparametric in the sense that one does not need to assume a particular distribution and subsequently apply the test statistic directly to the observed data. Moreover, the tests allow for serial dependence such that it is possible to apply the test on, for example, GARCH models.

Principally, weak-sense stationarity is required for applying the fluctuation tests. While this is fulfilled in GARCH models under certain conditions, conditional heteroscedasticity might be a problem for the tests as they might reject the null too often [63]. To circumvent this problem, some kind of pre-filtering on the data should be applied. More precisely, a generic form of these type of tests is a functional (maximum - functional), of the series,

$$TS_t = \frac{t}{T}\widehat{\omega}^{-1}|q_t - q_T|$$

where q_t is the quantity of interest calculated from the first t observations, q_T is the quantity of interest calculated from the first T ($T \geqslant 0$) observations and ω is an estimator (from all T observations) for the asymptotic variance (covariance matrix) under the null. $\widehat{\omega}$ captures serial dependence and fluctuations of higher moments, hence the assumptions for the estimator determines the assumptions for the dependence structure of the observations. Both $\frac{t}{T}$ and $\widehat{\omega}$ serve for normalization; with $\frac{t}{T}$ small, less weight is laid on the differences at the beginning, where the parameters cannot be well estimated. The process TS_t converges against a functional form of a Brownian motion process and thus, in practice we compare the functionals of TS_t with the respective quantiles of this functional.

Considering the literature, the authors in [64] provide a CUSUM stopping rule with application in computer vision problems. A CUSUM approach for CP detection on observations with an unknown distribution before and after a

change, has been recently developed in [65]. In [66] and [67], CUSUM based approaches were introduced for the detection of SYN attacks. Furthermore, an algorithm based on the Shiryaev-Roberts procedure was proposed in [68], to detect anomalies in computer network traffic.

On the other hand, parametric methods utilize as inputs values obtained from a specific model that has been fit to the original data. As an example, Kalman filtering is combined with several CP methods in [69]. In [30], traffic flows are modeled using Markov chains and an anomaly detection mechanism based on the generalized likelihood ratio test (LRT) algorithm. Further examples assuming specific distributions for the data include [32], in which a bivariate sequential generalized ratio test (LRT) was proposed, assuming that the packet rate and the packet size follow a Poisson and a normal distribution, respectively.

Other parametric anomaly detection approaches assume a particular underlying process for the normal behavior and search for anomalies on the residuals of the process. For example, in [69], Kalman filtering is combined with several CP methods, such as CUSUM and LRT, to detect anomalies in origin-destination flows. In [30], traffic flows (in the form of TCP's finite state machine), are modeled using Markov chains and an anomaly detection mechanism based on the generalized LRT algorithm is developed. Furthermore, non residual methods include estimates' detectors based on the differences between the estimated model parameters (see [70], [71]), or based on the quasi-likelihood scores estimators of the parameters of a GARCH process [72].

Besides to retrospective (off-line) methods, sequential (on-line) detection procedures have been developed, where one monitors the output of a process and wishes to signal deviations from the null hypothesis quickly. Starting with the fundamental work of [73] the core of the approach has slowly changed. These authors have developed fluctuation tests that are based on the general paradigm that an initial time period of length n is used to estimate a model with the goal to monitor for parameter changes on-line. To test the null hypothesis of structural stability sequentially, one defines a stopping time τ_n that rejects the null as soon as a suitably constructed detector function TS_n crosses an appropriate threshold cv_n (measuring the growth of the detector under the

null), that is,

$$\tau_n = \inf\{m \geqslant 1 : |TS_n(m) \geqslant cv_n(m)|\}.$$

Sequential tests based on CUSUM-based detector functions were considered in [74] and [75] for linear and time series regressions, in [76] for AR processes, and in [72] for GARCH processes.

The theory of CP analysis is typically pertinent to anomaly detection. In the domain of networking in particular, the theory of CP detection has played an instrumental role in the modelling of network traffic monitoring represented through time series [77] and network anomaly / intrusion detection [28]; for a comprehensive review the interested reader may refer to [78]. In this framework, CP detection techniques [79] are used for the identification of: (i) point anomalies and outliers, i.e., data points deviating distinctively from the bulk of collected data; (ii) pattern anomalies, i.e., groups of data points that are collectively anomalous with respect to historical data; and, (iii) CP anomalies due to changes in the time series's statistical structure (in the mean / variance and in general in the underlying distribution). In this work, we focus on the detection of CP anomalies and consider the other two categories as disturbances. The reasoning behind this choice is that, on one hand, a resource allocation scheduler should be insensitive to instantaneous / very short-term changes in resource demand (e.g., represented as outliers in the content demand), but, on the other hand, should be highly responsive to changes in the underlying statistics of the demand.

To the best of our knowledge, our papers [21, 80–82] are the first in the literature proposing CP techniques for content popularity detection.

2.3.3 Content Popularity Monitoring

The topic of content popularity attracted a lot of attention in recent years, because of its importance in a number of applications, such as network dimensioning (e.g., capacity planning or scaling of resources), on-line marketing (e.g., advertising, recommendation systems) or real-world outcome prediction (e.g., analysis of economical trends) [45]. Especially the prediction of video content popularity characteristics and dynamics [83], as well as models to predict popularity evolution, e.g., [46] and [84], is a well studied topic in the

literature.

The main approaches used for content popularity estimation can be categorized as: (i) cumulative growth studies, estimating the "amount of attention" from the publication instance to the prediction moment [46]; (ii) temporal analysis approaches, i.e., how content visits evolve over time [85]; and (iii) clustering methods of content with similar popularity trends [49]. We note that many content popularity studies consider the aggregate behavior of a particular content, e.g., [46], [85], whereas we study the real-time behavior of video views time-series. Among others, in [86], the authors perform a detailed analysis to characterize the YouTube traffic within a campus network and conclude that in this scenario the content popularity can be well approximated by the Zipf distribution. A comprehensive survey on video traffic models can be found in [87]. Overall, several methods have been proposed in this context, including time series models, regression models [88–90] and machine learning (deep neural networks) techniques [91, 92].

Recently, there has recently been a surge of research in the area of content popularity prediction using artificial intelligence (AI) [93]. In this context, machine learning based methods (e.g., deep learning) need effective feature mining and a huge mass of labeled examples to provide successful performance [94], [95]. In applications in which *real time* content popularity monitoring is required this might become a challenge. As an example, in [96] the authors propose an *off-line* deep learning approach to detect popularity that is subsequently integrated into the on-line caching policy in fog radio applications; however, whenever there is an important change in the underlying dynamics of content popularity, it follows that a new off-line training might be required to run the algorithm properly. We alternatively turn our attention to lightweight *statistical* procedures that fall in the general context of AI (instead of deep learning specifically), in order to operate in an on-line manner (real-time) and to keep the size of the required set of historical data as small as possible. Our proposed algorithm is autonomous, in the sense that all its parameters are determined without manual intervention during a training period; furthermore, the training period is limited to only a few hundred data points (instead of thousands or millions as is typical in deep learning).

As opposed to previous content popularity prediction works, in this Chap-

ter we introduce a novel CP detection methodology that provides accurate, lightweight, autonomous and on-line CP detection of content popularity. We formulate the detection of a change in the average content popularity as a statistical hypothesis test and employ non-parametric procedures to avoid a particular distribution assumption (such as a specific copula model). This context ensures low convergence time since it avoids estimating a large number of model parameters and restrictive assumptions that may not match the structure of the time-series. Furthermore, we avoid problems of parametric models that require parameters' fitting and selection, which become challenging as new data become available. In the proposed RCPD algorithm, an off-line phase specifies important parameters for the on-line phase; these parameters are re-evaluated dynamically after a detected CP.

2.4 Training (Off-line) Phase

In this Subsection, the training phase of the algorithm is discussed and the fundamental components of the off-line scheme are presented. We note that standard off-line CP schemes can only detect a single CP. To address the issue of detection of multiple CPs, we modify the basic algorithm with a novel time-series segmentation heuristic, that belongs to the family of binary segmentation algorithms.

We also note that the statistical characterization of the distribution of visits, have been showed (or assumed) to follow either a Zipf [97] or a Zipf-Mandelbrot [98]) in numerous analyses for both commercial and user-generated content. Even more, time series modeling according to long-range dependent processes has become quite widespread. It is clear that a series with long periods where the observations are away from the mean can also naturally be modeled by a nonstationary process whose mean changes. In [99] authors confirmed that the so-called Hurst effect, which motivated Mandelbrot and his collaborators to advocate the use of self-similar processes, can also be explained if the observations X_k are assumed to follow the model $X_t = Y_t + f(t)$, where Y_t is a weak dependent stationary process and $f(t)$ is a deterministic function. In the context of network traffic, similar findings are reported in [100] and show that periodicity can obscure the analysis of a signal giving false evidence of

long-range dependence.

Hence, we conclude that assuming a weak dependent time series with CP in the mean for the description of content views is in agreement with the general statistical properties of content popularity problems. Moreover, with respect to the fact that estimating long-range dependence is not straightforward as there is no systematic or definitive methodology [100]. To address this issue, we evaluate the assumption of short-range dependence in the real YouTube datasets used in the present work through the evaluation of the time series' Hurst exponents, as will be discussed in Section 2.7.1.

2.4.1 Basic Off-line Approach

Let $\{X_n : n \in \mathbb{N}\}$ be a sequence of r- dimensional random vectors (r.v.). The first dimension represents the number of views for a specific video content within a time period $n \in \{1, \ldots, N\}$, while the other dimensions could be optionally used to represent other content popularity features, such as likes, comments, etc. We assume that $X_1, ..., X_N$ can be written as,

$$X_n = \mu_n + Y_n, \quad 1 \leqslant n \leqslant N \tag{2.14}$$

where $\{\mu_n : n \in \mathbb{N}\}$ is the mean value of video visits, $\{Y_n : n \in \mathbb{N}\}$ a random component with zero mean $\mathbb{E}[Y_n] = 0$ and positive definite covariance matrix, $\mathbb{E}\left[Y_n Y_n^T\right] = \Sigma$, while $\mathbb{E}[\cdot]$ denotes expectation. We further assume that the time-series is t-dependent, implying that for $t_1, t_2, t \in \mathbb{N}$, Y_{t_1} is independent of Y_{t_2} if $|t_1 - t_2| > t$.

The off-line analysis tests the constancy (or not) of the mean values up to the current time N. Hence, we define the following null hypothesis of constant mean,

$$H_0 : \quad \mu_1 = \ldots = \mu_N,$$

against the alternative,

$$H_1 : \quad \mu_1 = \ldots = \mu_{k^*_{off}} \neq \mu_{k^*_{off}+1} = \ldots = \mu_N,$$

indicating that the mean value changed at the unknown (time) point $k^*_{off} \in$

$\{1, \ldots, N\}$.

Considering (2.14) and the corresponding assumptions for the stochastic process X_n, we develop a non-parametric CUSUM test statistic following [61]. The test statistic TS_{off}, can be viewed as a max-type procedure,

$$TS_{off} = \max_{1 \leqslant n \leqslant N} C_n^T \widehat{\Omega}_N^{-1} C_n, \tag{2.15}$$

where the parameter C_n is the retrospective CUSUM detector,

$$C_n = \frac{1}{\sqrt{N}} \left(\sum_{i=1}^{n} X_i - n\overline{X}_{1,N} \right), \tag{2.16}$$

while $\overline{X}_{1,N} = \frac{1}{N} \sum_{i=1}^{N} X_i$ denotes the sample mean. $\widehat{\Omega}_N$ represents a suitable estimator of the long-run covariance Ω, where

$$\Omega = \sum_{i=-\infty}^{\infty} \mathrm{Cov}\left(X_n X_{n-i}\right). \tag{2.17}$$

The estimator should satisfy,

$$\widehat{\Omega}_N \xrightarrow{P} \Omega \tag{2.18}$$

where $\xrightarrow{P}$ denotes convergence in probability.

Several estimators have been proposed in the literature that satisfy (2.18), including kernel-based [101], bootstrap-based [102], etc. Considering our requirement for real-time detection (low computational time), a kernel-based estimator is more suitable; in this context, we employ the Bartlett estimator, so that

$$\widehat{\Omega}_N = \widehat{\Sigma}_0 + \sum_{w=1}^{W} k_{BT}\left(\frac{w}{W+1}\right)\left(\widehat{\Sigma}_w + \widehat{\Sigma}_w^T\right), \tag{2.19}$$

which satisfies (2.18), while the function $k_{BT}(.)$ corresponds to the Bartlett weight,

$$k_{BT}\left(x\right) = \begin{cases} 1 - |x|, & \text{for } |x| \leqslant 1 \\ 0, & \text{otherwise} \end{cases}, \tag{2.20}$$

and $\widehat{\Sigma}_w$ denotes the empirical auto-covariance matrix for lag w,

$$\widehat{\Sigma}_w = \frac{1}{N} \sum_{n=w+1}^{N} \left(X_n - \overline{X}\right) \left(X_{n-w} - \overline{X}\right)^T. \tag{2.21}$$

Finally, we chose $W = \log_{10}(N)$ as in [101].

The long-run covariance is involved in the test statistic to incorporate the dependence structure of the r.v. into the statistical analysis, through the integration of second order statistical properties. This approach is suitable for the targeted context since we avoid a restrictive assumption for the dependence structure of the observations.

Going back to the basic question of rejecting or not H_0, we need to obtain critical values, denoted by cv_{off}, for the test statistic. We approach this issue by considering the asymptotic distribution of the test statistic under H_0,

$$TS_{off} \xrightarrow{D} cv_{off} = \sup_{0 \leqslant t \leqslant 1} \sum_{j=1}^{r} B_j^2(t) \quad (N \to \infty), \tag{2.22}$$

where $\xrightarrow{D}$ denotes convergence in distribution, $(B_j(t) : t \in [0,1])$, $1 \leqslant j \leqslant r$, are independent standard Brownian bridges $B(t) = W(t) - tW(1)$, and $W(t)$ denotes the standard Brownian motion with mean 0 and variance t. The critical values for several significance levels α can be computed using Monte Carlo simulations that approximate the paths of the Brownian bridge on a fine grid. The last step is to estimate the unknown CP, defined previously as k^*_{off}, under H_1, given by:

$$\hat{k}^*_{off} = \frac{1}{N} \underset{1 \leqslant n \leqslant N}{\operatorname{argmax}} TS_{off}. \tag{2.23}$$

2.4.2 Extended Off-line Approach

The above hypothesis test identifies the existence of at most one CP and does not ensure that the sample remains statistically stationary in either direction of the detection. In particular, by construction (see (2.15)), the off-line test statistic detects the CP with the highest magnitude. Therefore, for the detection

of multiple CPs we need to rephrase the hypothesis test H_1, as follows:

$$H_1: \quad \mu_1 = \ldots = \mu_{k_1} \neq \mu_{k_1+1} = \ldots = \mu_{k_2} \neq \ldots$$
$$\cdots \neq \mu_{k_{\tau-1}+1} = \ldots = \mu_{k_\tau} \neq \mu_{k_\tau+1} = \ldots = \mu_N.$$

A greedy technique to identify multiple CPs is the binary segmentation (BS) algorithm. The standard BS algorithm relies on the general concept of binary segmentation and is an extension of the single CP estimator. First, a single CP is searched for in the time-series. In case of no change, the procedure stops and H_0 is accepted. Otherwise, the detected CP is used to divide the time-series into two segments in which new searches are performed. The procedure is iterated until no more CPs are detected. The BS algorithm is lightweight (computational time $O(N\log N)$), while its conceptual simplicity leads to efficient implementations. On the other hand, it has been shown in the literature [103], [104], that the standard BS algorithm tends to overestimate the number of CPs, as it does not cross-validate them after their detection.

In the extended off-line approach, we propose the modification of the standard BS with a cross-validation step of the estimated CPs. The cross-validation step is similar to that used in the iterative cumulative sum of squares (ICSS) segmentation algorithm [105], which is used to search for CPs on the marginal variance of i.i.d. r.vs. In the extended off-line algorithm we consider the CPs estimated from the standard BS in pairs and check if H_0 is rejected in the segment delimited by each pair. If H_0 is not rejected in a particular segment, then no change can be detected in it; as a result, all CPs that fall in the respective segment are eliminated. The improvement, in terms of accuracy, is shown through simulation results in Section 2.6. The pseudo-code of the modified BS algorithm is given in *Algorithm 1*; note that we integrate the algorithm with the test statistic TS_{off}, given in equation (2.15) and the corresponding critical value (cv_{off}) given in (2.22).

Algorithm 1 Modified Binary Segmentation (MBS)

procedure MBS(start,end,A)
    ; A: BS method selection (0: standard, 1: modified)
    ; TS_{off}: the off-line test statistic (eq. 2.15)
    ; cv_{off}: the critical value (eq. 2.22)
    ; $\hat{k}^*_{off}$: the identified CP (eq. 2.23)
    calculate $TS_{off}(start, end)$ and cv_{off}
    if $TS_{off}(start, end) > cv_{off}$ **then**
        calculate $\hat{k}^*_{off}$ and store it in array s
        MBS(start,$\hat{k}^*_{off}$,0)
        MBS($\hat{k}^*_{off}$+1,end,0)
    end if
    if `array_length`(s) > 0 and A=1 **then**
        $\widehat{S} \leftarrow \{1\} \cup \{s\} \cup \{N\}$; N: the time-series length
        for i=2:N-1 **do**
            MBS($\widehat{S}_{i-1}, \widehat{S}_{i+1}$,0)
            keep in l the validated CPs only
        end for
    end if
end procedure

2.5 On-line Phase

In this Subsection, we describe the on-line scheme that includes: (i) two alternative CUSUM-type approaches for the detection of a change in the mean; and (ii) two alternative approaches to estimate the direction of a change.

2.5.1 On-line Analysis

We rewrite equation (2.14) in the form,

$$X_n = \begin{cases} \mu + Y_n, & n = 1, \ldots, m + k^* - 1 \\ \mu + Y_n + I, & n = m + k^*, \ldots \end{cases} \tag{2.24}$$

where μ, $I \in \mathbb{R}^r$ represents the mean value vector before and after the unknown time of possible change $k^* \in \mathbb{N}^*$ respectively. As a reminder, the first dimension of the time-series represents the video views; the rest could be likes, comments, etc., and $\{Y_n : n \in \mathbb{N}\}$ is a random component. The term $m \in \mathbb{N}$ denotes the length of the training period, i.e., an interval of length m over the historical

period during which the mean is assumed to remain unchanged, so that,

$$\mu_1 = \cdots = \mu_m. \tag{2.25}$$

To satisfy this assumption, the modified off-line CP test previously presented is run in order to identify a suitable m. With m determined, the on-line procedure can be used to check whether (2.25) holds as new data become available. In the form of a statistical hypothesis test, the on-line problem becomes,

$$\begin{aligned} H_0 &: I = 0, \\ H_1 &: I \neq 0. \end{aligned} \tag{2.26}$$

The on-line sequential analysis belongs to the category of stopping time stochastic processes. In general, a chosen on-line test statistic $TS_{on}(m, l)$ and a given threshold $F(m, l)$ define the stopping time $\tau(m)$:

$$\tau(m) = \begin{cases} \min\{l \in \mathbb{N} : TS_{on}(m, l) \geqslant F(m, l)\}, \\ \infty, \text{ if } TS_{on}(m, l) < F(m, l) \ \forall l \in \mathbb{N}, \end{cases} \tag{2.27}$$

implying that $TS_{on}(m, l)$ is calculated on-line for every l in the monitoring period. The procedure stops if the test statistic exceeds the value of the threshold function $F(m, l)$. As soon as this happens, the null hypothesis is rejected and a CP is detected. The following properties should hold for $\tau(m)$,

$$\lim_{m \to \infty} Pr\{\tau(m) < \infty | H_0\} = \alpha,$$

ensuring that the probability of false alarm is asymptotically bounded by $\alpha \in (0, 1)$, and,

$$\lim_{m \to \infty} Pr\{\tau(m) < \infty | H_1\} = 1,$$

ensuring that under H_1 the asymptotic power of the statistical test is unity. The threshold $F(m, l)$ is given by,

$$F(m, l) = cv_{on,a} g(m, l), \tag{2.28}$$

where: (i) the critical value $cv_{on,a}$ is determined from the asymptotic behavior

of the stopping time procedure under H_0 by letting $m \to \infty$; and (ii) the weight function,

$$g(m,l) = \sqrt{m}\left(1+\frac{l}{m}\right)\left(\frac{l}{l+m}\right)^{\gamma} \tag{2.29}$$

depends on the sensitivity parameter $\gamma \in [0, 1/2)$.

We use two different CUSUM approaches; the standard [51], with test statistic denoted by TS^{ct}_{on}, and, the ratio-type [52], with test statistic denoted by TS^{rt}_{on}. Their corresponding critical values are denoted by $cv^{ct}_{on,a}$ and $cv^{rt}_{on,a}$, respectively, and their stopping rules by $\tau_{ct}(m)$ and $\tau_{rt}(m)$, correspondingly. Both tests are based on the sequential CUSUM detector, $E(m,l)$,

$$E(m,l) = \left(\overline{X}_{m+1,m+l} - \overline{X}_{1,m}\right) \tag{2.30}$$

The standard CUSUM test is expressed as:

$$TS^{ct}_{on}(m,l) = l\widehat{\Omega}_m^{-\frac{1}{2}} E(m,l), \tag{2.31}$$

where $\widehat{\Omega}_m$ is the estimated long-run covariance, defined as in (2.17), that captures the dependence between observations. Then, the stopping rule $\tau_{ct}(m)$, is defined as:

$$\tau_{ct}(m) = \min\{l \in \mathbb{N} : \|TS^{ct}_{on}(m,l)\|_1 \geq cv^{ct}_{on,a} g(m,l)\}, \tag{2.32}$$

where the ℓ_1 norm is involved to modify TS^{ct}_{on} so that it can be compared to a one dimensional threshold function. The critical value, $cv^{ct}_{on,a}$, is derived from the asymptotic behavior of the stopping rule under H_0:

$$\begin{aligned}\lim_{m\to\infty} Pr\{\tau(m) < \infty\} &= \lim_{m\to\infty} Pr\left\{\sup_{1\leqslant l\leqslant\infty} \frac{\|TS^{ct}_{on}(m,l)\|_1}{g(m,l)} > cv^{ct}_{on,\alpha}\right\} \\ &= Pr\left\{\sup_{t\in[0,1]} \frac{\|W(t)\|_1}{t^{\gamma}} > cv^{ct}_{on,\alpha}\right\} = \alpha. \end{aligned} \tag{2.33}$$

Unlike standard CUSUM tests, ratio type statistics do not require to estimate the long-run covariance and are also considered for this reason in this analysis. The precise form of the chosen statistic is given in the following

quadratic form,

$$TS_{on}^{rt}(m,l) = \frac{l^2}{m} E^T(m,l) \left\{ \frac{1}{m^2} \sum_{j=1}^{m} j^2 \left(\overline{X}_{1,j} - \overline{X}_{1,m}\right) \left(\overline{X}_{1,j} - \overline{X}_{1,m}\right)^T \right\}^{-1} E(m,l), \tag{2.34}$$

with its equivalent stopping rule,

$$\tau_{rt}(m) = \min\{l \in \mathbb{N} : TS_{on}^{rt} \geq cv_{on,a}^{rt} g^2(m,l)\}. \tag{2.35}$$

Similarly to the standard CUSUM, the critical value, $cv_{on,a}^{rt}$, is estimated by,

$$\lim_{m\to\infty} Pr\{\tau(m) < \infty\} = Pr\left\{ \sup_{t\in[0,\infty)} \Delta_\gamma(t) > cv_{on,\alpha}^{rt} \right\} = \alpha, \tag{2.36}$$

where,

$$\Delta_\gamma(t) = \frac{1}{\eta_\gamma^2(t)} B^T(1+t) \left(\int_0^1 B(r) B^T(r) dr \right)^{-1} B(1+t),$$

$$\eta_\gamma^2(t) = (1+t) \left(\frac{t}{1+t} \right)^\gamma,$$

and $B(t)$ is a standard Brownian bridge, $t \in [0,\infty)$.

Similarly to the off-line case, the on-line critical values for both test statistics can be computed using Monte Carlo simulations, considering that,

$$cv_{on,\alpha}^{ct} = \sup_{t\in[0,1]} \frac{W(t)}{t^\gamma}, \tag{2.37}$$

$$cv_{on,\alpha}^{rt} = \sup_{t\in[0,\infty)} \Delta_\gamma(t). \tag{2.38}$$

The estimated on-line CP, $\hat{k}_{on}^*$, is derived directly from the value of the stopping time $\tau(m)$, as,

$$\hat{k}_{on}^* = m + \{\tau(m) | \tau(m) < \infty\}. \tag{2.39}$$

2.5.2 Trend Indicator

Considering the on-line procedure, the hypothesis H_1 is two-tailed because the test statistics TS_{on}^{rt} and TS_{on}^{ct} are formulated in a quadratic form and a ℓ_1 norm, respectively. This means that the stopping time rule $\tau_{ct}(m)$ (or $\tau_{rt}(m)$) cannot be an indicator of the direction of a detected change. Thus, to estimate the direction of a change we introduce two indicators: i) based on the CUSUM detector in (2.30), denoted by TI_{ts}; and ii) based on the moving average convergence divergence (MACD) filter [106], denoted by TI_f.

Focusing on TI_{ts}, the indicator is directly derived from the form of the sequential CUSUM detector $E(m, l)$. The detector compares the mean value of the observations that are collected on-line for a chosen monitoring period l, with the mean value of a subsample of the historical data over the predetermined training sample. Hence, for a detected CP, we have that,

$$\begin{cases} E(m,l) > 0, \text{ denotes an upward change} \\ E(m,l) < 0, \text{ denotes a downward change} \end{cases} . \tag{2.40}$$

However, in certain cases, limiting the window over which the direction of a change is estimated to the immediate neighbourhood of a detected CP can be unreliable due to the continuous variability of the time-series. In such cases, we have to estimate the direction of a change by incorporating more elaborate filters; in this context, we estimate the direction of detected changes by applying the MACD indicator. The MACD is based on an exponential moving average (EMA) filter, of the form,

$$EMA_p(n) = \frac{2}{p+1} X_n + \frac{p-1}{p+1} EMA_p(n-1), \tag{2.41}$$

with p denoting the lag parameter. The MACD series can be derived from the subtraction from a short p_2 lag EMA (sensitive filter) of a longer p_3 lag EMA (blunt filter), as described below:

$$MACD(n) = EMA_{p_2} - EMA_{p_3}. \tag{2.42}$$

The trend indicator TI_f is then obtained by the subtraction of a short p_1 lag

EMA filter of a MACD series from the raw MACD series, as described below

$$TI_f(n) = MACD(n) - EMA_{p_1}(MACD(n)),\ p_1 < p_2 < p_3. \tag{2.43}$$

In the evaluation of TI_f three exponential filters are involved. In essence, TI_f is an estimation of the second derivative over an interval around the change (considering that the subtraction of a filtered variable from the variable generates an estimate of its time derivative). In contrast to other works [106], we only adopt TI_f to characterize the direction from the specific value of TI_f at the estimated time of change. We announce an upward change if $TI_f(\hat{k}^*_{on}) > 0$, otherwise, if $TI_f(\hat{k}^*_{on}) < 0$, a downward change.

Finally, we propose a modification of the trend indicator TI_f, converting it from a point estimator to an interval estimator; instead of evaluating $TI_f(\hat{k}^*_{on})$, we propose to evaluate the trend indicator at a time interval $(\hat{k}^*_{on}, \hat{k}^*_{on} + h)$, where h is a threshold parameter:

$$TI_f(\hat{k}^*_{on}, h) = \sum_{l=\hat{k}^*_{on}}^{\hat{k}^*_{on}+h} TI_f(l). \tag{2.44}$$

The proposed $TI_f(\hat{k}^*_{on}, h)$ modification improves the estimator's accuracy; the calculation of the sum of a multitude of observations, after a CP, can smooth out a potential false one-point estimation, especially in the case of small changes.

2.5.3 Overall Algorithm

We outline in *Algorithm 2* the RCPD algorithm, as a combination of the off-line and the on-line phase, in the form of pseudo-code. Beginning from the initial value set for the monitoring starting period, denoted by m_s, the modified off-line algorithm is applied over the whole historical period; the training period m is then defined as the interval elapsed from the last detected off-line CP (if one exists) to m_s. As a second step, the on-line test statistic, $TS_{on}(m, l)$ in (2.27), is applied for a specified monitoring time frame l. If a content popularity change is detected at time instance $\hat{k}^*_{on}$, the trend indicator subroutine is called

Algorithm 2 The Real-time CP Detector (RCPD)

procedure RCPD(X_n,m_s,k)
    ; X_n: time-series of video views
    ; m_s: running end of training period ;
    ; m: training period
    ; l: monitoring time frame
    ; d: period assuming no change
    ; TS_{on}: on-line test statistic (eq. 2.31 or 2.34)
    ; cv_{on}: critical value (eq. 2.37 or 2.38)
    ; $\hat{k}^*_{on}$: the estimated on-line CP (eq. 2.39)
    ; TI: trend indicator (TI_{ts} or TI_f)
    for n in X_n **do**
        if $n = m_s$ **then**
            s=MBS(1,m_s,1) ; calculate off-line CPs
            if `array_length`(s) > 0 **then**
                m=$\{\max(s), m_s\}$; $max(s)$ is the latest CP
            else
                m=$\{max(1, m_s - u), m_s\}$; u a large value
            end if
        else if $m_s < n < m_s + l$ **then**
            calculate $\texttt{TS}_{on}$`(m,l)`
            if $\texttt{TS}_{on}$`(m,l)`$>\text{cv}_{on}$ **then**
                calculate `TI`
                signal CP and estimated direction
                $m_s = \widehat{cp}_{on} + d$; keep a distance from $\widehat{cp}_{on}$
            end if
        else if $n = m_s + l$ **then**
            $m_s = m_s + l$; start a new training period
        end if
    end for
end procedure

to reveal the direction of change.[2] At this point the procedure stops and a new starting point for the monitoring window is defined as $m_s = \hat{k}^*_{on} + d$, where d is a constant value specifying a period assuming no change. Otherwise, if no change is detected after a maximum of l instances, the procedure restarts from the last time point, $m_s = m_s + l$.

[2]In the load balancing scenario discussed in Chapter 3, in the case of an increase in the content popularity a new content cache is being deployed, while conversely a decrease leads to the removal of an existing cache.

2.6 Validation of the RCPD Using Synthetic Data

In this Subsection, we validate the performance of the overall algorithm by performing a series of four different experiments on synthetic data. The use of synthetic data allows us to regulate the parameters of the time-series in terms of mean changes and thus obtain quantitative metrics for the performance of the proposed algorithms.

The choice of the time-series model for the generation of the synthetic data is based on the fact that several studies have shown that ARMA models capture very well content popularity evolution. For example, in [49] it has been concluded that an ARMA model can efficiently describe the daily access patterns of YouTube content, based on an extensive analysis of $100,000$ videos. Similarly, in [107] an ARMA model has been proposed for the estimation of the popularity of video content. Motivated by these findings, for the validation of the proposed algorithm we use an ARMA$(1,1)$ time-series. We generate $1,000$ time-series of length $N = 600$ samples. Without loss of generality, we assume an initial mean value $\mu_0 = 0$, noting that the performance of the RCPD is independent of the initial mean value and only depends on the magnitude of the variation of the mean value before and after a CP.

In the first experiment, we begin with a comparison of the standard BS to the proposed modified BS algorithms described in Section 2.4. We perform two tests; in the first test we introduce two CPs at the instances $k_i^* = (iN)/3,\ i = 1, 2$, while in second test, we introduce four CPs at $k_i^* = (iN)/5,\ i = 1, \ldots, 4$. The two tests are repeated for three different values of the magnitude of a change $\mu_1 = 1,\ \mu_2 = 1.5, \mu_3 = 2$, i.e., we randomly increase or decrease the mean value by $\mu_j,\ j = 1, \ldots, 3$ at the time of change. Table 2.1 summarizes our findings regarding the true and false alarm rates of the two algorithms.

Both the standard and the modified BS algorithms provide similar true alarm rates, exceeding 94%, in the first test. On the contrary, in the more challenging second test, the superiority of the modified BS over the standard BS algorithm is clear. The modified BS algorithm achieves true alarm rates in excess of 70%, even in the demanding scenario of a relatively small change

Table 2.1: Percentage of successful CP detections for the standard and modified BS algorithm

	Test 1: two CPs		Test 2: four CPs	
	True (false) alarm rate		True (false) alarm rate	
μ	BS	modified BS	BS	modified BS
μ_1=1	0.94 (0.06)	0.95 (0.05)	0.5 (0.258)	0.7 (0.05)
μ_2=1.5	0.95 (0.05)	0.95 (0.05)	0.5 (0.258)	0.9 (0.08)
μ_3=2	0.95 (0.05)	0.95 (0.05)	0.47 (0.53)	0.9 (0.1)

Table 2.2: Success rates of trend indicators

	Test 1: two CPs		Test 2: four CPs	
	Success rate		Success rate	
μ	TI_{ts}	TI_f	TI_{ts}	TI_f
μ_1=1	0.99	0.99	0.99	0.99
μ_2=1.5	1	1	1	1
μ_3=2	1	1	1	1

in the mean $\mu_1 = 1$. On the other hand, the standard BS algorithm has in all cases a true alarm rate of less than 50%, rendering any CP detection highly questionable. The second test confirms that the standard BS algorithm is prone to an overestimation of the number of CPs as shown by the high false alarm rates (in excess of 25% in all cases), an issue that can be effectively addressed by the modified BS algorithm which scores false alarm rates below 10%.

Next, in the second experiment, using the same test sets as above, we measure the success rates achieved by the proposed trend indicators TI_{ts} and TI_f for $h = 0$ (larger thresholds provided the same true identification rates). The results are summarized in Table 2.2. The two trend indicators successfully identify the direction of a change in more than 99% of the cases, which shows that they can be interchangeably employed. In the assessment of the performance using real datasets in Sections 2.6 and 2.7, we solely employ the TI_f trend indicator.

We proceed by assessing the proposed RCPD algorithm using both the

standard and the ratio type CUSUM. In this third experiment, we measure the average number of CPs detected, averaged over $1,000$ simulations when a single CP is introduced in the ARMA time-series at the time instance $\frac{N}{2} = 300$. We consider different values for the magnitude of change $\mu \in \{0, 0.5, 0.7, 1, 1.2, 1.5, 2\}$ and the monitoring window length $l \in \{25, 50, 100\}$. We note that we included the case $\mu = 0$ – which corresponds to the absence of a change – to evaluate the false alarm rate of the overall algorithm. We omit results with true alarm rates lower than 50% as they are statistically unreliable. In terms of the remaining algorithmic parameters, we have set the minimum distance between two successive CPs to $d = 50$,[3] the sensitivity parameter to $\gamma = 0.25$ [108] (we choose a neutral value as the behaviour of γ is well studied), and, the significance level to $\alpha = 0.05$. In each test of the third experiment we measure the exact number of CPs detected, tabulated as one the following three values: i) 0 when (falsely[4]) no CP is detected; ii) 1 when (correctly) a single CP is detected; and iii) > 1 when (falsely) multiple CPs are detected. Finally, we measure the median of the time instance of the single CP detection, denoted by $\hat{k}^*$.[5] The results of this experiment are presented in Table 2.3 in the next page, and are discussed below.

Firstly, we observe that both the standard and the ratio type CUSUM achieve very small false alarm rates, inferior to 6% when no CP is inserted, irrespective of the choice of l. On the contrary, the choice of l readily affects the algorithm's success rate for $\mu > 0$; for small changes in the mean value, $\mu = 0.5, 0.7$, a larger monitoring window l increases the algorithm's true alarm rates in identifying correctly the existence of the CP. For medium and high changes in the magnitude of change $\mu = 1, 1.2, 1.5, 2$, it is observed that a high true alarm rate - in excess of 93% for the standard CUSUM - is achieved, while choosing a smaller l can slightly increase the true alarm rates. As a result, depending on the application, a choice of a larger l can be appropriate if the algorithm is to be employed as a universal CP detector. Alternatively, a smaller l can be chosen when the focus is on the identification of large changes in the mean value, i.e., we are interested primarily in detecting CPs of larger

[3]This choice is justified by our observations of the minimum distance between successive CPs in real data sets, presented in Section 2.7.

[4]Except for the $\mu = 0$ case.

[5]We omit the results with true detection rate lower than 50%.

Table 2.3: Results of the RCPDs' algorithm CPs detection for one change in the mean value

		ARMA(1,1)							
μ	l	standard CUSUM				ratio-type CUSUM			
		Number of detected CPs			$\hat{k}^*$	Number of detected CPs			$\hat{k}^*$
		0	1	> 1	med	0	1	> 1	med
	25	0.95	0.05	0	-	0.95	0.05	0	-
$\mu = 0$	50	0.95	0.05	0	-	0.95	0.05	0	-
	100	0.94	0.06	0	-	0.95	0.05	0	-
	25	0.7	0.29	0.01	-	0.8	0.19	0.01	-
$\mu = 0.5$	50	0.16	0.8	0.04	343	0.55	0.43	0.02	-
	100	0	0.93	0.07	341	0.2	0.76	0.04	348
	25	0.26	0.73	0.01	332	0.69	0.3	0.01	-
$\mu = 0.7$	50	0	0.96	0.04	326	0.3	0.65	0.05	328
	100	0.01	0.91	0.08	331	0.05	0.89	0.06	335
	25	0.01	0.97	0.02	327	0.52	0.46	0.02	-
$\mu = 1$	50	0	0.96	0.04	316	0.08	0.86	0.06	321
	100	0	0.92	0.08	321	0	0.95	0.05	323
	25	0.01	0.97	0.02	323	0.43	0.54	0.03	331
$\mu = 1.2$	50	0	0.95	0.05	316	0.02	0.93	0.05	317
	100	0	0.93	0.07	318	0	0.93	0.07	318
	25	0	0.97	0.03	320	0.36	0.6	0.04	329
$\mu = 1.5$	50	0	0.95	0.05	310	0	0.94	0.06	313
	100	0	0.93	0.07	314	0	0.94	0.06	318
	25	0	0.97	0.03	310	0.26	0.71	0.03	317
$\mu = 2$	50	0	0.95	0.05	307	0	0.93	0.07	310
	100	0	0.94	0.06	310	0	0.94	0.06	313

magnitude.

Secondly, we observe that overall, the ratio type CUSUM is outperformed by the standard CUSUM in all tests. Consequently, the standard CUSUM based detector can be considered as an efficient universal choice. Finally, we observe that the lag between $\hat{k}^*$ and the actual instance of change at the point 300 decreases with increasing μ, ranging from 343 to 307, while it appears less sensitive to changes in l. This demonstrates that, intuitively, larger magnitude changes can be detected faster. This result is important for load balancing applications as it provides us with the means to quickly respond to significant changes in the network traffic.

Subsequently, in Table 2.4 in the next page, we present the outputs of the fourth experiment in which we assess the performance, averaged over $1,000$ simulations, of the RCPD algorithm when two CPs are inserted in the ARMA time-series. We introduce a change at the time instance $k_1^* = \frac{N}{3} = 200$ and a second CP at the time instance $k_2^* = \frac{2N}{3} = 400$. We investigate the true and false alarm rates for $\mu \in \{0.5, 0.7, 1, 1.2, 1.5, 2\}$ and $l \in \{25, 50, 100\}$, while the rest of the parameters retain the values of the third experiment. In each test of the fourth experiment we measure the exact number of CPs detected, tabulated as one the following three values: i) < 2 when (falsely) less than two CPs are detected, ii) 2 when (correctly) two CPs are detected, and iii) > 2 when (falsely) more than two CPs are detected. Finally, we measure the median of the detection instances of the two CPs, denoted by $\hat{k}_1^*$ and $\hat{k}_2^*$, respectively (we omit the results with true detection rate lower than 50%).

Similarly to the third experiment, we observe that increasing l increases the true alarm rates for small magnitudes in the mean changes $\mu = 0.5,\ 0.7$, while this trend is reversed in high magnitudes $\mu = 1.5,\ 2$. For medium values $\mu = 1,\ 1.2$ the effect of l on the true alarm rates is less than 2%. Furthermore, in agreement with the outputs of the third experiment, with increasing μ the algorithms achieve increasingly high success rates, over 93% for the standard CUSUM when $\mu \geq 1$.

In addition, the superior performance of the standard CUSUM is reconfirmed in all the tests of the fourth experiment. Finally, with respect to the lag in the estimation of the time instances of the CPs, we observe that, as in experiment three, larger magnitude changes can be detected faster, e.g.,

Table 2.4: Results of the RCPDs' algorithm CPs detection for two mean changes

		ARMA(1,1)									
μ	l	standard CUSUM					ratio-type CUSUM				
		Number of detected CPs			$\hat{k}_1^*$	$\hat{k}_2^*$	Number of detected CPs			$\hat{k}_1^*$	$\hat{k}_2^*$
		< 2	2	> 2	med		< 2	2	> 2	med	
	25	0.88	0.12	0	-	-	0.95	0.05	0	-	-
$\mu_1 = 0.5$	50	0.38	0.60	0.02	251	440	0.79	0.2	0.01	-	-
	100	0.1	0.87	0.03	242	443	0.54	0.44	0.02	-	-
	25	0.41	0.58	0.01	230	427	0.9	0.1	0	-	-
$\mu_1 = 0.7$	50	0.06	0.91	0.03	223	427	0.58	0.41	0.01	-	-
	100	0.01	0.93	0.06	227	428	0.25	0.72	0.03	231	439
	25	0.04	0.93	0.03	219	420	0.74	0.25	0.01	-	-
$\mu_1 = 1$	50	0.03	0.93	0.04	215	419	0.26	0.71	0.03	221	423
	100	0	0.94	0.06	217	420	0.05	0.9	0.05	220	424
	25	0.01	0.96	0.03	214	414	0.56	0.42	0.02	-	-
$\mu_1 = 1.2$	50	0	0.95	0.05	212	416	0.17	0.79	0.04	215	428
	100	0	0.94	0.06	217	420	0.02	0.93	0.05	216	421
	25	0	0.98	0.02	211	411	0.33	0.63	0.04	213	417
$\mu_1 = 1.5$	50	0	0.94	0.06	209	413	0.1	0.85	0.05	213	415
	100	0	0.94	0.06	211	415	0	0.96	0.04	216	419
	25	0	0.98	0.02	208	407	0.12	0.85	0.03	210	412
$\mu_1 = 2$	50	0	0.95	0.05	207	410	0.3	0.91	0.06	209	413
	100	0	0.94	0.06	209	411	0	0.96	0.04	211	414

for $\mu = 2$ a lag inferior to 11 instances is observed for both CPs with the standard CUSUM, irrespective of l.

Concluding this Section, we have presented an extensive set of experiments that provide strong evidence for the efficiency of the proposed algorithms. We have explicitly demonstrated the superiority of the modified BS over the standard BS algorithm and confirmed the validity of the proposed trend indicators. Subsequently, we evaluated the performance of the overall algorithm for various values of μ and l. We have shown that the RCPD algorithm achieves extremely high true alarm rates for larger values of μ, while increasing the length of the monitoring window l can significantly impact the performance for small values of μ. Finally, overall, the standard type CUSUM outperforms the ratio type CUSUM and should be preferred.

2.7 Performance Evaluation Using Real Data

Here, we investigate the performance of the proposed algorithms using a real dataset provided within the framework of the CONGAS project [109]; the dataset consists of the number of views of 882 YouTube videos, observed over $N = 1,000$ instances.

2.7.1 Statistical Properties of the Real Dataset

First, we evaluate the validity of the most important underlying assumption of this analysis, that the content popularity can be modeled as the sum of a constant mean and a weak-dependent (t-dependent) stochastic process, as given in (2.14). A first intuitive method to test whether the time-series is short-range dependent (SRD) is through its autocorrelation function (ACF). The ACF for a weakly-stationary process $\{X_t : t \in \mathbb{N}\}$ with mean value μ is given by,

$$\rho(k) = \frac{\varepsilon(X_t - \mu)(X_{t+k} - \mu)}{\sigma^2}.$$

Note that if $\sum_{k=-\infty}^{\infty} \rho(k) \to \infty$ the process has long-range dependence (LRD), while if $\sum_{k=-\infty}^{\infty} |\rho(k)| < \infty$ it exhibits SRD. To distinguish between these two

phenomena, we use the following functional form of the ACF,

$$\rho(k) \sim C_i^{2H-2}, \text{ as } i \to \infty,$$

where $C_i > 0$ and $H \in (0, 1)$ is the Hurst exponent characterizing the LRD,i.e., $H \in (1/2, 1)$ indicates the presence of LRD. It is challenging to accurately estimate the Hurst exponent out of real data [110] and several methods have been proposed in the literature [111].

In this work, we apply two semi-parametric tests, identified as accurate options among others presented in the survey paper [111]. The first method uses the discrete second order derivative in the time domain while the second uses the discrete second order derivative in the wavelet domain. Both methods estimate an $H \leq 0.5$ for 92% of the YouTube time-series, indicating the validity of our assumptions related to the equation (2.14).

2.7.2 Performance of the Off-line Training Phase

First, we test the hypothesis H_0 of no change in the mean structure on our dataset. H_0 is rejected in approximately 70% of the cases, for a significance level of $a = 0.05$. This outcome indicates that CP algorithms can identify changing content dynamics in real times series. Next, we estimate the number of CPs, by applying the extended off-line algorithm. The corresponding results are illustrated in Fig. 2.4 in the next page, and indicate a sufficiently high number of content popularity anomalies (i.e., mean changes). Hence, a CP analysis is indeed a suitable tool for content popularity detection.

To evaluate the performance of the proposed trend indicator TI_f, we need a baseline independent assessment of the direction of change. We declare that a real increase in the mean value of content visit exists if

$$\mathsf{E}[X(\hat{k}^*_{i-1,off}) : X(\hat{k}^*_{i,off})] < \mathsf{E}[X(\hat{k}^*_{i,off}) : X(\hat{k}^*_{i+1,off})], \tag{2.45}$$

or, that a real decrease in the number of visits exists if

$$\mathsf{E}[X(\hat{k}^*_{i-1,off}) : X(\hat{k}^*_{i,off})] > \mathsf{E}[X(\hat{k}^*_{i,off}) : X(\hat{k}^*_{i+1,off})], \tag{2.46}$$

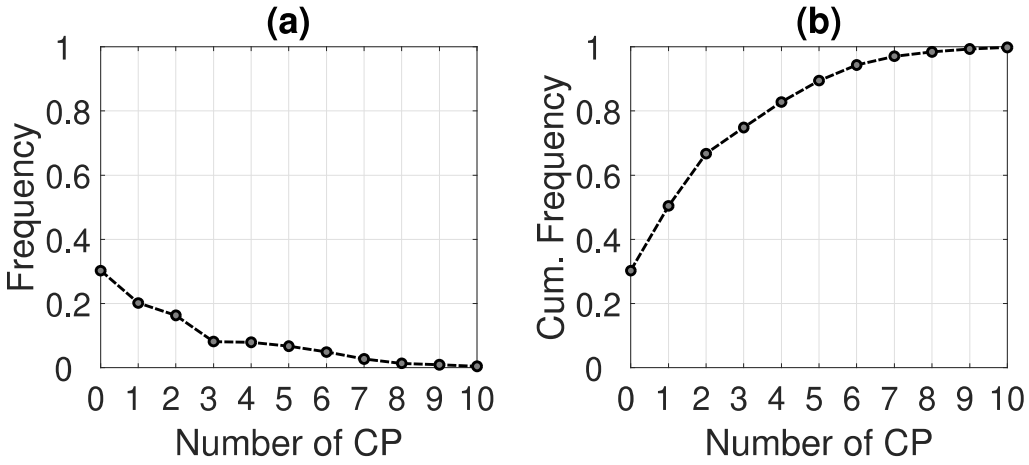


Figure 2.4: Estimated a) frequency and b) cumulative frequency of the number of CPs per time-series.

Table 2.5: Success rates of TI_f trend indicator

h	0	3	5	7	10
Video Set 1	0.69	0.91	0.95	0.97	0.98
Video Set 2	0.90	0.99	0.99	0.99	0.99

where $i = 2, \ldots, N-1$ and $\mathsf{E}[\cdot]$ denotes the numerical average. We test the modified MACD TI_f on two sets of videos. The first set, Video Set 1, comprises the whole dataset, while the second set, Video Set 2, comprises only the videos with a considerable average number of visits (> 10), i.e., for which, $\mathsf{E}[X(1) : X(1000)] > 10$.

The percentage of successful TI_f identifications are tabulated in Table 2.5 for five values of the parameter h, namely $h = 0$, 3, 5, 7 and 10, where h denotes the TI_f's calculation threshold. Commenting on the results for Video Set 1, the TI_f trend indicator works well, except for $h = 0$, providing at least 90% correct direction identifications. As expected, as h increases the procedure works better. More specifically, an $h \geq 5$ parameter choice yields a success rate of 95%, while if a more agile estimation is needed then an $h \geq 3$ still maintains a 91% accuracy. Considering the interim time between consecutive changes, we deduce that an $h \leq 7$ is preferable. Regarding Video Set 2, we see that the results are highly improved, indicating that the procedure works even better for the most popular videos. In practice, this represents the more interesting

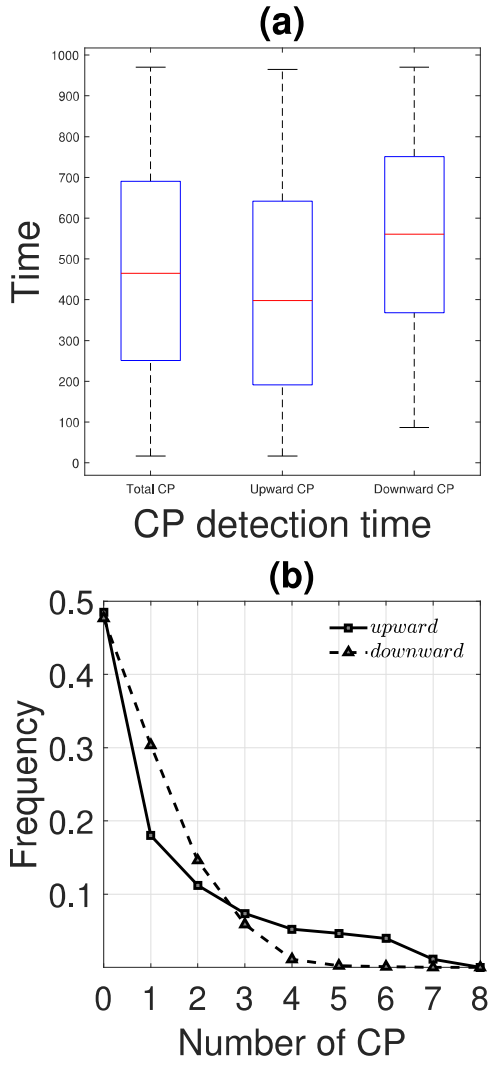


Figure 2.5: Frequency values of the number of upward and downward CPs, per time-series.

scenario as it will have a greater impact in terms of the applied load balancing mechanism.

Furthermore, in Fig. 2.5, the time instances of upward and downward changes are shown in the form of a boxplot. It is intuitive that upward changes occur earlier than downward changes. Moreover, Fig. 2.5 demonstrates that the multitude of upward changes is greater than the respective of downward changes, indicating that decreases in popularity are sharper than increases. In particular, we estimated that out of the total number of changes, 67% are upward.

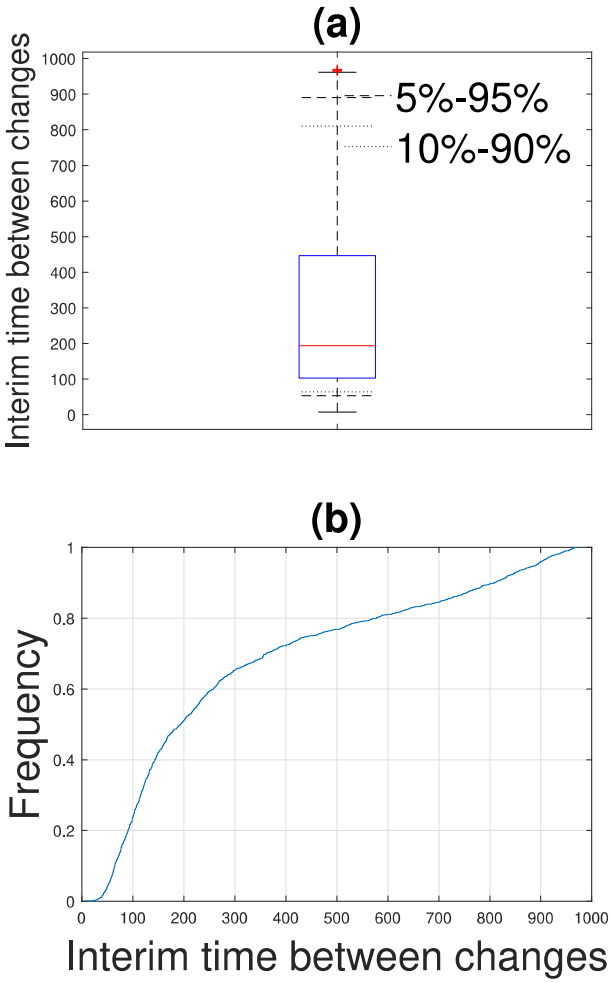


Figure 2.6: a) Boxplot including the interval $(5\% - 95\%)$ (dashed line) and $(10\% - 90\%)$ interval (dotted line), b) Cumulative frequency for the interim time of consecutive CPs.

Finally, we analyze the interim time between consecutive CPs. The results presented in Fig. 2.6 illustrate the existence of a sufficiently large gap between consecutive potential changes. 90% of the intervals corresponding to consecutive CPs exceed 70 time instances and only 5% of them are shorter than 50 time instances, ensuring that a sufficiently large training window can be applied. The results depicted in Fig. 2.6 allow adjusting parameters of the on-line phase, in particular the minimum time interval between consecutive changes, denoted by the parameter d.

2.7.3 Evaluation of the RCPD Algorithm

In the previous Subsection, we have evaluated the performance of the off-line algorithm and demonstrated its efficiency, as well as how it is employed in determining parameters of the on-line phase, such as the interval assuming no change d and the threshold parameter of TI_f, h.

We further employ the off-line algorithm as a benchmark against which the performance of the RCPD algorithm will be evaluated. We note that the off-line analysis provides the *best possible statistical detection* of the actual mean changes, as off-line algorithms operate retrospectively over the entirety of each of the time-series. Thus, in absence of a priori knowledge of the actual CPs in the real data (as opposed to the synthetic data in which the CPs were controlled), we evaluate the performance of the RCPD procedure by measuring the "similarity" of its outputs (detected CPs, instances of detection and trends) to the corresponding outputs of the off-line version.

As the number of detected CPs and / or their exact positions are likely to differ at the output of the retrospective (off-line) and of the RCPD algorithm, in order to obtain a measure of their similarity, we estimate their dynamic time warping (DTW) distance. The DTW is a dynamic programming tool that measures distances between asynchronous sequences and is widely used by the speech processing community [54].

The results are presented in Fig. 2.7, where the estimated DTW distances are depicted for several values of the monitoring window length $l \in [40, 150]$, to investigate the consistency of parameter l over different values. In the RCPD algorithm we use $d = 50$ (minimum distance between two changes) and have

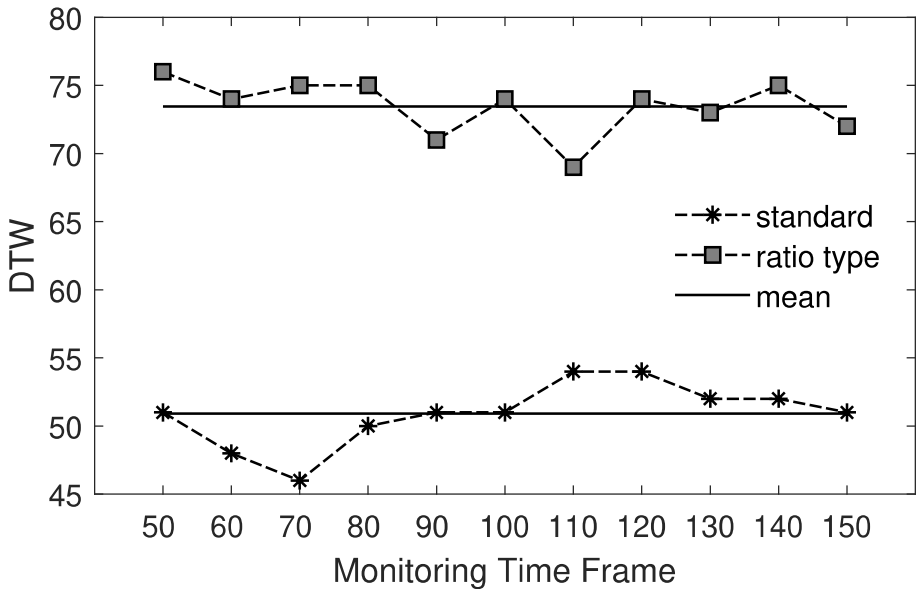


Figure 2.7: DTW distances for the two on-line detection schemes.

Table 2.6: Empirical percentiles of mean values change rate

	Percentiles Threshold			
	10%	15%	25%	50%
Standard	9%	13.1%	20.8%	42.21%
Ratio type	9.5%	14.82%	28.22%	67.40%

set the sensitivity parameter to $\gamma = 0.25$. The estimated mean DTW distance for the standard CUSUM is 52 and for the ratio-type CUSUM is 73. For comparison purposes, we note that the corresponding DTW distance over the synthetic data is 20 for medium / large changes, while the true CP detections are around 95%. As a result, we can infer, that the outputs of the on-line algorithm, using the standard CUSUM, are "very close" to the outputs of the benchmark off-line algorithm. In agreement with our observations over the synthetic data, the DTW distance using the ratio-type CUSUM is clearly larger.

We also study the magnitude of the detected CPs. We define as the CP magnitude the percentage-wise change in the mean values before and after the CP. We group the measured magnitudes for all change points using the four percentile threshold values 10%, 15%, 25% and 50%, i.e., reflecting the

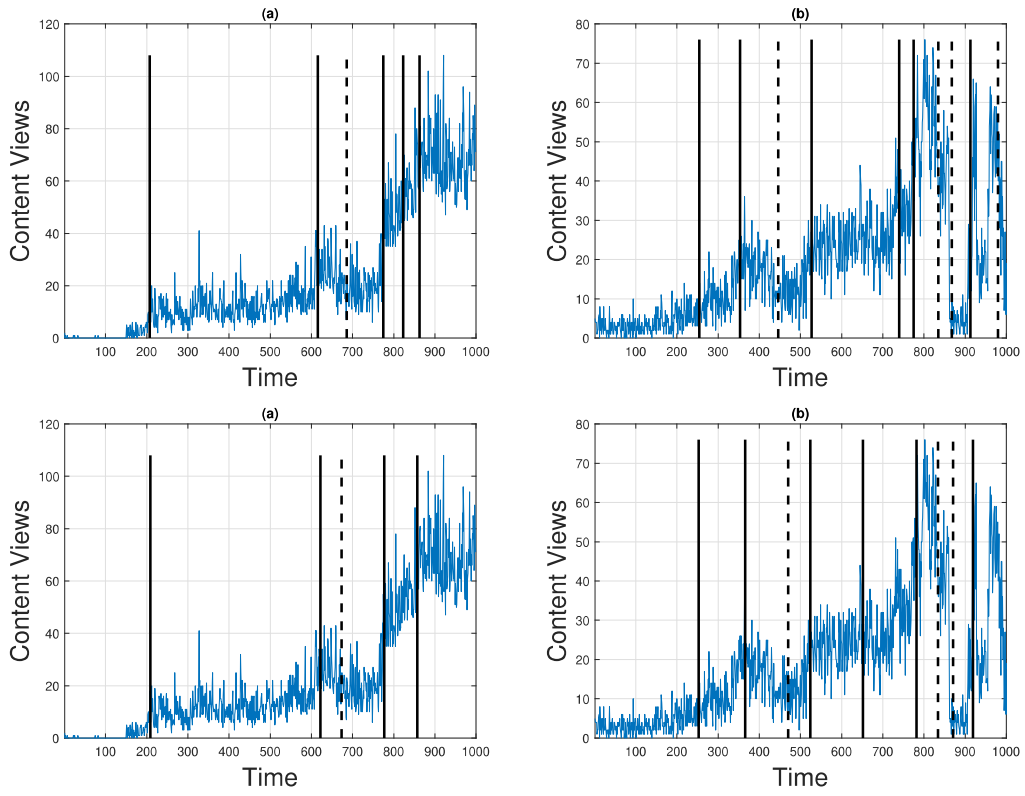


Figure 2.8: Outputs of the RCPD algorithm; using standard CUSUM (upper row) and ratio type CUSUM (lower row) for different time-series. Solid and dashed lines depict an upward and a downward change, respectively.

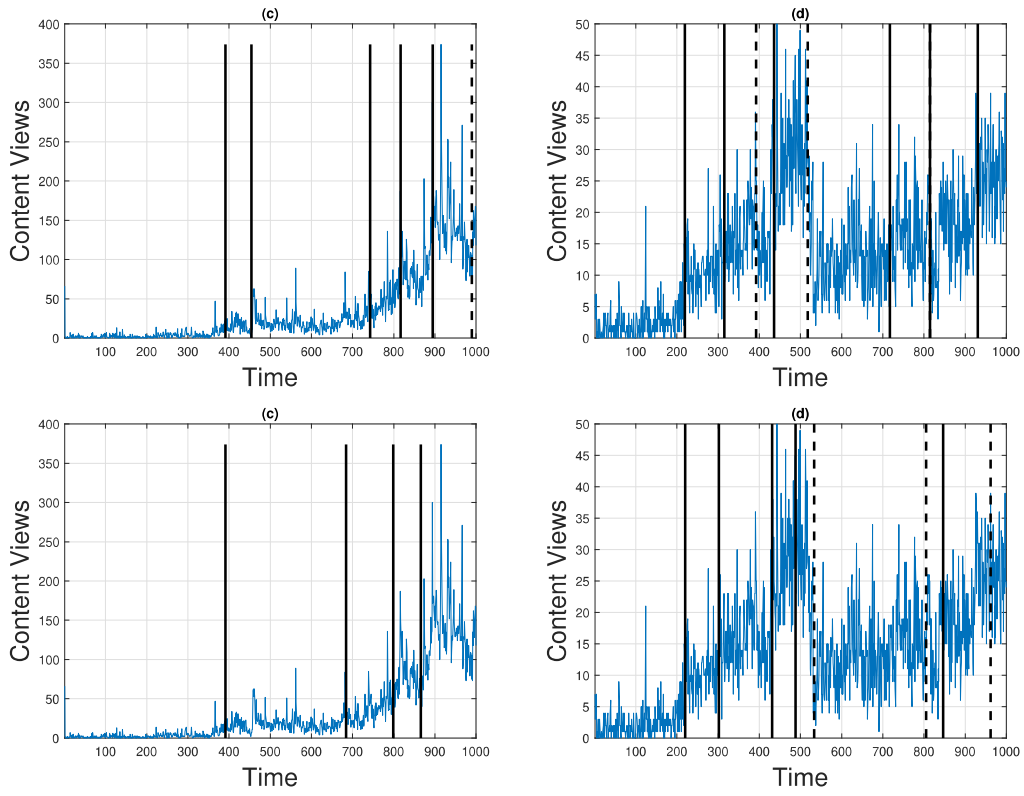


Figure 2.9: Outputs of the RCPD algorithm; using standard CUSUM (upper row) and ratio type CUSUM (lower row) for different time-series. Solid and dashed lines depict an upward and a downward change, respectively.

Table 2.7: Percentages of time-series with time dependencies exceeding t samples

t	≥ 1	≥ 5	≥ 10	≥ 15	≥ 20	≥ 25	≥ 40
CP segmented	0.97	0.63	0.41	0.23	0.14	0.087	0.03
Randomly segmented	0.97	0.88	0.81	0.73	0.67	0.57	0.28

frequency of magnitudes exceeding the respective thresholds. The results are summarized in Table 2.6. According to our results, both the standard and ratio type CUSUM algorithms detect the most significant changes in the content popularity. Moreover, ratio-type CUSUM detects, in general, CPs with the largest magnitude of change, in agreement with synthetic data results.

Additionally, for illustration purposes, we depict the RCPD algorithm's outputs for four different time-series. We set the beginning of the monitoring period at $m_s = 200$ and monitoring horizon $l = 50$, the on-line parameter $g = 0.25$ and the significance level to $a = 0.05$. The corresponding results are depicted in Fig. 2.8 and 2.9, showing the estimated CPs by applying the standard CUSUM and the ratio type CUSUM procedures, respectively. In both cases, the estimated changes correspond to the real content popularity changes; visual inspection suggests that the performance of the standard CUSUM is more reasonable (e.g., Fig. 2.9d). The RCPD, as it is illustrated in Fig. 2.8b seems to be adaptable to "fast" changes; without getting "confused" by random peaks in the time-series, such as those in Fig. 2.8a or in Fig. 2.9c.

2.7.4 Time Dependencies of Piecewise Time-series

We also compare the autocorrelation function of the piecewise - divided by the detected CPs - time series (CP segmented), with the randomly segmented - in equal lengths - time series (randomly segmented). Results are tabulated in Table 2.7 and advocate for the short dependence structure of the largest part of the dataset; significant lags in time dependencies higher than 15 instances and 20 instances can be found in less than the 23% and the 14% of the CP segmented time-series, respectively. Furthermore, the fact that the ACF of the CP segmented time-series drops to zero quickly, especially compared to that of the randomly segmented, intuitively indicates that the detected CPs split the

time-series into stationary segments, which, additionally, confirms indirectly the accuracy of the off-line CP estimations over the changes in the real data.

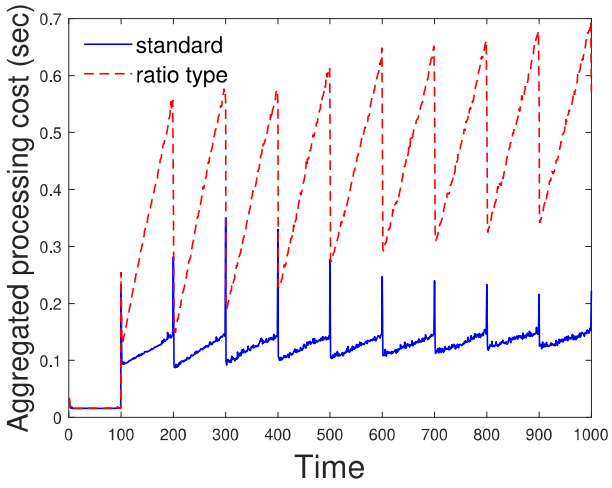


Figure 2.10: The aggregated overall processing cost, per time-instance, of the RCPD algorithm over 882 time-series.

2.7.5 Computational Complexity and Scalability

Finally, we present a MATLAB ® implementation of the overall algorithm with a large number of time-series (882 in this experiment) to quantify its performance in terms of processing cost. The computational time is measured on a Lenovo IdeaPad 510-15IKB laptop, with an Intel Core i7-7500U @ 2.70 GHz processor and 12 GB RAM. In Fig. 2.10, we show the aggregate processing cost per time instance for the two on-line methods and the total number of time-series. For the first 100 time instances, the algorithm collects the initial data, since it bootstraps. The peaks indicate the off-line part of the algorithm, which is more processing demanding mainly due to the segmentation algorithms running in parallel. The on-line part in the standard on-line algorithm indicates a linear complexity, since it is based on (2.31), while the equivalent quantity in (2.34) of the ratio-type is more CPU intensive, justifying the comparatively higher processing cost of the latter algorithm. In both cases, the aggregate processing cost is typically much less than a second, which demonstrates the lightweight nature of the proposed scheme. Such results could be further improved with a distributed deployment of scheme replicas since each of the

time-series could be processed independently.

2.8 Conclusions

In this Chapter, we proposed the employment of on-line change point (CP) analysis to implement real-time, autonomous and low-complexity video content popularity detection. The RCPD was validated against synthetic data, as well as a large database of real YouTube video visits. It was demonstrated that the RCPD can accurately identify changes in the average content popularity and the direction of change. In particular, the success rate of the RCPD over synthetic data was shown to exceed 94% for medium and large changes in content popularity. Additionally, the dynamic time warping distance, between the actual and the estimated changes, has been found to range between 20 samples on average, over synthetic data, to 52 samples, in real data. The rapid responsiveness of the RCPD is instrumental in the deployment of real-time, lightweight load balancing solutions, as shown in a real example.

Chapter 3

Extended Real-time Change Point Detector and Applications

3.1 Introduction

In this Chapter, we first focus on an extension of RCPD algorithm in the direction of revealing changes not only in the mean but also in the variance structure of content popularity metrics. Next, motivated by the efficient performance of the developed algorithms, i.e., in terms of accuracy and complexity, we investigate their behavior in two novel applications. More specifically, we apply the extended RCPD algorithm to drive a responsive, i.e., through lightweight virtual machines, edge cloud environment for content distribution, as well as in intrusion detection for software-defined wireless sensor networks (SDWSNs).

We now motivate the need to integrate a variance based CP method to RCPD. Higher order moments of an underlying random process are unarguably important for the efficient statistical characterization of content popularity; in particular, "volatility" plays a central role in capturing the underlying dynamics of content views. As an example, in caching applications, it has been established in [112] that a major factor greatly impacting efficiency is related to demand volatility; this reflects the fact that content might not be constantly requested, following a stationary model, but rather, only be requested once or twice and subsequently exhibit vanishing demand in time (e.g., volatility in YouTube content). Based on these findings, an efficient strategy for resource provisioning should, in principle, consider not only conditional mean demands but also

demand fluctuations, thus enabling efficient resource allocation strategies that avoid under-provisioning or over-provisioning of resources.

Similarly to Chapter 2, instead of attempting to *predict* the evolution of content popularity, in this work we rather focus on *detecting* changes in its underlying statistics, and doing so in real-time. To this end, we propose the use of on-line change point (CP) analysis; to complement our work [80, 82] that focused on the identification of changes in the mean of a time series; in this extension we alternatively investigate the performance of corresponding on-line algorithms to identify changes in the variance of a time series using CP analysis.

Moreover, we exploit the usage of RCPD in content distribution network (CDN) platforms [18]. CDNs are usually tightly coupled with cloud providers (e.g., Akamai, Mirror Image, Microsoft Azure, Amazon) that use their own hardware, sometimes customized. The CDN software may be proprietary, costly for SMEs and with specific hardware or OS requirements. Such approaches deliver transparently and efficiently content to the end-users. However, traditional CDN servers are typically far away from the content consumers and are unsuitable for the ultra delay-sensitive applications envisaged by the 5G networking initiatives [113]. Hence, we argue that there is a need for open, flexible, extensible, hardware-independent and resource-efficient CDN solutions hosting the content near to the users with unikernel-based virtual machines [19, 21]. We investigate the usage of the RCPD algorithm, presented in Chapter 2, as an essential feature of a relevant platform.

Finally, we note that customarily CP analysis is employed in the detection of anomalies in times series. Therefore, as a natural extension of this work, we further consider the application of the RCPD for anomaly detection in SDWSNs. SDN is a promising technology attempting to overcome many challenges in wireless sensor networks (WSN), particularly with respect to flexibility, resource-efficiency, and resource reuse. Notably, it is now argued that SDN and related technologies should be integrated to facilitate the management and operations of edge servers and various IoT devices [114]. Conversely, the centralization and the separation of the control from the data plane turn SDNs vulnerable to new security threats, e.g., distributed denial of service (DDoS) attacks, which carry over to SDWSNs. State-of-the-art approaches to identify DDoS do not

always take into consideration restrictions in typical WSNs, e.g., computational complexity and power constraints, while further performance improvement is always a target. Our objective in this study is to propose a lightweight but efficient DDoS attack detection approach using the RCPD.

3.2 Contributions and Chapter Organization

3.2.1 Extensions of the RCPD for the Detection of Changes in the Variance of Video Content Popularity

The rapid increase of the video content traffic, which tends to become the vast majority of the global IP traffic, increases even more the need for related efficient cloud resource allocation approaches. In this context, to avoid under- or over-provisioning of resources, it is important to rapidly detect and respond to changes in content popularity dynamics, including volatility, i.e., changes in the second order moment of the underlying process.

Thus, in this work we propose an efficient combination of an off-line and various on-line procedures for the detection of changes in the second order statistics of video content popularity, as soon as they occur (i.e., at real-time). The proposed detector is built upon our earlier proposal for a real-time CP detector of mean changes in data series, that we applied to monitor the average number of video content [80] and [82].

To further illustrate our motivation behind this work, we note that an overall approach considering both mean and variance changes allows for a more efficient handling of content popularity changes as highlighted in Fig. 3.1, illustrated in the next page. For example, Fig. 3.1 (a) depicts that a crucial popularity change may affect only the variance parameter, in the specific example at the third segment of the time series. On the other hand, Fig. 3.1(b), depicts that in the case of a simultaneous change in the mean and the variance, e.g., in the second segment of the time series, the latter is critical to estimate the actual impact of this change. Monitoring the variance may also be used as a measure of uncertainty, determining the degree of fluctuation of popularity around its expectation; for instance, compare the behaviour of the time series in Fig. 3.1(b) after the first and the second CP (second and third segments of

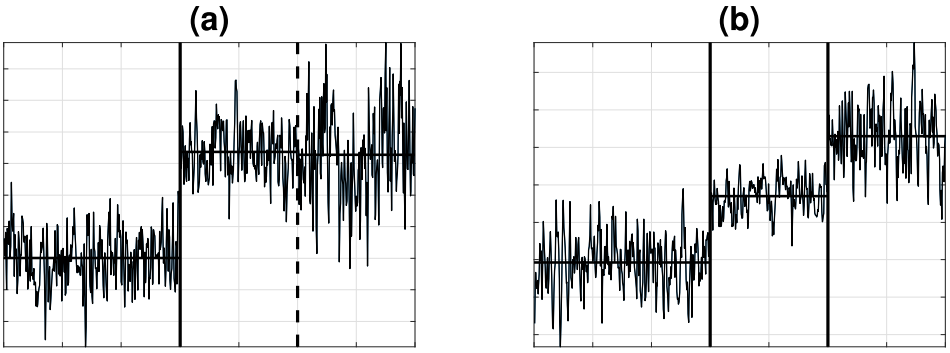


Figure 3.1: Simulated time series with CPs in the mean (solid line) and the variance (dashed line) for (a) separated and (b) simultaneous changes in the mean / variance. Horizontal lines illustrate the mean value.

the data series, respectively).

To identify changes in the variance, a more elaborate test statistic is employed in the present study. With respect to [80, 82], we further introduce novel on-line tracking mechanisms based on autoregressive moving average (ARMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models. The most important novel aspects of this Chapter are listed below:

- We show that variance CP detection is important in the context of content popularity.
- We introduce a relevant on-line detection algorithm, enhanced by the following two mechanisms: (a) an offline CP detection over training data for the estimation of the on-line test parameters; and (b) identification of the change magnitude in the pro- and post-change variance structure.
- Our algorithm supports three alternative on-line tests for content popularity detection – based on ARMA and GARCH models as well as a non-parametric approach – covering a wide-range of time series characteristics.
- We performed experiments both on synthetic and real time series datasets. Our results show that: (i) the GARCH and the non-parametric approaches perform better when the time series does not follow a linear model; (ii) overall, these approaches can generalize better with respect to the true

alarm rates; and (iii) the non parametric approach can identify CPs more rapidly.

3.2.2 CP Analysis for Resource Allocation in a Novel CDN Platform

In this thesis, we consider a novel CDN platform, called Unikernel-Based CDN (UNIC) [19], which provides efficient content caching through placing Unikernel-based VMs hosting popular content near the users. In comparison to traditional CDN solutions, UNIC provides the following advantages: (i) it can operate over heterogeneous hardware devices with diverse capabilities, including lightweight edge clouds; (ii) it provides significant elasticity capabilities through tiny VMs orchestration; (iii) it supports modular extensibility with new mechanisms; and (iv) it defines new research problems emerging from bringing together the content-caching approaches (e.g., [115], [116]) with the VM orchestration proposals.

Furthermore, UNIC supports the following novel features:

- Modular orchestration of Unikernel-based VMs hosting replicas of Internet content, such as for configurable VM placement;
- Content popularity changes detection mechanisms that drive the content VMs deployment based on a novel CP detection methodology tailored to the specific problem;
- Dynamic load balancing using a bespoke DNS service attached to the VM orchestration;
- Real-time monitoring of server resource utilization and end-user performance.

We designed and implemented UNIC in the context of the MONROE [1] research project, introducing the UNIC platform and focusing on its two core features: (i) the modular VM orchestration (e.g., placement); and (ii) the detection of content popularity changes. We tested our CP methodology and experimented with UNIC using real youtube popularity measurements. [109] to

[1] https://www.monroe-project.eu/

3.2.3 CP Analysis for Anomaly Detection in SDWSNs

We explore the application of the RCPD for anomaly (intrusion) detection in SDWSNs. The SDN paradigm was devised to simplify network management, avoid configuration errors and automate infrastructure sharing in wired networks [117]. The aforementioned benefits motivated the discussion of combining SDN and WSNs as a solution to many WSN challenges, in particular concerning flexibility and resource reuse [37]. This combination is referred to as SDWSN. The SDWSN approach decouples the control plane from the data plane and centralizes the control decisions; its main characteristic is the ability to program the network operation dynamically [14]. Recent results show that SDWSNs can perform as well as the IPv6 routing protocol for low-power and lossy networks (RPL) [118].

On the other hand, the SDN centralization and the planes' separation turn the network vulnerable to new security threats (explained in Section 3.9.1), a property that is inadvertently passed on to SDWSNs. Shielding SDNs from these vulnerabilities has already attracted a lot of attention in the literature with proposals to implement attack detection in IoT networks using SDN. Overall, in the case of SDWSNs, due to the resource constraints of the nodes, most of the security mechanisms designed for non-resource constrained SDNs have to be adapted or redesigned. This is one of the major challenges for SDWSN security.

Considering the limitations of previous works, our main objective is to propose a mechanism for DDoS detection with, i) a high detection rate, and, ii) low complexity, so that it would be suitable for "restricted" networks. To this end, we propose the employment of the RCPD [80] [82]. We study two DDoS attacks: a false data flow forwarding (FDFF) attack, and a false neighbor information (FNI) attack, chosen to illustrate the proposed algorithm's capabilities in the case of specific SDWSN vulnerabilities that exhibit largely different behavior. Both attacks are explained in Section 3.9.2. We have tested our approach on the IT-SDN framework[2] [118] and our results show that we can detect these attacks with a detection rate close to 100%, improving the state of the art; importantly, it is further possible to gain insight regarding the

[2]http://www.larc.usp.br/users/cbmargi/www/it-sdn/

type of the attack, based on the metric that provides the quickest detection, a feature, that to the best of our knowledge, breaks new ground in the domain of DDoS analysis for SDWSNs.

3.2.4 Chapter Organization

The rest of this Chapter is structured as follows: In Section 3.3, we consider a complementary - to that of Section's 2.3 - background, related to parametric time-series approaches for content popularity modeling. Also, a brief discussion to the application fields in which the RCPD is involved, is included. In Section 3.4, the offline training is presented in detail, while Section 3.5 presents three different approaches for the construction of the online test statistic. The integrated algorithms are assessed on synthetic data in Section 3.6 and applied to real YouTube content view data in Section 3.7. Next, we demonstrate the load balancing gains achieved through the use of RCPD, introducing the UNIC platform's architecture and presenting our experimentation methodology and results. Moving to intrusion detection in SDWSNs, Section 3.9.1 discusses the challenges in SDWSN security analysis, while, in Section 3.9.2 we illustrate the FDFF and FNI attacks. Experimental methods are presented in Section 3.9.3 and results on intrusion detection using the RCPD are presented in Section 3.9.4.

3.3 Background and Related Works

In this Subsection, we discuss how this work relates to i) the literature of video content popularity prediction; ii) the traditional cloud architectures and especially the existing CDN platforms; and iii) the anomaly detection, in general, and in SDNs and SDWSNs, in particular.

3.3.1 Time Series Models for Video Content Popularity

As we discussed in Chapter 2, the prediction of video content popularity characteristics and dynamics [83], as well as models to predict popularity evolution, e.g., [46] and [84], is a well studied topic in the literature. In this

Section we focus on time series modeling, aligned to the scope of the specific Chapter; parametric CP tests based on time series.

In particular, linear, non linear and hybrid models have invariably been proposed. In early works, linear time series models have been used, e.g., the authors in [49] introduce an ARMA(7, 7) model to describe and predict the daily views of individual videos. Alternatively, in [119], by taking into consideration seasonality, an autoregressive integrated moving average (ARIMA) model is used to forecast the popularity of online content. Other approaches include fractional ARIMA (FARIMA) models, that capture both short-range dependence (SRD) and long-range dependence (LRD) statistical properties [120].

Recently, non linear models have further been proposed to take into account the conditional heteroskedasticity and the conditional volatility of the data series (seen as a stochastic process). In these cases, GARCH models are involved. For example, in the comparative study [121], the authors showed that a hybrid ARIMA / GARCH model was superior to FARIMA and wavelet neural network models, while in [122], a similar hybrid FARIMA / GARCH approach was also introduced. In essence, the existing hybrid models consider the second order characteristics of a time series as a supplementary element to further improve the forecasting or estimation of the content popularity. More precisely, these solutions assume conditional heteroskedasticity for the errors of the ARMA or FARIMA model. An exception can be found in [123], where a video demand predictor forecasts the volatility and correlation of the streaming traffic associated with different videos, based on multivariate GARCH models.

On the other hand, the problem of detecting (i.e., estimating), non- parametrically and in real time, CPs on content popularity sequences, has not been adequately investigated yet. Among a handful of related studies, in our previous works [19, 21, 80–82] we proposed and implemented a real-time, non-parametric and low-complexity video content popularity CP detector (as opposed to predictor) for changes in the mean value of video content popularity. In the present contribution, in contrast to [80, 82], we introduce an innovative online algorithm for the detection of CPs in the second order statistics of content popularity data. We also present an enlarged statistical framework, that includes parametric as well as non-parametric detectors.

Our algorithm can be used as a “stand alone” mechanism, but may also

be a helpful complementary tool for prediction approaches. With respect to the latter, it can be employed in validating whether assumptions made by a prediction model are still reasonably satisfied, or, whether the prediction model / procedure needs adjustment. Since, data are often influenced by a multitude of external factors, stationarity assumptions cannot be guaranteed over the whole monitoring period, especially for long time ranges.

3.3.2 Cloud Architectures for Efficient Resource Allocation

The efficient VM placement is a challenging issue in cloud computing. However, existing relevant proposals do not consider the high-dynamicity of unikernels. For example, the survey paper [124] studies and categorizes a large number of existing VM placement approaches, but none considering unikernels. UNIC is an ideal platform to experiment with such problems, since it supports modular VM orchestration, e.g., the definition of VM placement algorithms using short code snippets through a novel graphical user interface (GUI).

In our understanding, the only relevant CDN platforms to UNIC is [125]. The MOSTO platform [125] deploys unikernels as TCP proxies (i.e., to improve TCP's slow-start algorithm performance). On the other hand, UNIC: (i) considers heterogeneity in terms of virtualization and unikernel technology; (ii) conducts real experimentation; and (iii) achieves flexibility through unikernel-oriented VM placement driven by novel early content popularity change detection.

Up to now there are only a handful of proposals addressing the challenges of new flexible networking and cloud architectures accounting for content popularity. Exceptions include [126] in which a machine learning approach to content popularity prediction is applied for a fog radio access network (RAN) environment, and, our recent papers [19] and [80]. In [19], the algorithm – outlined in [80] and presented extensively here – is integrated into an elastic content distribution network (CDN) framework based on lightweight cloud capabilities using Unikernels. [19] focuses on the platform details rather than on the CP algorithm; it confirms experimentally the suitability of the latter for relevant flexible network and cloud architectures.

3.3.3 Anomaly Detection in Software-Defined Wireless Sensor Networks

With respect to the SDWSN security, looking at existing literature in SDN anomaly detection, the authors in [127] proposed *SoftThings*, an SDN-based IoT framework with security support. The framework was developed for OpenFlow [14], which, however, can be a limiting factor for its use in networks composed of low-end nodes. The use of support vector machines (SVM) was proposed to detect control plane attacks; it was shown that a detection rate of around 96% and 98% could be achieved. The algorithm was tested in Mininet, simulating scenarios with only five nodes and considering one node as attacker. Moreover, Yin *et al.* [41] developed the framework SD-IoT, which included a security system for DDoS attacks detection, based on the difference of packets received by the controller. The difference was calculated using the *cosine similarity* method. This mechanism was devised for networks where all the nodes had periodic communication with the controller, which could be not optimal for very "restricted" networks with low-end nodes. The authors tested their proposal through simulations using Mininet. The network size was not explicitly specified, but can be inferred to be around 50 to 60 nodes.

Furthermore, Wang *et al.* [128] proposed an SDWSN trust management and routing mechanism. They compared their proposal to SDN-WISE when both networks were under attack. The focus of the work was on the selective forwarding attacks and new flow requests. The first attack applied to any type of WSNs, while the second was specific to SDNs. The mechanism was tested in simulations with 100 nodes, varying the number of attackers between 5 and 20. Their results showed an attack detection rate between 90% and 96% when 5 nodes were attackers, and between 60% and 79% when 20 nodes were attackers.

Compared to these previous works, our proposal for the employment of the RCPD in SDWSN anomaly detection [129], has the advantages of being i) lightweight, ii) fast and iii) highly accurate as will be demonstrated in Sections 3.9 to 3.9.4. In the next Section we discuss the extended RCPD and its main components.

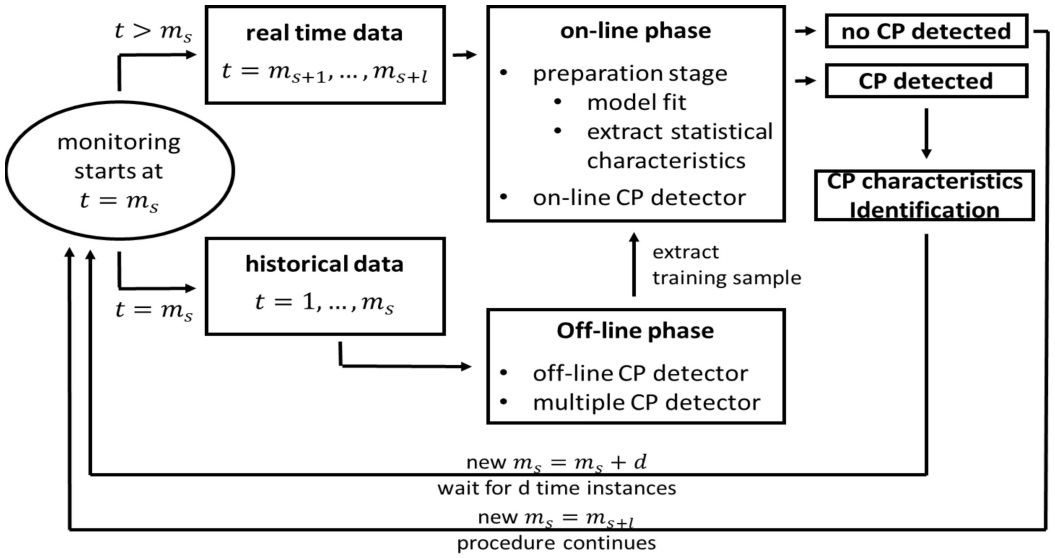


Figure 3.2: Flow diagram of the real-time variance CP detector for content views data.

3.3.4 Overview of the Proposed Integrated Algorithm

We summarize in Fig. 3.2 the overall algorithm of the extended RCPD; as a flow diagram of its individual components. Similarly to Chapter 2, we assume an arbitrary time instance m_s as the starting point of a monitoring period. Then, the off-line analysis is applied to the historical (training) data until $t = m_s$, resulting in the division of the data sequence in stable subsequences. The last subsequence is the training sample representing the initial sample of the on-line phase. During the training stage, if a parametric approach is chosen, we estimate the model parameters (e.g., ARMA or GARCH) and any other necessary statistical characteristics that describe the last stable subsequence's (time series) behavior. We note that without having first obtained a statistically robust division of the training sample into stable subsequences, the estimation of a model's parameters could be seriously impacted.

Next, an on-line detector is implemented for a monitoring period $t = m_{s+1}, ..., m_{s+l}$. If a CP is detected at cp^*_{on}, the CP magnitude on the data structure is evaluated. The new starting point for the subsequent monitoring window is then set to $m'_s = cp^*_{on} + d$, where d is a constant specifying a period assuming no change. Alternatively, if no change is detected after l instances, the procedure restarts automatically from the time point $m'_s = m_{s+l}$. The

reasons behind this choice are twofold. First, to keep the algorithm running over a window of size at most l, in order to keep the computational complexity low (lightweight), as opposed to allowing increasing window sizes. Second, to facilitate the fast responsiveness of the algorithm, as will be demonstrated through numerical examples in Section 3.6.

3.4 Off-line Phase

In this Subsection, the training phase of the algorithm is discussed and the fundamental components of the off-line scheme are presented. We choose a retrospective CP scheme to ascertain that the on-line phase is indeed carried out on homogeneous data. We note that standard off-line CP schemes can only detect a single CP. To address the issue of detection of multiple CPs, we modify the basic scheme with a novel time series segmentation heuristic, that belongs to the family of binary segmentation algorithms, similarly to [80, 82].

Let $\{X_n : n \in \mathbb{N}\}$ be a time series representing the content views, for a specific video. Since we are interested only in the variance fluctuation of the underlying random value (r.v.), we assume a constant, over time, expected value $\mathrm{E}[X_i]$, where $\mathrm{E}[\cdot]$ denotes expectation. The stability of the mean value can be ensured by a data transformation, such as taking the first differences , $\Delta_n = X_n - X_{n-1}$, thus rendering $\mathrm{E}[X_i] = 0$.

Considering the training phase, we have to check if the variance structure remains stable over the whole training period N. Consequently we study the null hypothesis,

$$H_0 : \sigma_1^2 = \cdots = \sigma_N^2, \tag{3.1}$$

where $\sigma_n^2 = \mathrm{Var}(X_n) = \mathrm{E}[X_n^2]$, given that we have modified the time series so that $\mathrm{E}[X_n] = 0$. The (general) alternative hypothesis is designed to allow the existence of multiple changes $l_i \in \{1, \ldots, N\},\ i = 1, \ldots, r$, where r is the multitude of changes,

$$\begin{aligned} H_1 : \quad & \sigma_1^2 = \ldots = \sigma_{l_1}^2 \neq \sigma_{l_1+1}^2 = \ldots = \sigma_{l_2}^2 \neq \ldots \\ & \cdots \neq \sigma_{l_{\tau-1}+1}^2 = \ldots = \sigma_{l_\tau}^2 \neq \sigma_{l_\tau+1}^2 = \ldots = \sigma_N^2. \end{aligned} \tag{3.2}$$

We develop a CP detector that only requires very general sufficient assump-

tions to be satisfied by the time series of content views. More specifically, we followed the work in [130] in which the authors introduce a non-parametric test statistic that requires only that the time series $\{X_n : n \in \mathbb{N}\}$ i) can be expressed by a possible nonlinear representation of an i.i.d. sequence $\{\epsilon_n : n \in \mathbb{N}\}$ and ii) can be approximated, with a distance measure, by an m-dependent r.v.; $\{X_n : n \in \mathbb{N}\}$ itself need not be m-dependent.

The exact form of the procedure is given in the quadratic scheme,

$$TS_N^{off} = \frac{1}{N} S_n^T \hat{\Omega}_N^{-1} S_n, \tag{3.3}$$

with $(\cdot)^T$ denoting transposition, and, it converges in distribution asymptotically to,

$$\int_0^1 B^2(n)dn, \quad (N \to \infty), \tag{3.4}$$

where $(B(n) : n \in [0,1])$ are independent standard Brownian bridges. Equation (3.4) can be used to derive the critical values (cv^{off}) of the test statistic TS_N^{off}, i.e., with Monte Carlo simulations that approximate the paths of the Brownian bridge on a fine grid. As an example, using this approach, the crossing boundaries of (3.4) for alarm rates of 5% and 1% can be found to be 1.8 and 2.6, respectively.

The detector S_n is a variation of the squared CUSUM method,

$$S_n = \frac{1}{\sqrt{N}} \left(\sum_{i=1}^{n} \text{vech}[\tilde{X}_i \tilde{X}_i^T] - \frac{n}{N} \sum_{i=1}^{N} \text{vech}[\tilde{X}_i \tilde{X}_i^T] \right), \tag{3.5}$$

where the vech(.) operator denotes the half-vectorization of a matrix (as the covariance matrix is symmetric, half-vectorization contains all the strictly necessary information) and $\tilde{X}_i = X_i - \overline{X}_N$, with $\overline{X}_N = \frac{1}{N} \sum_{j=1}^{N} X_j$ the sample average.

Since the procedure (3.3) is non-parametric, the dependence between the observations enters only in the form of the long-run covariance Ω_N, expressed as:

$$\Omega_N = \sum_{i=1}^{N} \text{Cov}(\text{vech}[X_0 X_0^T], \text{vech}[X_i X_i^T]). \tag{3.6}$$

To build a consistent estimator of Ω_N, denoted by $\hat{\Omega}_N$, various different ap-

proaches exist. This estimation problem is well studied and we focus on the kernel based approach through the use of Newey-West estimator (see [101]),

$$\widehat{\Omega}_N = \widehat{\Sigma}_0 + \sum_{w=1}^{W} k_{BT}\left(\frac{w}{W+1}\right)\left(\widehat{\Sigma}_w + \widehat{\Sigma}_w^T\right), \tag{3.7}$$

where $k_{BT}(.)$ corresponds to the Bartlett weight,

$$k_{BT}(x) = \begin{cases} 1-|x|, & \text{for } |x| \leqslant 1 \\ 0, & \text{otherwise} \end{cases}, \tag{3.8}$$

and $\widehat{\Sigma}_w$ denotes the empirical auto-covariance matrix for lag w,

$$\widehat{\Sigma}_w = \frac{1}{N}\sum_{n=w+1}^{N}\left(X_n - \overline{X}\right)\left(X_{n-w} - \overline{X}\right)^T. \tag{3.9}$$

Following common practice in literature we chose $W = \log_{10}(N)$. To summarize, the existence of a CP is announced if $TS_N^{off} > cv_V^{off}$ and the estimated time of change is,

$$cp_{off}^* = \frac{1}{N}\underset{1\leqslant n\leqslant N}{\operatorname{argmax}}\, TS_N^{off}. \tag{3.10}$$

Finally, to face the potential of detecting multiple CPs on the historical data set, we have integrated an extended version of the binary segmentation (BS) algorithm, proposed in [80], to the original test TS_N^{off}.

3.5 On-Line Methods

Next, we present three alternative on-line approaches and discuss jointly for each one: the preparation stage and the corresponding on-line CP detector. The on-line phase is based on the assumption of a homogeneous data sequence of length $m \in \mathbb{N}_+$, determined by the off-line phase, for which,

$$\sigma_1^2 = \cdots = \sigma_m^2. \tag{3.11}$$

Our aim is to test if (3.11) holds as new observations become available in a real time framework. Hence, the statistical problem is formulated as the

following hypothesis test,

$$H_0 : \sigma_1^2 = \cdots = \sigma_m^2 = \sigma_{m+1}^2 = \cdots ,$$
$$H_1 : \sigma_{m+1}^2 = \cdots = \sigma_{m+l-1}^2 \neq \sigma_{m+l}^2 = \sigma_{m+l+1}^2 \cdots , m, l \in \mathbb{N}_+.$$

In general, any on-line CP method can be described as a stopping time procedure with stopping time $\tau(m)$,

$$\tau(m) = \min\{l \in \mathbb{N} : TS^{on}(m, l) \geq b\}. \tag{3.12}$$

The value of the test statistic $TS^{on}(m, l)$ is calculated online for every l, in the monitoring period. The rule stops, and a change is announced, if the test statistic exceeds the boundary function $b = cv^{on}g$. The critical value cv^{on} is derived from the asymptotic behavior of the detector TS^{on}/g under the null hypothesis, for which $Pr\,(\tau(m) < \infty) = \alpha$, $\alpha \in (0, 1)$ the significance level. We note that γ, $\gamma \in (0, \frac{1}{2}]$ is a sensitivity parameter; the larger the value of γ, the smaller the value of the boundary function b, which leads to a quicker detection of a potential CP, at the cost of an increase in the false alarm rate.

Below, we consider three on-line CP approaches, based on the general assumptions for the underlying process: i) a non-parametric approach based on [131], denoted by NP; ii) a linear time series (ARMA) approach as in [132], denoted by L; and, iii) a nonlinear time series (GARCH) approach like in [133], denoted by NL. The quantities $\{TS^{on}, b, cv^{on}, g\}$ will be indexed accordingly.

3.5.1 Non-Parametric (NP) Approach

Non-parametric approaches work directly with the observed data and are ideal for datasets with a high degree of model fitting ambiguity. In this framework, in the preparation phase we only compute a particular form of the long-run estimator, avoiding the difficulties related to the estimation of a parametric model.

The proposed procedure is applied under the assumption that the observations $\{X_n : n \in \mathbb{Z}\}$ satisfy the generalized dependence concept of L-2 near epoch dependence (see [134]). Since the test is model-independent, the dependence between observations is captured through the long-run function D_n,

expressed as

$$D_n := \lim_{n\to\infty} \mathrm{E}\left(\frac{1}{n} A_i A_i^T\right), \tag{3.13}$$

where $A_i = \sum_{t=1}^{i} \left(X_t^2 - \mathrm{E}\left(X_t^2\right)\right)$. We also assume that D_n is finite under the H_0 hypothesis, which is necessary for the convergence of the asymptotic null behaviour.

As explained above, the long-run factor is computed in the preparation phase, considering the training sample. For its evaluation we choose the kernel estimation method, as in [135]. More specifically,

$$\hat{D}_m = \sum_{i=1}^{u} \sum_{j=1}^{u} k_{BT}\left(\frac{i-j}{r}\right) \hat{V}_i \hat{V}_j^T, \tag{3.14}$$

is an estimator of D_m, $\hat{V}_t = \frac{1}{\sqrt{m}}\left(X_t^2 - \frac{1}{m}\sum_{i=1}^{m} X_i^2\right)$ and $k_{BT}(\cdot)$ is the Bartlett kernel, already mentioned in (3.7).

The test statistic is expressed as

$$TS_{NP}^{on}(m,l) = \frac{l}{\sqrt{m}} \hat{D}_m^{-\frac{1}{2}} \left(\sum_{i=m}^{m+l} X_i^2 - \frac{1}{m}\sum_{i=1}^{m} X_i^2\right). \tag{3.15}$$

The boundary function $b_{NP} = cv_{NP}^{on} g_{NP}$ is strictly aligned with the chosen size of the monitoring period l normalized to the length of the training period, denoted by $H = l/m$. Then the weight function is expressed as $g_{NP} = \left(1 + \frac{l}{m}\right)\left(\frac{l}{m+l}\right)^{\gamma}, \gamma \in [0, 1/2)$ and the critical value is derived from the asymptotic behavior of the stopping rule,

$$\begin{aligned}
\lim_{m\to\infty} Pr\{\tau(m) < \infty\} &= \lim_{m\to\infty} Pr\left\{TS_{NP}^{on} \geq b_{NP}(\alpha)\right\} \\
&= \lim_{m\to\infty} Pr\left\{\frac{TS_{NP}^{on}}{g_{NP}} \geq c_{NP}^{on}(\alpha)\right\} \\
&= Pr\left(\sup_{n\in[0,1]} \left(\frac{H}{1+H}\right)^{\frac{1}{2}-\gamma} \frac{|W(n)|}{n^{\gamma}}\right) = \alpha.
\end{aligned} \tag{3.16}$$

3.5.2 Linear (L) Parametric Approach Using an Autoregressive Moving Average (ARMA) Model

Parametric approaches, monitor the estimated values obtained from a specific model fit to the observed time-series. This is very efficient whenever a parametric model sufficiently describes the dependence structure of the real data. We present two residual based parametric schemes, constructed from the residuals of the model fit to the data, starting with an ARMA model. In the preparation stage, the model residuals are estimated, under the assumption of a homogeneous underlying process. Under H_0, the residuals before and after the beginning of the monitoring should behave similarly. On the other hand, if a CP exists in the monitoring period, the residuals are expected to deviate from those in the training period.

ARMA processes provide linear and parsimonious descriptions of (weakly) stationary processes. A time series $\{X_n : n \in \mathbb{N}\}$ is called an ARMA(p, q) process of orders p and q, if it satisfies the stochastic equation,

$$\phi_n(B)(X_n - \mu_n) = \theta_n(B)\epsilon_n, \ n \in \mathbb{Z}, \tag{3.17}$$

where μ_n are mean parameters (usually non stationary), $\phi_n(z) = 1 - \phi_{1n}z - \cdots - \phi_{pn}z^p$ and $\theta_n(z) = 1 - \theta_{1n}z - \cdots - \theta_{qn}z^q$ are the autoregressive and moving average polynomials of the model respectively, and B the backshift operator. It is also assumed that the ARMA process is causal and invertible, i.e.,

$$\phi_n(z) \neq 0 \text{ and } \theta_n(z) \neq 0, \text{ for all } |z| \leq 1. \tag{3.18}$$

The error terms $\{\epsilon_n : n \in \mathbb{Z}\}$ are a sequence of i.i.d. r.v. with zero mean, $\mathrm{E}[\epsilon_1] = 0$ and constant variance, $\mathrm{E}[\epsilon_1^2] = \sigma^2$.

The ARMA model in (3.17) depends on $p + q + 2$ parameters, represented by the vector $\beta_n = (\mu_n, \phi_n, \theta_n, \sigma_n^2)$, where $\phi_n = (\phi_{1n}, \cdots, \phi_{pn})$ and $\theta_n = (\theta_{1n}, \cdots, \theta_{qn})$. In the defined training period of size m the parameters of the ARMA model are not time dependent, i.e., they are the same for the observations $X_1, \cdots, X_m$, denoted by β_0 in the following,

$$\beta_0 = (\mu_0, \phi_0, \theta_0, \sigma_0^2). \tag{3.19}$$

The preparation stage is applied to the training sample for two reasons. Firstly, in order to specify the order (p, q) of the corresponding ARMA model, by selecting the combination that provides the lower value for the Bayes information criterion (BIC),

$$BIC = -2\ln(\hat{L}) + k\ln(n), \tag{3.20}$$

where $\hat{L}$ is the maximum value of the likelihood function of the model, k is the number of the estimated parameters and n is the sample size. Secondly, in order to estimate the parameters β_0 of the ARMA model through the estimators $\hat{\beta}_0 = (\hat{\mu}_0, \hat{\phi}_0, \hat{\theta}_0, \hat{\sigma}_0^2)$, computed, for example, by the method of maximum likelihood estimation or least squares.

Then, the model residuals are given by

$$\hat{\epsilon}_n = \hat{X}_n - \sum_{i=1}^{p} \hat{\phi}_{i0}\hat{X}_{n-i} - \sum_{i=1}^{q} \hat{\theta}_{i0}\hat{\epsilon}_{n-i}, \tag{3.21}$$

where $\hat{X}_n = X_n - \hat{\mu}_0$. The detector is built from the (squared) residuals $\hat{\epsilon}_n$, as:

$$\frac{1}{\sqrt{m}} TS_L^{on}(m, l) = \frac{1}{\sqrt{m}\hat{\eta}_m} \left| \sum_{n=m+1}^{m+l} \hat{\epsilon}_n^2 - \sum_{n=1}^{m} \hat{\epsilon}_n^2 \right|, \tag{3.22}$$

where $\hat{\eta}_m^2$ is a weakly consistent estimator of the moment $\eta_m^2 = \mathrm{E}\left[(\epsilon_m^2 - \sigma_m^2)^2\right]$.

Finally, the boundary function is expressed as $b_L = cv_L^{on} g_L$, where $g_L = \left(1 + \frac{l}{m}\right)\left(\frac{l}{m+l}\right)^{\gamma}$, $\gamma \in [0, 1/2)$ and the critical value is obtained according to [132] as

$$\begin{aligned} \lim_{m\to\infty} Pr\{\tau(m) < \infty\} &= \lim_{m\to\infty} Pr\left\{\frac{TS_L^{on}}{g_L} \geq c_L^{on}(\alpha)\right\} \\ &= Pr\left(\sup_{n\in(0,1)} \frac{|W(n)|}{n^{\gamma}} \geq cv_L^{on}(\alpha)\right) = \alpha. \end{aligned} \tag{3.23}$$

3.5.3 Nonlinear (NL) Parametric Approach Using a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Model

A time series $\{X_n : n \in \mathbb{Z}\}$ follows the GARCH(p, q) process, if,

$$\begin{aligned} X_n &= \sigma_n \epsilon_n, \\ \sigma_n^2 &= \omega_n + \sum_{i=1}^{q} \alpha_{in} X_{n-i}^2 + \sum_{j=1}^{p} \beta_{jn} \sigma_{n-j}^2, \end{aligned}$$

where $\omega_n > 0$, $\alpha_{in},\ \beta_{jn} \geqslant 0$ and $\{\epsilon_n : n \in \mathbb{Z}\}$ is a sequence of i.i.d. r.v. with $\mathrm{E}[\epsilon_1] = 0$ and $\mathrm{E}[\epsilon_1^2] = 1$. We estimate the set of parameters θ_m during the initial training period, denoted in the following by $\theta_0 = (\omega_0, \alpha_{10,}, \cdots, \alpha_{q0}, \beta_{10}, \cdots, \beta_{p0})$; the estimation is performed by applying the quasi maximum-likelihood estimator (QMLE) $\hat{\theta}_0$ of θ_0 on the last m observations, as proposed in [136]. The Gaussian quasi-likelihood function is given by

$$L_m(\theta; X_1, \cdots, X_m) = \prod_{n=1}^{m} \frac{1}{\sqrt{2\pi\hat{\sigma}_n^2}} \exp\left(-\frac{X_n^2}{2\hat{\sigma}_n^2}\right), \tag{3.24}$$

where $\hat{\sigma}_n^2$ are constructed recursively, as,

$$\hat{\sigma}_n^2 = \omega + \sum_{i=1}^{q} \alpha_{in} X_{n-i}^2 + \sum_{j=1}^{p} \beta_{jn} \hat{\sigma}_{n-j}^2. \tag{3.25}$$

Then, the QMLE of θ_m is,

$$\begin{aligned} \hat{\theta}_m &= \underset{\theta \in \Theta}{\operatorname{argmax}}\, L_m(\theta; X_1, \cdots, X_m) = \\ &= \underset{\theta \in \Theta}{\operatorname{argmin}}\, \frac{1}{m} \sum_{n=1}^{m} \left(\frac{X_n^2}{\hat{\sigma}_n^2} + \ln(\hat{\sigma}_n^2)\right). \end{aligned} \tag{3.26}$$

The residuals of the GARCH process are subsequently obtained from the QMLE as:

$$\hat{\epsilon}_n = \frac{X_n}{\hat{\sigma}_n(\hat{\theta}_m)}. \tag{3.27}$$

Based on the (squared) residuals, the test statistic is described as in [137],

$$TS_{NL}^{on}(m,l) = \sqrt{\frac{m}{\mathrm{Var}(\hat{\epsilon}_m^2)}}\left|\frac{1}{l}\sum_{n=1}^{l}\hat{\epsilon}_n^2 - \frac{1}{m}\sum_{n=1}^{m}\hat{\epsilon}_n^2\right|, \tag{3.28}$$

where $\mathrm{Var}(\hat{\epsilon}_n^2)$ denotes the variance of the squared residuals of the training period, i.e., $\mathrm{Var}(\hat{\epsilon}_m^2) = \mathrm{E}[\hat{\epsilon}_m^4] - (\mathrm{E}[\hat{\epsilon}_m^2])^2$.

Considering the boundary function $b_{NL} = cv_{NL}^{on} g_{NL}$, we choose to work with $g_{NL} = 1$ as in [133]; consequently, the critical value is given by

$$\begin{aligned}\lim_{m\to\infty} Pr\{\tau(m) < \infty\} &= \lim_{m\to\infty} Pr\left\{TS_{NL}^{on} \geq cv_{NL}^{on}(\alpha)\right\} \\ &= Pr\left(\sup_{n\in(0,1)} |W(n)| \geq cv_{NL}^{on}(\alpha)\right) = \alpha. \end{aligned} \tag{3.29}$$

3.5.4 Evaluation of Critical Values for CPs Tests

The on-line critical values for the three procedures are estimated using Monte Carlo simulations, similarly to the off-line case, considering that:

$$cv_{NP}^{on}(\alpha) = \sup_{n\in[0,1]}\left(\frac{H}{1+H}\right)^{\frac{1}{2}-\gamma}\frac{|W(n)|}{n^{\gamma}}, \tag{3.30}$$

$$cv_{L}^{on}(\alpha) = \sup_{n\in(0,1)}\frac{|W(n)|}{n^{\gamma}}, \tag{3.31}$$

$$cv_{NL}^{on}(\alpha) = \sup_{n\in(0,1)}|W(n)|. \tag{3.32}$$

With respect to the estimation of the magnitude of a detected CP denoted by cp_{on}^*, in the NP scenario, we estimate the deviation of the variance in pre-CP and post-CP data by comparing the variance of a pre-determined historical subsample, $(X_{m_s} : X_{cp_{on}^*-h})$ to the variance "in the range" of the detected CP as $(X_{cp_{on}^*-h} : X_{cp_{on}^*+h})$, accounting for the fact that a time lag $\pm h$ is required to establish the presence of an actual change.

We finally propose an alternative scheme to predict the post CP behavior in the case of a parametric model. We apply the parametric models (ARMA or GARCH) on the time horizon $t_{cp_{on}^*-h}, \ldots, t_{cp_{on}^*}$, in which we assume that the actual change has already occurred. Thus, a well defined subsample is provided to fit the model parameters and predict the next values of the model.

3.6 Performance Evaluation of the Variance CP Detection Approaches on Synthetic Data

In this Subsection, we evaluate the performance of the integrated algorithm with the three aforementioned variations of the on-line phase – NP, L and NL, – on two sets of synthetic data. In further detail, we report the results of Monte Carlo simulations using either an ARMA(1,1) or a GARCH(1,1) process to generate the time series; as a reminder, both ARMA and GARCH are well known models that have been shown to fit well video content popularity dynamics (see Subsection 3.3.1).

The synthetic sample size under consideration is $N = 1000$ while we introduce a variance CP at $cp^* = 500$; this is achieved by transforming the initial parameters vector of the chosen model. Evaluations are conducted based on 1000 repetitions for a significance level $\alpha = 0.01$. In all tests we set the beginning of the monitoring period at $m_s = 200$, the monitoring window length at $l = 100$ and the minimum interval between two successive CPs at $d = 80$ (this latter choice is justified by experiments with real data that will be presented in Section 3.7). We experiment with two values for the sensitivity parameter $\gamma \in \{0, 0.25\}$ (as a reminder, γ only affects cv_{NP}^{on} and cv_{L}^{on}, see (3.30) and (3.31)).

We first evaluate the performance of the three alternative on-line procedures in the integrated algorithm, for a wide range of ARMA(1,1) models. We recall that the variance of an ARMA(1,1) model depends on the model parameters $\phi_i,\ \theta_i$ and the variance of the error terms σ_i^2, i.e.,

$$\mathrm{Var}(X_n) = \frac{(1 + 2\phi_i\theta_i + \theta_i^2)\,\sigma_i^2}{1 - \phi_i^2}.$$

We consider a change by transforming the time series model defined by the parameter vector β_0 to one of the vectors $\beta_i,\ i = 1, 2, 3, 4$.

- Model 0: $\beta_0 = (\phi_0, \theta_0, \sigma_0) = (0.4, 0.2, 0.5)$,
- Model 1: $\beta_1 = (0.4, 0.2, 1)$,

- Model 2: $\beta_2 = (0.3, 0.3, 1.5)$,
- Model 3: $\beta_3 = (0.5, 0.3, 1.5)$,
- Model 4: $\beta_4 = (0.4, 0.2, 2)$.

We use Model 0 as the baseline. In Model 1 a small change in the error variance is introduced, which increases the uncertainty. Models 2 and 3 lead to medium changes in the variance and also transform the dependence structure between the r.v.. On the other hand in Model 4 a large change is introduced by increasing the uncertainty.

In Table 3.1 we report the results of the simulation study. We depict the aggregate percentage of the CPs over the multitude of the simulations. For every test and each iteration we calculate the exact number of CPs detected:

- 0 when no CPs are detected, denoting the percentage of false negatives in all cases but the first (in which β_0 does not change); in this latter case it corresponds to the true success rate;
- 1 when a single CP is detected, denoting the true success rate in all cases but the first, in which it corresponds to a false positive rate;
- > 1 when more than one CPs are detected, denoting the percentage of false positives, in all cases other than the first. To obtain the overall false positive percentage, this value needs to be added to the false positive percentage above.

Furthermore, we denote by $\hat{cp}^*$ the median of the time instance of the identification of the true CP, evaluated in all cases but the first. The closest this number to the true point of the CP at 500, the quicker the detection and the better the responsiveness of the integrated algorithm.

Table 3.1: Results from an ARMA generating process and for one change in the variance

		ARMA(1,1)											
		Non parametric approach (NP)				ARMA approach (L)				GARCH approach (NL)			
		Detected CPs			$\hat{cp}^*$	Detected CPs			$\hat{cp}^*$	Detected CPs			$\hat{cp}^*$
β	γ	0	1	> 1	med	0	1	> 1	med	0	1	> 1	med
β_0	0	**0.99**	0.01	0	-	**0.99**	0.01	0	-	**0.98**	0.02	0	-
	0.25	**0.95**	0.05	0	-	**0.98**	0.02	0	-				
β_1	0	0.49	**0.5**	0.01	-	0.48	**0.52**	0	554	0.74	**0.26**	0	-
	0.25	0.18	**0.76**	0.06	549	0.07	**0.93**	0	548				
β_2	0	0.04	**0.94**	0.02	550	0.03	**0.95**	0.02	546	0.15	**0.83**	0.02	549
	0.25	0	**0.93**	0.07	531	0	**0.96**	0.04	521				
β_3	0	0.01	**0.96**	0.03	536	0.01	**0.98**	0.01	535	0	**0.97**	0.03	548
	0.25	0	**0.92**	0.08	521	0	**0.97**	0.03	521				
β_4	0	0	**0.97**	0.03	533	0	**0.99**	0.01	530	0.01	**0.97**	0.02	544
	0.25	0	**0.93**	0.07	519	0	**0.97**	0.03	513				

Initially, we discuss the impact of the choice of the sensitivity parameter γ in the L and NP approaches. Studying Table 3.1, we conclude that $\gamma = 0$ is the most reasonable choice in the case of medium or more significant changes in the variance, since it leads to significantly lower false positive rates. On the other hand, in the case of only small changes in the variance, captured in our study in the transformation from the β_0 to the β_1 model, a higher value of γ is needed (intuitively, for smaller changes a larger sensitivity is required). Therefore, depending on whether smaller or larger deviations need to be rapidly detected we can fine-tune the value of γ. For the sake of simplicity, in the following we focus on $\gamma = 0$ (larger deviations).

According to Table 3.1, the three approaches provide appropriate empirical sizes, and the false alarm rates are in all cases close to the significance level $\alpha = 0.01$. Overall the L procedure outperforms the NP and the NL, both in terms of the true alarm rates as well as in terms of the detection time; this is intuitive as in this first experiment the underlying process is generated by a linear ARMA(1,1) model and therefore a linear parametric model is excellently suited to capture the underlying dynamics. Furthermore, comparing the NP and the NL approaches, Table 3.1 illustrates that the NP is more sensitive than the NL approach, leading to more accurate detection for small changes at the cost of increased false positive rates in the case of larger changes. The opposite is true for the NL approach that appears to be more "conservative". Moreover, the fact that the NP procedure is statistically more sensitive leads to a quicker detection of a CP as captured through $\hat{cp}^*$.

We proceed to the more challenging case of a GARCH(1,1) generating model, with parameter vector $\theta_i = (\omega_i, \alpha_i, \beta_i)$ that fully describes the model and unconditional variance,

$$\text{Var}(X_n) = \frac{\omega_i}{(1 - \alpha_i - \beta_i)}.$$

To examine the alarm rates we assume the following models,

- Model 0: $\theta_0 = (\omega_0, \alpha_0, \beta_0) = (0.05, 0.4, 0.3)$,
- Model 1: $\theta_1 = (0.5, 0.2, 0.1)$,
- Model 2: $\theta_2 = (0.5, 0.3, 0.2)$,

- Model 3: $\theta_3 = (1, 0.3, 0.2)$.

GARCH is a varying volatility model, allowing volatility changes over time. Being more elaborate and complex in terms of the dependence of the variance on the model parameters, the higher false alarm and the lower true alarm rates in Table 3.2 are reasonable. In this case, the L procedure seems fully inappropriate irrespective of the choice of $\gamma = 0$ or $\gamma = 0.25$, suffering from very high false positive rates, since constant variance is assumed. The NL procedure, as expected, surpasses both the L and the NP procedures, as it is excellently suited to capture the GARCH process. More specifically, the true alarm rate estimation is stable for the different magnitudes of changes, with a detection time lag ranging from 50 instances for small changes to 31 instances for larger changes. On the other hand, the NP procedure appears to capture well the actual changes for $\gamma = 0$, with success rates relatively close to the those of the NL procedure, especially for medium / large changes. However, for $\gamma = 0.25$, the approach leads to ineligible false positive rates, despite the fact that it can identify small changes more efficiently. The NP method also achieves faster detection of changes, with $\hat{cp}^*$ ranging from 5 to 28 time instances.

Based on the analysis of the Monte Carlo results for the three procedures under the two different time series generating models, we can synthesize our overall conclusions in the following two points:

1. The NL and the NP approaches adapt better to a wider range of models and underlying assumptions; if there are indications of a highly nonlinear underlying procedure the NP approach could render better results;

2. The L approach is strongly related to the ARMA model assumptions and therefore it is advisable to be applied only if these can be readily shown to hold.

Table 3.2: Results from a GARCH generating process and for one change in the variance

		GARCH(1,1)											
		non parametric approach (NP)				ARMA approach (L)				GARCH approach (NL)			
θ	γ	Detected CPs			$\hat{cp}^*$	Detected CPs			$\hat{cp}^*$	Detected CPs			$\hat{cp}^*$
		0	1	> 1	med	0	1	> 1	med	0	1	> 1	med
θ_0	0	**0.85**	0.15	0	-	**0.75**	0.25	0	-	**0.9**	0.1	0	-
	0.25	**0.65**	0.35	0	-	**0.42**	0.58	0	-				
θ_1	0	0.16	**0.8**	0.04	527	0.03	**0.77**	0.23	528	0.04	**0.92**	0.04	550
	0.25	0	**0.87**	0.13	521	0	**0.6**	0.4	515				
θ_2	0	0.03	**0.87**	0.1	524	0.01	**0.76**	0.23	521	0.01	**0.93**	0.06	544
	0.25	0.01	**0.85**	0.14	516	0	**0.56**	0.44	510				
θ_3	0	0	**0.93**	0.07	511	0	**0.7**	0.3	511	0	**0.93**	0.07	531
	0.25	0	**0.81**	0.19	508	0	**0.58**	0.42	505				

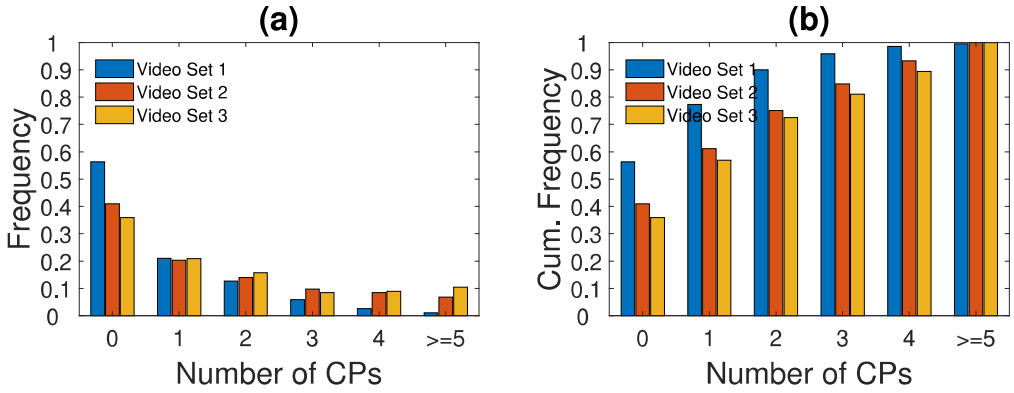


Figure 3.3: Estimated a) frequency and b) cumulative frequency of the number of CPs per time series, for three different Video Sets.

3.7 Illustration of the Integrated Algorithm Using Real Data

Finally, we study the performance of the proposed algorithms on monitoring real YouTube video traces provided within the framework of the CONGAS project [109]; the dataset consists of 882 videos traces and the observation period is of $N = 1000$ time instances.

In this Section, we only adopt the non parametric (NP) and the GARCH (NL) approaches. We exclude the ARMA (L) approach from the evaluation, based on the conclusions of the previous Section. We work with the centered simple returns of the content popularity time series,

$$Y_n = (X_{n+1} - X_n) - \frac{1}{900}\sum_{n=1}^{900}(X_{n+1} - X_n),\ n = 1, \ldots, 900$$

and then apply the methods on Y_n.

In order to clarify some general characteristics of the dataset, in terms of changing content dynamics, we first apply the off-line algorithm to the video traces. In Fig. 3.3, we consider three video sets; Video Set 1 contains the whole dataset, Video Set 2 contains the videos with average number of visits $\mathrm{E}\left((Y(1) : Y(1000)\right) \geq 10$ and Video Set 3 contains the videos with average number of visits greater or equal to 20. Fig. 3.3, depicts a high percentage of rejecting the H_0 hypothesis, for a significance level of $\alpha = 0.05$. Especially for

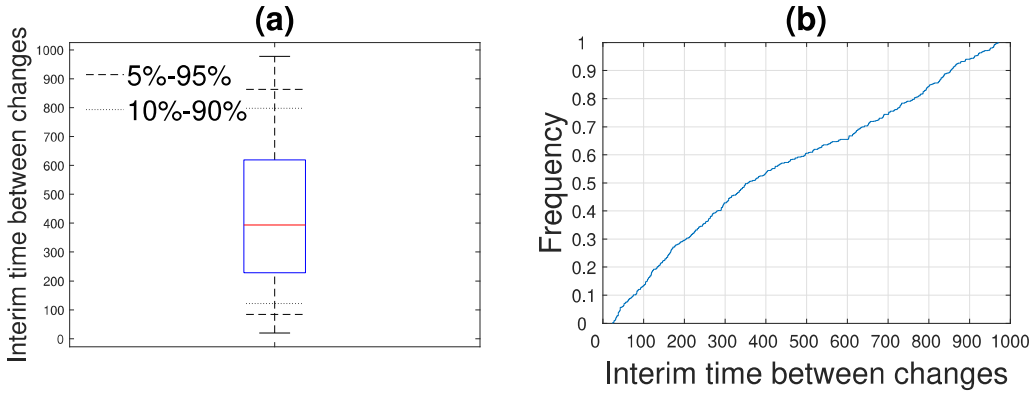


Figure 3.4: Interim time between consecutive CPs: a) Boxplot including the interval $(5\% - 95\%)$ (dashed line) and $(10\% - 90\%)$ interval (dotted line), b) Cumulative frequency for the interim time of consecutive CPs.

the Video Sets 1 and 2, the rejection of the assumption of normal behavior exceeds 60% and 65% of the time series, respectively. This result confirms that a sufficiently high number of time series provide content popularity anomalies, for example in Video Set 3, in 10% of the cases there are over than four CPs per time series. This small analysis confirms the suitability of change point analysis as a viable approach for the detection of changes in video content popularity dynamics.

Subsequently, in Fig. 3.4, we analyze the interim time between consecutive CPs. The respective boxplot diagrams illustrate the existence of sufficiently large intervals between consecutive changes; this fact supports our subtle assumption in Section 3.4 regarding the existence of a sufficient gap between two consecutive CPs (e.g., > 80 instances). In particular, 90% and 95% of the intervals correspond to consecutive CPs exceeding 100 and 80 time instances, respectively. This outcome assures that a sufficiently large training window after a detected change can be applied, denoted by the parameter d.

Additionally, Fig. 3.5, illustrates the time instances of upward (increase in volatility) and downward changes (decrease in volatility) in the form of a boxplot. It is shown that upward changes occur earlier in time than downward changes.

We consider now the performance of the on-line approach, by illustrating the estimated CPs in the second order characteristics of different time series. We choose the beginning of the monitoring period at $m_s = 200$, the sensitivity

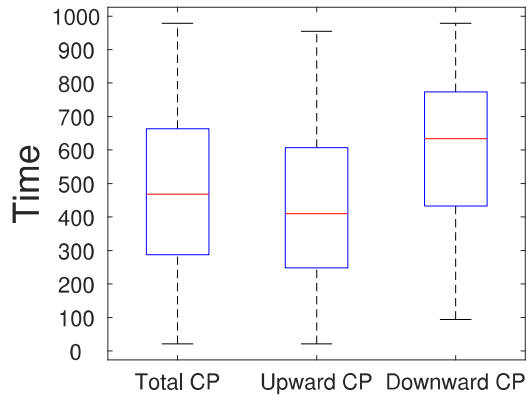
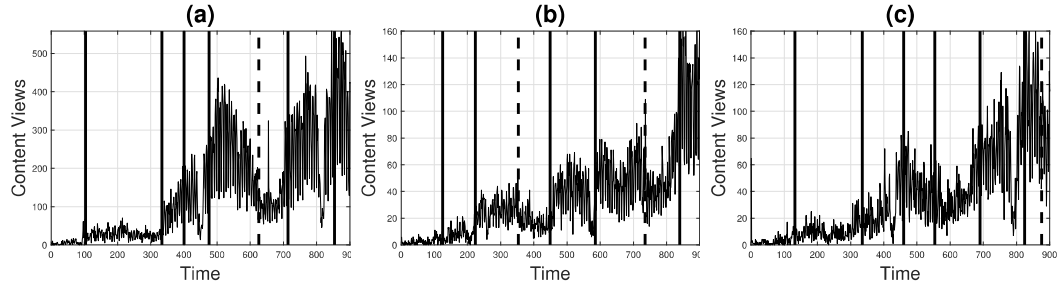


Figure 3.5: Boxplot of the number of upward and downward CPs, per time series.

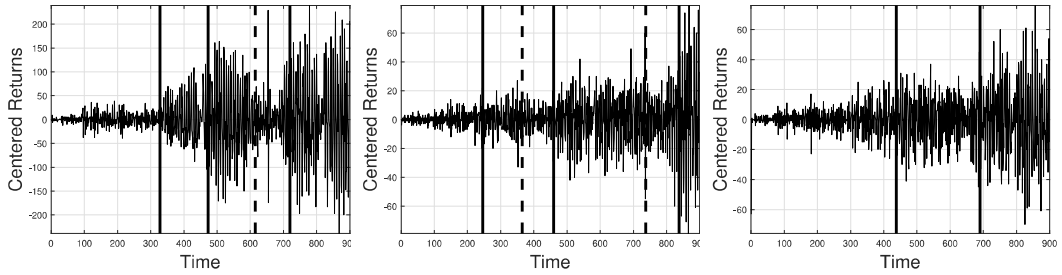
parameter $\gamma = 0$ and the significance level $\alpha = 0.05$. To fit a GARCH(p, q) model we consider all the possible combinations of the $p, q = 1, \cdots, 4$ and choose the orders p, q that minimize the Akaike information criterion (AIC).

The corresponding results are depicted in Fig. 3.6 in the previous page. The first row of results represent the detected changes in the mean value by using the RCPD algorithm presented in [80]. In the second and third row the estimated CPs in the variance are depicted, for the same time series, by applying on the first order differences Y_n the non parametric (NP) approach and the GARCH (NL) approach, respectively. Solid lines represent upwards changes while dashed lines represent downward changes.

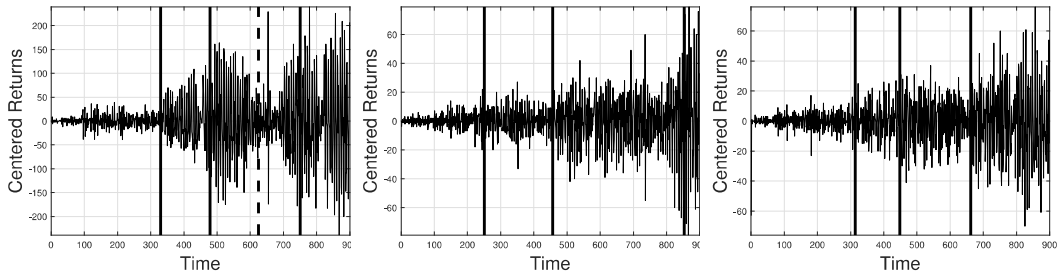
Firstly, we observe that the variance changes are closely connected to a corresponding mean change. In particular, variance changes are less in multitude and seem to be related to the most significant mean changes, which can be intuitively explained by considering that if the average number of views changes significantly, the variance in the number of views at the respective interval will follow a similar trend. The importance of jointly studying the changes in the mean and the variance value is also depicted in Fig. 3.6. For instance, in Fig. 3.6a, to describe or handle the content popularity dynamics it is crucial to estimate quickly the"explosion" in variance after time instances 500 or 700, that leads to a high instability of the values from the mean. On the other hand, variance "reduction" detection is also important, as it implies that values remain relatively constant, like in Fig. 3.6a between time instances 600 and



CPs detected in the mean for three different time series.



CPs detected in the variance of the corresponding centered returns, applying the non parametric process.



Corresponding outputs, applying the GARCH process.

Figure 3.6: CPs detected in the mean (first row) and variance (second and third row) for three different content views time series. Solid and dashed lines represent an upward and a downward change, respectively.

700.

Both the NP and the NL approaches provide similar results in terms of the number of CPs and the detection time of the estimated CPs. More precisely, in Fig. 3.6a, both procedures detect the same number of changes, while the NP method gives a slightly quicker detection.

Focusing on the capability of the proposed algorithm to estimate the magnitude of a detected CP, we use the GARCH model. We estimate the parameters of the model considering 10 time instances before the detected change and forecast the variance for 10 time instances after the CP. For the time series in Fig. 3.6b, the actual variance after each change is 7.92, 12.51 and 38.66, while the predicted variance values are 7.39, 13.52 and 39.24, respectively. As we observe, in this case the NL algorithm can efficiently describe the post change variance behavior.

In the future, we will develop a joint approach identifying CPs simultaneously in the first and the second order characteristics, providing an aggregated and compact view of content popularity dynamics.

3.8 Application of RCPD in a Next-generation CDN Platform

In this Subsection, we propose an elastic content distribution platform that serves the Internet content using tiny unikernel-based VMs. Such VMs are hosting one or a few videos each, appear rapidly in nearby cloud deployments, serve users and then disappear. In other words, the studied environment provides content dissemination through very dynamic, almost "fluid" VM placement, since the content is packaged with the server software with just a minor increase in size. So, we reposition the content caching and provisioning as a VM orchestration problem. We demonstrate the complete implementation of the proposed platform with experimental results.

3.8.1 The UNIC Platform Architecture

UNIC is an intelligent lightweight cloud orchestration platform providing efficient content distribution to the end-users through unikernel-based VMs

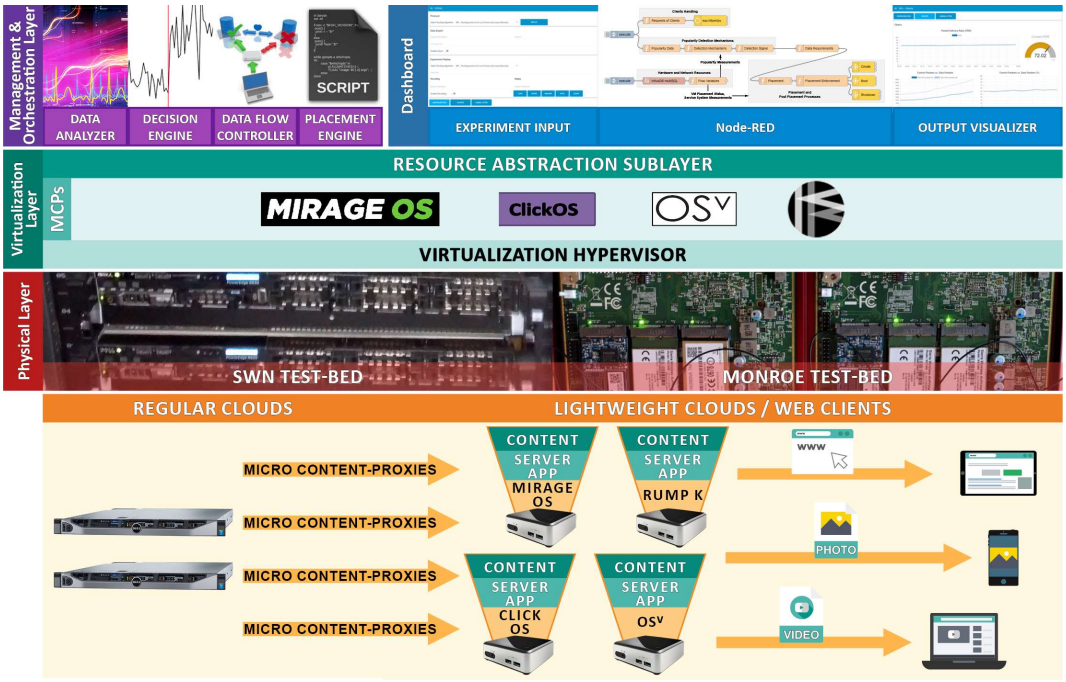


Figure 3.7: The Architecture of the UNIC platform

scattered to a cloud hierarchy (i.e., with both regular and lightweight clouds). The UNIC platform realizes flexible and scalable content distribution over heterogeneous virtual and physical resources. We focus on two main UNIC aspects here: (i) the modular VM placement algorithms considering real time server resource utilization and content provisioning requirements; and (ii) the ability of RCPD to drive VM orchestration.

Here, we give a bottom-up description of the UNIC platform architecture (i.e., Figure 3.7), which consists of the following three main layers:

a) The *Physical Layer* utilizes hardware providing the required functionality. More precisely, we use our own SWN test-bed [3]. SWN test-bed provides the regular and lightweight cloud facilities as well as the end-users emulation.

b) The *Virtualization Layer* supports lightweight cloud capabilities and multiple unikernel technologies (e.g., Mirage OS, Rump Kernel or Click OS). UNIC architecture is independent from the virtualization technologies used; hence, carefully designed abstractions hide the virtualization heterogeneity (i.e., the *Resource Abstraction Sublayer* exports a uniform interface for VM control).

c) The *Management and Orchestration Layer* controls and orchestrates the

[3] http://emulab.swn.uom.gr

UNIC platform and test-beds, including providing the efficient VM placement through the *Placement Engine*, the traffic control through the *Data Flow Controller* and network analytics enabling intelligent network configuration decisions through the *Data Analyzer* and the *Decision Engine*, respectively.

As shown in Figure 3.7, the *UNIC dashboard* provides the experimentation input, results visualization and modular extensibility of the evaluated mechanisms through the Node-RED tool [138].

An important UNIC aspect is the ability to implement VM orchestration processes in the form of Node-RED work-flows, in a plug-and-play fashion. All the UNIC *Management and Orchestration Layer* components have been implemented in the form of Node-RED nodes, such as: (i) the content popularity detection for the *Decision Engine*; (ii) the VM placement functions for the *Placement Engine*; (iii) the content popularity and web client performance monitoring for the *Network Analytics*; and (iv) the traffic load balancing (i.e., our own dynamic DNS server matching content replicas with web clients) for the *Data Flow Controller*. These nodes are standalone components that can be manipulated / configured independently of each other and be connected to form complete VM orchestration processes.

To further analyze the UNIC platform, we describe two of its core features below, i.e., the content popularity changes detection and the modular VM placement mechanisms.

The UNIC platform detects early changes in the content popularity and signals new unikernel placements, in case of an upward qualitative change in the content popularity or a removal of unikernels in case of a downward change. We maintain two parallel change-point detection processes with different significance level α and parameter γ values. We place one more unikernel, in case the change is detected from both processes, otherwise we may remove one or two, in a similar way. The content VM placements and removals are being handled from the RCPD algorithm.

As discussed above, UNIC supports modular extensibility of VM orchestration functionalities in the form of independent software entities, called Node-RED nodes. We exploited the capability to implement the *Objective Weight Function (OWF)* VM placement mechanism. We indicate with cpu_{np_i}, mem_{np_i}, tt_{np_i} and rt_{np_i} the real-time resource utilization of the CPU, RAM,

theoritical capability of the network throughput for incoming (RT) and outgoing (TT) interfaces, respectively, for the i-th node pn_i. The OWF runs periodically and place the VM on the one pn_i - of the set of the $\mathcal{PN}$ physical nodes - that provides the minimum resource utilization:

$$\min_{pn_i \in \mathcal{PN}} \alpha cpu_{pn_i} + \beta mem_{pn_i} + \gamma TT_{pn_i} + \delta RT_{pn_i} \tag{3.33}$$

$$\text{s.t.} \quad 0 \leq cpu_{pn_i}, mem_{pn_i}, tt_{pn_i}, rt_{pn_i} < 1 \tag{3.34}$$

The coefficients $\alpha,\ \beta,\ \gamma,\ \delta \geq 0$ weight the importance of each resource type, e.g., to match particular application-level requirements.

3.8.2 Experimental Methodology

We conducted real experiments that required the implementation and configuration of separate technical features, such as: (i) the VM orchestration; (ii) the content popularity detection mechanisms; (iii) the end-user traffic emulation and control; and (iv) the physical server resource utilization and end-user performance monitoring. We briefly outline each technical aspect below, i.e., configuration parameters, basic implementation details or open-source tools we used.

The VM Orchestration: We created lightweight web-servers delivering content with Mirage OS[4] unikernels. Such tiny VMs are being orchestrated according to a work-flow diagram created through the Node-RED tool. Such work-flow defines the communication of independent software entities, i.e., the Node-RED nodes. We created one node per VM orchestration process (e.g., the VM deployment, the placement decision making, etc). An experimenter can introduce new nodes (e.g., placement algorithms) or connect them in alternative ways (e.g., to create new orchestration work-flows). All processes communicate with the hypervisor through the *Resource Abstraction Sublayer (RAS)*, exposing a unified north interface to the orchestration features but virtualization-technology specific south interfaces. The latter communicate through ansible scripts [139]. *RAS* allows us to introduce heterogeneous

[4]https://mirage.io/

virtualization technologies, in the near future.

The Content Popularity Detection Mechanism: We implemented the RCPD mechanisms in Matlab. To ensure an online operation, we created at a separate host a Matlab TCP server application that receives periodic content popularity measurements and returns a notification for each detected change-point and an estimation of its basic characteristics (i.e., direction and rough magnitude). The other end is a Node-RED node that triggers VM deployments or removals.

The end-user traffic emulation and control: We emulated the web-clients using the httperf open-source tool [140]. We create and deploy web-clients based on real content popularity measurements extracted from youtube. In Figure 3.8, we show the content requests per minute we used as an input for the emulation of web clients in these experiments. The duration of the particular measurements match the duration of the experiments, i.e., 310 minutes for each run. We created a DNS-based load-balancing Node-RED node that keeps track and redirects the web requests to particular content caches (i.e., unikernel VMs), in a round-robin fashion.

The physical server resource utilization and end-user performance monitoring: We are monitoring the servers' resource utilization, in terms of CPU, memory, incoming / outgoing traffic and the performance of web-clients using the open-source tool CollectD [141]. We store the measurements in InfluxDB [142], a time-series database, and visualize them with the Grafana tool [143]. The measurements reach to the orchestration processes that take informed decisions for the context environment, e.g., to the VM placement algorithms. The monitoring takes place at regular time intervals (i.e., every 10 secs).

We use the SWN test-bed to conduct the experiments. More precisely we used: (i) seven physical servers, five to host the VMs, one as a management and orchestration server and one to host the CP mechanisms; (ii) ten Raspberry pis to emulate the web-clients requesting web content from the VMs; and (iii) one L3 100 Mbps switch.

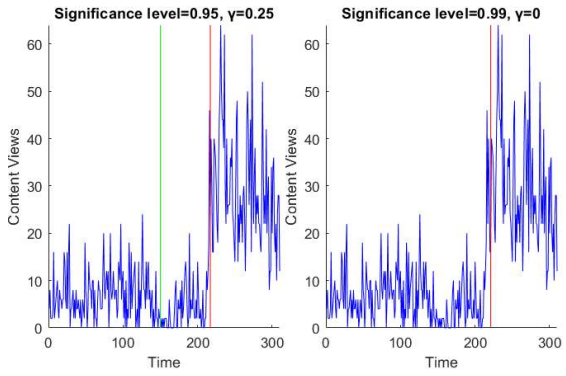


Figure 3.8: Content-views per minute of a particular youtube video and detected change-points for different α and γ values. Red lines denote the upward and downward change.

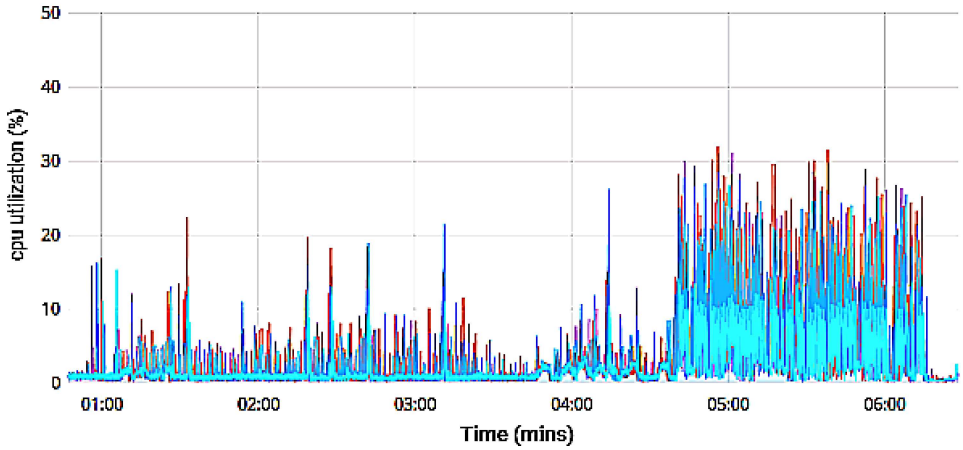


Figure 3.9: The servers' CPU utilization with the change-point detection mechanisms disabled.

3.8.3 Experimental Results

We validate the impact of the change-point detection mechanism in terms of physical servers' resource utilization, carrying out two separate experiments. For the first experimental scenarios we assume a running operation of UNIC with three VMs hosting particular web content and we allocate web-clients based on the real content popularity traces illustrated in Fig. 3.8.

We apply two CP processes in parallel with variable sensitivity, i.e., a more sensitive with parameters $\gamma = 0.25$, $\alpha = 0.95$ and a less sensitive with parameters $\gamma = 0$, $\alpha = 0.99$, as shown in Figure 3.8. In case both of them

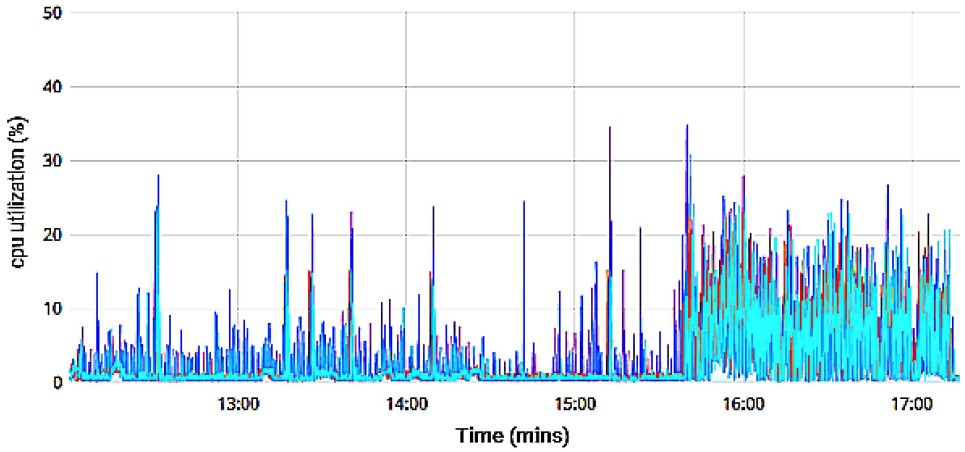


Figure 3.10: The servers' CPU utilization with the change-point detection mechanisms enabled.

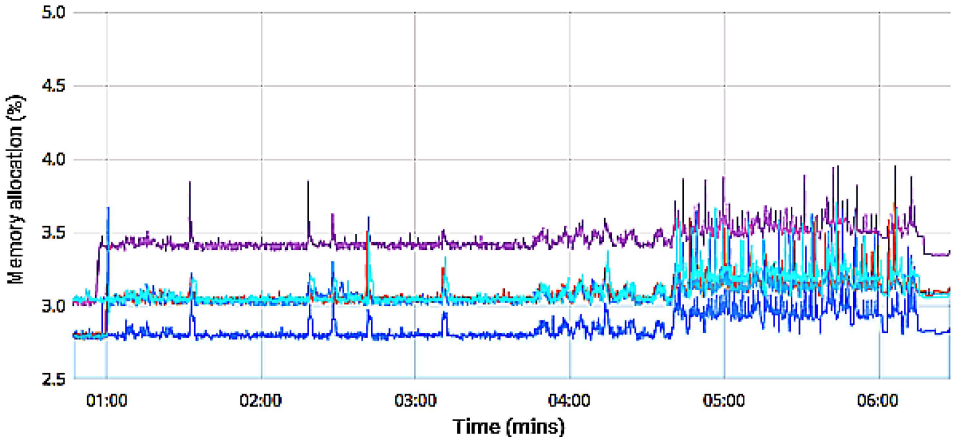


Figure 3.11: The servers' memory allocation with the change-point detection mechanisms disabled.

estimate a change point at the same time interval, we deploy two VMs, assuming a larger magnitude of change. In the typical case the sensitive approach detects a change but not the less sensitive one, we deploy one VM.

We set the *OWF* algorithm's coefficients $\alpha,\ \beta,\ \gamma,\ \delta \geq 0$ to the values 60%, 30%, 5%, 5%, respectively. In the following Figs. (i.e., 3.9 to 3.12), the different colors represent measurements from different physical machines.

Figs. 3.9 and 3.10 compare the percentage of CPU utilization and Figures 3.11 and 3.12 the percentage of memory allocation per physical server, with the change-point detection mechanisms disabled and enabled, respectively. According to these experimental results, we have the following observations:

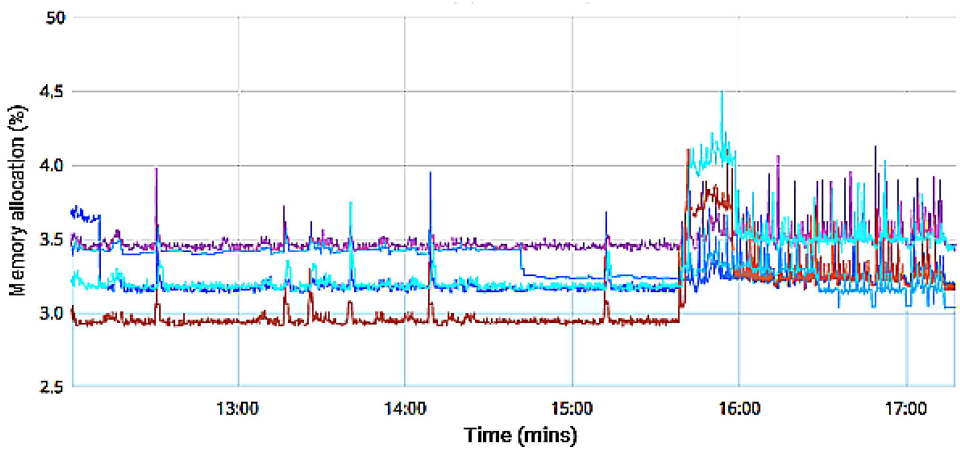


Figure 3.12: The servers' memory allocation with the change-point detection mechanisms enabled.

- For the first time period (i.e., 0 to 220 minutes), the CPU and the memory allocation is similar for both cases;
- For the second time period (i.e., 220 to 310 minutes), the maximum CPU utilization was reduced around 10%, while there is a 0.5% increase in the memory allocation.

These outcomes can be explained as follows. In the second experimental run, the CP detection mechanisms take the decision to boot two more VMs due to the abrupt increase of content views (i.e., both CP processes detect the change, as shown in Fig. 3.8). This decision reduces the CPU utilization, but has a minor impact on the memory allocation (i.e., due to the additional VM deployment). This is consistent with the content popularity traces used for the web-clients deployment, dictated by the real measurements, where there is a change-point at the same time-period (i.e., see Fig. 3.8). We note the smaller change-point, detected from the sensitive CP process only, leads to the removal of a VM, without significant impact on both CPU utilization and memory.

In the second experiment, the number of clients at each time instance is based on a real time-series of YouTube content views, illustrated in Fig. 3.13a. In practice, an experimental run without the RCPD mechanisms uses three content caches constantly and a run with the RCPD mechanism enabled uses initially two and then three, four and five content caches, after each of the three detected change points, respectively. As we show in Fig. 3.13c, the

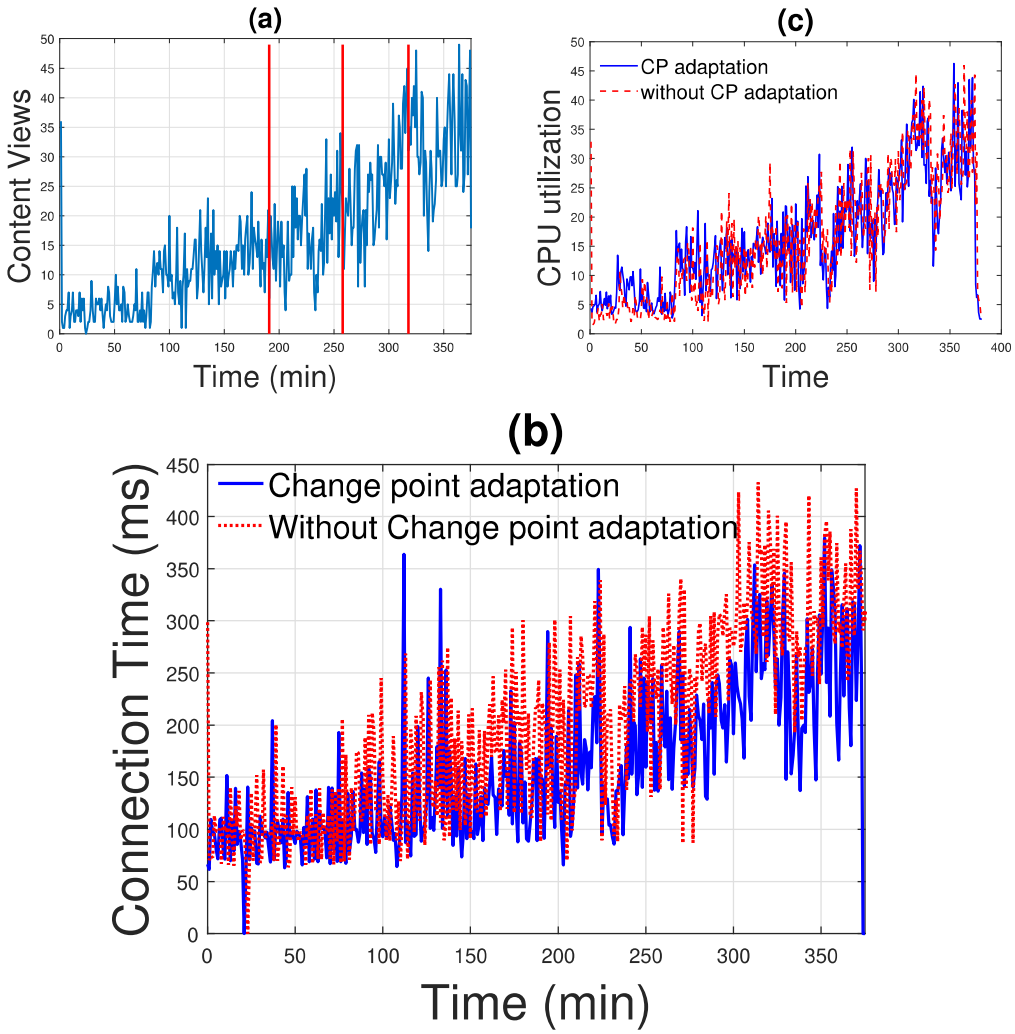


Figure 3.13: a) Time-series of video content views, red lines depict the detected CPs, b) The connection time with and without RCPD adaptation and c) The equivalent servers' CPU utilization.

web clients improve their connectivity times to download the content, while as demonstrated in Fig. 3.13b the CPU utilization in the servers hosting the content remains almost the same.

3.9 Application of the RCPD for Intrusion Detection in SDWSNs

Considering the limitations of previous works in SDWSN anomaly detection, outlined in Section 3.3, our main objective is to propose in the remainder of

this Chapter a mechanism for DDoS detection with, i) a high detection rate, and, ii) low complexity, so that it would be suitable for "restricted" networks. To this end, we propose the employment of the RCPD. As will be explained in detail next, we study two different DDoS attacks: FDFF attack and FNI attack, chosen to illustrate the proposed algorithm's capabilities in the case of specific SDWSN vulnerabilities that exhibit largely different behavior. Both attacks are explained in Subsection 3.9.2, next.

3.9.1 SDWSN Security Analysis

The SDN networks security threats are grouped in three categories [38]: application plane attacks, control plane attacks, and data plane attacks. Among the three, the control plane attacks are pointed out as the most high impact and attractive [38] [39], as the control plane is responsible for the overall management of the network [40]. This characteristic turns the control plane prone to distributed denial of service (DDoS) attacks. For example, an intruder may flood the network with flow rule requests, which could lead to an exhaustion of the controller's resources. This attack can be intensified using multiple intruders.

The threats and vulnerabilities explained before also apply to SDWSNs. Moreover, there are specific attacks that can attain SDWSNs due to resources constraints, for example: in SDWSN the forwarding devices have low storage capacity, which limits the memory assigned for flow tables and buffers. These constraints make the forwarding devices prone to saturation attacks. Also, SDWSN networks are characterized for having a limited bandwidth and low processing power. This means that a saturation attack can also result in a DoS attack.

Another vulnerability concerns the gateway between the SDN controller and the WSN. The gateway has a radio module of limited bandwidth, rendering it a weak link even when the controller has enough resources to overcome an attack.

For the reasons outlined above, most of the security mechanisms designed for standard SDN networks have to be adapted or redesigned. This is one of the major challenges for SDWSN security.

3.9.2 FDFF and FNI DDoS Attacks

We briefly describe the two attacks considered in this Section, FDFF and FNI. The design of the attacks are inspired by [144].

The FDFF attack sends false flow rule requests to the controller, using each attacker's neighbors (benign nodes). Each attacker send one data packet to its neighbors tagged with an unknown flow identifier; as the neighbors do not have a rule to apply to the packet, they send a flow request to the controller asking a rule for the unknown flow identifier. Thus, the intensity of the attack is multiplied by the number of neighbors. The FDFF attack could triple the number of control packets in the whole network, having a minor impact on the delivery rate, as it is shown in [144].

On the other hand, in the FNI attack, each attacker intercept packets containing neighbor information, modifying them with false neighbor information and forwarding them to the controller. The controller then, updates the network topology graph using the false information, and reconfigures the network with wrong forwarding rules. Experimental results in [144] showed that the FNI attack could double the number of control packets in the whole network having also a significant impact on the delivery rate.

3.9.3 RCPD for Intrusion Detection

We employed the RCPD algorithm in SDWSNs under FDFF and FNI attacks. We simulated grid topologies with 36 and 100 nodes, with 20% attackers in the network. Each simulation run during 10 hours and each scenario was replicated 30 times. During the first 8 hours the network operated normally, then the attack was triggered. The choice of 8 hours was made because empirically it was seen that we needed at least 250 samples for the training period and we obtained one sample every 2 minutes. The simulations were performed using the COOJA simulator [145] and sky motes, and were designed and executed by Gustavo Alonso Nunez Segura at the premises of São Paolo University in Brazil. The MAC layer was the IEEE 802.15.4, configured to work without radio duty cycle (`nullrdc_driver`). The data sink received the application data, while the management sink received performance metrics information. Notice that the SDN controller is a different node from the sink. Table 3.3

depicts the simulation parameters.

Table 3.3: Simulation parameters

Simulation parameters	
Topology	Square grid
Number of nodes	36 and 100
Simulation duration	36000 s
Node boot interval	$[0, 1]$ s
Number of sinks	2
Sinks position	Middle of the grid edge
Data traffic rate	1 packet every 30 seconds
Management traffic rate	1 packet every two minutes
Data payload size	10 bytes
Management payload size	10 bytes
Data traffic start time	$[2, 3]$ min
Radio module power	0 dB
Distance between neighbors	50 m
Attacks begins after	28800 s

IT-SDN parameters	
Controller position	center
ND protocol	Collect-based
Link metric	ETX
CD protocol	none
Flow setup	source routed
Route calculation algorithm	Dijkstra
Route recalculation threshold	10%
Flow setup types	regular or source routed
Flow table size	10 entries

We analyzed the data packets delivery rate and the control packets overhead. The delivery rate was calculated by dividing the total number of packets

successfully received by the total number of packets sent. The control packets overhead was quantified as the total amount of control packets sent. Those metrics were updated every two minutes.

The metrics measuring the performance of the intrusion detection algorithm were the following: i) the detection rate (DR); ii) the false positive rate (FPR); iii) the false negative rate (FNR); iv) the detection time median (DTM), indicating the median of the time instances elapsed from the launch of the attack to the instance it was identified; and v) the median absolute deviation (MAD). The detection rate is defined as the ratio between the correctly detected attacks and the total number of attacks. The false positive rate is defined as the ratio between the number of attack events classified as attack and the total number of attack events. The false negative rate is defined as the ratio between attack events classified as non-attack event and the number of attack events. The detection time median is defined as the median of the number of samples required to detect the attack. The median absolute deviation measures the variability of the detection times and is calculated as shown in (3.35), where X_i is the detection time for replication i, and $\tilde{X}$ is the median of all the detection times,

$$\text{MAD} = \text{median}(|X_i - \tilde{X}|). \tag{3.35}$$

The delivery rate and control overhead time series were analyzed for three monitoring windows and three critical values. We used monitoring periods $K \in \{50, 100, 150\}$ samples. This means that the test statistic was run over K samples to extract changes in the mean value. We considered as confidence intervals the set $\{90\%, 95\%, 99\%\}$ for the calculation of the critical values. Finally, in this analysis, we discarded the first 15 samples because during this time the network was bootstrapping.

3.9.4 Results and Analysis

In this Subsection we present and analyze the simulation results. In Subsection 3.9.5 we compare the FDFF attack detection performance when monitoring the data packets delivery rate and the control overhead. In Subsection 3.9.6 we repeat this analysis for the FNI attack.

3.9.5 FDFF Attack Detection

Table 3.4: FDFF attack detection, 36 nodes, 20% attackers

Data packets delivery rate

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	28	28	28	30	24	28	29	28	28
MAD	5	8	6	11	7	8	6	5	8
DR	77	80	73	73	83	73	77	80	77
FPR	3	07	7	0	3	0	0	3	0
FNR	20	13	20	27	13	27	23	17	23

Control overhead

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	8	7	7	5	5	5	8	7	7
MAD	2	2	2	1	1	1	2	2	2
DR	100	100	100	97	87	97	100	100	100
FPR	0	0	0	3	13	3	0	0	0
FNR	0	0	0	0	0	0	0	0	0

Tables 3.4 and 3.5 summarize the FDFF attack detection results when 20% of nodes are attackers. In the case of 36 nodes, the DR was between 73% and 83% when monitoring the data packets delivery rate, and between 87% and 100% when monitoring the control packets overhead. In terms of detection time, the best DTM when monitoring the data packets delivery rate was 24 samples and the DTM when monitoring the control packets overhead was 5 samples. Configuring the monitoring period in 100 we obtain the best DTM, but there was a drop in the DR if compared with the cases when using monitoring periods of 50 and 150 samples.

The results for 100 nodes showed it is possible to obtain a DR of 100%

Table 3.5: FDFF attack detection, 100 nodes, 20% attackers

Data packets delivery rate

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	15	13	14	8	7	7	15	14	14
MAD	5	6	5	6	5	5	5	5	5
DR	100	93	100	97	93	97	100	97	97
FPR	0	7	0	3	7	3	0	3	3
FNR	0	0	0	0	0	0	0	0	0

Control overhead

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	4	4	4	3	3	3	4	4	4
MAD	0	0	0	0	0	0	0	0	0
DR	100	97	100	97	90	97	100	97	100
FPR	0	3	0	3	10	3	0	3	0
FNR	0	0	0	0	0	0	0	0	0

monitoring any of the metrics, but there were significant differences in the detection time. The DTM when monitoring the control overhead is between 3 and 4 samples, while when monitoring the data packets delivery rate the DTM was between 7 and 15 samples. Considering the earliest detection with the highest DR for both monitoring metrics, it occurred when using a monitoring period of 100 samples. For both cases the DR obtained was 97%. In terms of FPR and FNR, the best performance was obtained when monitoring the control overhead and using a monitoring period of 50 and 150 samples. Monitoring the control overhead using a monitoring window of 100 samples provided a FPR between 3% and 10%.

Summarizing, the algorithm was able to detect the FDFF attack using either the data packet packets delivery rate or the control packets overhead as

Table 3.6: FNI attack detection, 36 nodes, 20% attackers

Data packets delivery rate

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	7	7	7	7	7	7	8	7	7
MAD	2	2	2	3	4	3	2	2	2
DR	100	100	100	100	100	100	100	100	100
FPR	0	0	0	0	0	0	0	0	0
FNR	0	0	0	0	0	0	0	0	0

Control overhead

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	26	24	26	26	24	27	26	24	26
MAD	8	7	7	17	11	13	8	7	7
DR	57	70	60	43	63	57	57	70	60
FPR	0	0	0	0	0	0	0	0	0
FNR	43	30	40	57	37	43	43	30	40

inputs; achieving better performance monitoring the control packet overhead. Aiming for the quickest detection captured through the detection time median, the algorithm achieved far better results when monitoring the control packets overhead in all scenarios.

3.9.6 FNI attack detection

Tables 3.6 and 3.7 summarize the FNI attack detection results when 20% of nodes were attackers. Opposite to the FDFF attack results, the algorithm obtained a better performance detecting the FNI attack when monitoring the data packets delivery rate. The above result was due to the high FNR when monitoring the control packets overhead. In the case of 36 nodes, the DR

Table 3.7: FNI attack detection, 100 nodes, 20% attackers

Data packets delivery rate

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	9	10	10	8	9	8	10	12	11
MAD	5	8	7	4	6	4	5	9	8
DR	100	100	100	100	100	100	100	100	97
FPR	0	0	0	0	0	0	0	0	3
FNR	0	0	0	0	0	0	0	0	0

Control overhead

K	**50**			**100**			**150**		
α	**90**	**95**	**99**	**90**	**95**	**99**	**90**	**95**	**99**
DTM	27	24	26	26	25	25	27	24	26
MAD	6	3	6	6	6	6	6	3	6
DR	93	97	97	93	97	93	93	97	97
FPR	0	0	0	0	0	0	0	0	0
FNR	7	3	3	7	3	7	7	3	3

when monitoring the data packets delivery rate was 100%, and the DR when monitoring the control packets overhead was between 43% and 70%. For 100 nodes, the DR when monitoring the data packets delivery rate was between 97% and 100%, and the DR when monitoring the control packets overhead was between 93% and 97%.

About the DTM, the results for the scenarios when monitoring the data packets delivery rate were between 4 and 9 samples. The results for this same metric when monitoring the control packets overhead were between 24 and 26 samples. This means, for grid topologies with 100 nodes where 20% of nodes were attackers, we obtained similar DRs regardless of the monitoring metric, but when monitoring the delivery rate the detection was at least 3 times faster.

Summarizing our findings, the algorithm was able to detect the FNI attack

monitoring either the data packet packets delivery rate or the control packets overhead. Then, comparing the detection performance based on the detection rate and the detection time median, the algorithm obtained a far better performance when monitoring the data packets delivery rate in all scenarios. This effect was directly related to the type of the attack.

3.10 Conclusions

We developed an extension of the RCPD in order to identify changes in the variance structure of content popularity time series. In this extended form, we also incorporated parametric CP tests. The simulation results over synthetic data demonstrated that non parametric and GARCH model based approaches can better suited for content views time series with unknown statistics. Furthermore, the non-parametric and the GARCH based variations were applied on real YouTube video content views time series, to illustrate the performance of the proposed approach of volatility change detection.

Last but not least, we evaluated the performance of the RCPD algorithm on two particular cases. More precisely, we incorporated the RCPD in an elastic next-generation CDN platform, in order to signal content popularity changes providing the means for efficient VMs allocation. We also investigated the performance of RCPD in the context of detecting SDWSN DDoS attacks.

Chapter 4

Scheduling Optimization of Heterogeneous Services in 5G NR

4.1 Introduction

The international telecommunication union (ITU) has defined new requirements and capabilities for 5G mobile communication systems so as to support a wide variety of new devices and services with varying quality of service (QoS) requirements and characteristics. The 3rd generation partnership project (3GPP) standardized 5G in the form of a novel radio interface technology, referred to as new radio (NR) [6]. 5G NR introduced flexible numerology and frame structure to accommodate heterogeneous service requirements, by supporting various values of subcarrier spacing and symbol / frame duration. Optimizing resource allocation in the NR numerology setting to deliver heterogeneous QoS requirements remains a challenging task [146], [147], [148], while the major challenges related to radio resource optimization for ultra-reliable low-latency communication (URLLC) systems are described in [149].

In 5G and beyond, URLLC services with extreme delay constraints will coexist with enhanced mobile broadband (eMBB) [150], that require very high bit rates (Gigabits per second) and have moderate latency (a few milliseconds) requirements [151]. Moreover, at present, URLLC services are expected to

have lower traffic volumes than eMBB services [152], while this might not hold in the future for applications such as virtual reality and haptics. In this framework, the design of radio resource allocation strategies for URLLC traffic when coexisting with eMBB has been a focal point of recent research efforts [153–156].

In this Chapter, we propose near optimal scheduling algorithms and strategies of radio resources for URLLC when coexisting with eMBB services. The novelty in our analysis is the re-formulation of the standard eMBB throughput maximization problem as an equivalent conflict minimization with URLLC. We provide a conflict-aware heuristic solution to solve the re-formulated problem. Subsequently, we show that the conflict can be treated as a bin backing optimization, so that the objective becomes to fit URLLC demands in a minimum number of resource blocks (bins). Simulation results show that the proposed low complexity heuristic algorithms performs near optimally.

Moreover, in order to further increase the efficiency of resource utilization, non-orthogonal multiple access (NOMA) is also investigated for URLLC and eMBB coexistence. The superior performance of NOMA, with superposition of services over the same resource blocks, is due to alleviating conflicts, as shown by an extensive set of numerical results. The present work highlights that the most important potential gains in employing NOMA technologies in B5G might not stem from the mere increase of the spectral efficiency, but rather from the avoidance of conflicts. Although not treated in the present thesis, these conclusions carry on to the B5G uplink by reducing the potential number of collisions, particularly in the massive connectivity regime.

4.2 Contributions and Chapter Organization

In this Chapter, we consider a flexible, 2-dimensional grid of resource blocks with different sizes in the time and frequency domains. The problem of identifying the resource allocation that maximizes eMBB sum-rate is studied under the constraint of covering all URLLC throughput demands under different latency constraints ranging from 0.25 to 2 milliseconds (ms). We notice that due to the potential full or partial overlap of resource blocks, not all service placements are feasible once a placement is executed. Here, we argue that managing

infeasible placements is a key, but overlooked, aspect of this process. The main contributions of this work are outlined below:

We first re-formulate the problem of eMBB throughput maximization, introducing the URLLC conflicts minimization in the objective function. The novel concept of "conflict" captures the penalties occurring due to the fact that orthogonal multiple access (OMA) does not allow overlapping of resources; as a result, OMA scheduling incurs a large number of infeasible resource allocation combinations. This new view angle allows for proposing completely novel solutions for the problem at hand.

Next, we propose three conflict-aware, multi numerology radio resource allocation heuristics to maximize scheduling efficiency for URLLC, when coexisting with eMBB services. Three different functions of the i) average, ii) the instantaneous (placement specific), or iii) the aggregate conflict are used to normalize the throughput utility function and incorporate penalties, when increasing conflicts. We argue and showcase through extensive simulation results that employing the proposed utilities improves the performance of proposed algorithms in the literature, as this in [1].

Subsequently, we depart on a completely different approach with a high accuracy and low computational complexity. We treat the scheduling problem as a specific instance of bin packing optimization, solved by minimizing the placements of URLLC services in the time-frequency resource grid; to this end, we propose to group the resource blocks in different categories with respect to URLLC demands. Within each category, we solve a knapsack maximization of the sum eMBB throughput. Our proposal builds on previous results in [157] and is inspired by the refined-first-fit family of heuristics to solve bin packing problems. Simulation results show that the novel heuristic algorithm, of complexity $N \log(N)$, provides a quick, lightweight and near optimal solution to the resource allocation scheduling of URLLC, when coexisting with eMBB.

Moreover, having shed light to the importance of minimizing conflicts between different services, the utilization of NOMA schemes naturally emerges as a competitive candidate [43]. NOMA allows for the superposition of services, even at the 5G NR mini-slot level, by employing superposition coding at the transmitter and successive interference cancellation at the receivers [158], [159]. NOMA has in the past been proposed as a competitive scheme to enhance

throughput per resource block [160]; in the present work we provide further motivation for it's employment in B5G as the means to mitigate conflicts in the allocation of resource blocks, i.e., in layer 2 scheduling. We provide an extensive set of numerical results that show the significant gains in terms of eMBB throughput when adopting NOMA in both fixed and flexible numerology settings.

4.2.1 Chapter Organization

The rest of this Chapter is organized as follows. In Section 4.3 a brief background on radio resource allocation in the case of coexisting heterogeneous services is presented, while, a short discussion about multiple access techniques and NR numerologies is also included. The resource allocation optimization problem is described in Section 4.4, along with the equivalent novel formulation as a conflict minimization problem. Conflict-aware heuristic algorithms are proposed in Section 4.5, including one inspired by a heuristic solution to the bin packing problem, along with the problem re-formulation when using NOMA. Section 4.6 presents numerical results showing the near-optimal performance of the heuristics as well as the superiority of NOMA for URRLC and eMBB coexistence, both in the case of flexible as well as fixed numerologies. Finally, conclusions are drawn in Section 4.7.

4.3 Background and Related Works

4.3.1 Brief Review of Mathematical Optimization Concepts

We briefly present fundamental aspects of mathematical optimization, focusing on techniques that we later discuss in Chapter 4 for the scheduling of radio resources for URLLC when coexisting with eMBB services.

Following the notation in [161] a mathematical optimization problem has

the form,

$$\begin{aligned} &\max_x f_0(x) \\ &\text{s.t. } f_i(x) \leqslant a_i, \quad i = 1, \ldots, m. \end{aligned} \tag{4.1}$$

The vector $x = (x_1, \ldots, x_n)$ is the *optimization variable* of the problem, the linear function f_0 is the *objective function*, the linear functions f_i are the inequality constraint functions, and the constants $b_1, \ldots, b_m$ are the limits, or bounds, for the constraints. A vector x^* is called optimal, if it has the maximum objective value among all vectors that satisfy the constraints.

The class of *linear programming* (LP) problems is defined for $x \in \mathbb{R}_+^n$; respectively, the class of *integer programming* (IP) problems is defined for $x \in \mathbb{Z}_+^n$; finally, the class of *binary programming* (BP) problems is defined for $x \in \{0, 1\}^n$.

Duality

Duality is a fundamental technique to solve complex, e.g., computational intensive, optimization problems. With the dual problem one can determine a lower (upper) bound to the solution of the primal minimization (maximization) problem. Moreover, under certain conditions, the solutions of both problems are equal. The basic idea behind duality is to interchange the constraints and the objective function. Consider the general form of a LP (primal) problem,

$$\begin{aligned} &\max_x cx \\ &\text{s.t.} Ax \leqslant b, \\ &\qquad x \geqslant 0, \end{aligned} \tag{4.2}$$

where $A \in \mathbb{R}^{m \times n}$, $c,\ x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. The dual problem can be written as,

$$\begin{aligned} &\min_u ub \\ &\text{s.t } uA \geqslant c \\ &\qquad u \geqslant 0, \end{aligned} \tag{4.3}$$

where $u \in \mathbb{R}^m$ is a vector of *dual variables*. Note that the dual problem has m variables and n constraints. This is because variables in the dual correspond to constraints in the primal, and vice versa.

The value of any feasible dual solution provides an upper bound on the optimal value of the primal problem. For any x and u that are primal and dual feasible, respectively, $ub \geqslant uAx \geqslant cx$. The first inequality is due to the fact that $Ax \leqslant b$ and $u \geqslant 0$, and the second is due to the fact that $uA \geqslant c$ and $x \geqslant 0$. By finding a dual feasible solution, one can estimate how much a primal feasible solution falls short of optimality. Although this property is less important for LP, where robust solution algorithms are available, it is essential for IP.

Hence, the *weak duality* states that if x^* and y^* are feasible solutions for the primal and dual, respectively, and the primal problem is a maximization (minimization) problem, then $cx^* \leqslant bu^*$ ($cx^* \geqslant bu^*$). While, the *strong duality* states that if the primal (dual) problem has a finite optimal solution, then so does the dual (primal) problem, and these two values are equal.

Let us discuss in further detail the basic intuition behind the Lagrangian duality function. Consider the optimization problem given in eq. (4.1) with variable $x \in \mathcal{X}$. We assume $\mathcal{D}$ the domain of the problem (which is the intersection of the domains of all the functions involved) and $\mathcal{X} \subseteq \mathcal{D}$. Furthermore, we denote by p^* the optimal solution of the problem.

The basic idea in Lagrangian duality (LD) is to augment the objective function with a weighted sum of the constraint functions of the primal problem; thus some of the troublesome constraints can be removed, but inserting a penalty for violating them in the objective function. Then the *Lagrangian* relaxation $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ may be introduced,

$$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x). \tag{4.4}$$

The vectors $\lambda \in \mathbb{R}^m$ are referred to as *Lagrange multipliers*. Then it holds that,

$$\forall x \in \mathcal{X},\ \forall \lambda \geqslant 0,\ L(x, \lambda) \geqslant p^*. \tag{4.5}$$

Next the LD is defined as,

$$g(\lambda) = \max_{x \in \mathcal{X}} L(x, \lambda). \tag{4.6}$$

This is a relaxation in the sense that its optimal value $g(\lambda)$ is the upper bound on the optimal value p^*,

$$\forall \lambda \geqslant 0,\ g(\lambda) \geqslant p^*. \tag{4.7}$$

Given that the *Lagrange dual function* gives valid upper bounds $\forall \lambda \geqslant 0$, the best upper bound is the

$$\min_{\lambda \geqslant 0} g(\lambda). \tag{4.8}$$

The above problem is the dual problem, and the parameter $\lambda \in \mathbb{R}^m$ is called as the *dual variable*. The dual problem involves the minimization of a convex function under concave constraints, so it is a concave problem.

Optimization Problems

In this Subsection we present two well known optimization problems aligned with the general goals of the Chapter, that is to maximize the throughput of eMBB services with respect to the URLLC demands in a minimum number of resource blocks.

In general, *Knapsack Problems* (KP) refer to the class of problems in which we want to select an optimum subset of bounded weights, from a set of elements each of which has a weight and a profit. In other words, in the KP, a set of N items $i = \{1, \ldots, N\}$ is given, where each item has value $v_i \in \mathbb{Z}_+$ and a weight $w_i \in \mathbb{Z}_+$; the items need to be optimally selected to fit in a knapsack with overall capacity $W \in \mathbb{Z}_+$. Thus, the goal is to find the subset of items $S \subseteq I$

that,

$$\max_{x_i \in \{0,1\}} \sum_{i=1}^{N} v_i x_i$$
$$\text{s.t.} \sum_{i=1}^{n} w_i x_i \leqslant W.$$

Dantzig [162] showed that, for $x_i \in \mathbb{R}_+$, the problem has an optimal solution that is to sort in a decreasing order the ratios $\frac{v_i}{w_i}$, $i = 1, \ldots, N$ and the variable with the highest ratio is the best item to place in the knapsack. Then the item with the second highest ratio is put in and so on until we reach an item that cannot fit, $k = \min\left\{j \in \{1, \ldots, N\} : \sum_{i=1}^{j} w_i > W\right\}$, where we take the fraction of this last item, $x_k = \frac{W - \sum_{j=1}^{k-1} w_j}{w_k}$, which satisfies the constraint. Sorting the element takes $\mathcal{O}(N \log N)$ and the calculations $\mathcal{O}(N)$. A much faster solution is provided by the weighted median algorithm [163] that solves the problem in linear time.

KP is $\mathcal{NP}$-hard and, among others, an efficient way to solve the problem is by dynamic programming [164], with solving time $\mathcal{O}(NW)$. For the dynamic programming solution the array $M[0 \ldots N, 0 \ldots W]$ is defined. For $k = 1, \ldots, N$ and $w = 0, \ldots, W$ the entry $M[k, w]$ corresponds to the maximum value of any subset $\{1, \ldots, k\}$ of capacity at most w. We initialize as,

$$M[0, w] = 0, \text{ for } 0 \leqslant w \leqslant W$$

and calculate the recursive formula,

$$M[k, w] = \begin{cases} \max\{M[k-1, w], u_k + M[k-1, w-1]\}, & \text{if } w_k \geqslant w \\ M[k-1, w], & \text{otherwise} \end{cases}$$

The solution is reached by calculating $M[N, W]$.

A second class of optimization problems is the *Bin Packing Problem* (BPP). The BPP is a commonly studied combinatorial problem in which a finite set of

items of known and different weights must be assigned into a finite number of bins without exceeding the capacity of each bin. The goal is to minimize the number of bins used. The problem can be described, given the terminology of KP, as in [165]

$$\begin{aligned}
\min \quad & \sum_{k=1}^{N} y_k \\
\text{s. t.} \quad & \sum_{k=1}^{N} x_{i,k} = 1, && i = 1, \ldots, N \\
& \sum_{i=1}^{N} w_i x_{i,k} \leqslant W y_k, && k = 1, \ldots, K \\
& y_k \in \{0, 1\}, && k = 1, \ldots, K \\
& x_{i,k} \in \{0, 1\}, && i = 1, \ldots, N,\ k = 1, \ldots, K
\end{aligned}$$

where N is the number of items, w_i is the weight of item i, W is the bin capacity and the variables are:

$$y_k = \begin{cases} 1, & \text{if bin } k \text{ is used,} \\ 0, & \text{otherwise;} \end{cases} \qquad x_{i,k} = \begin{cases} 1, & \text{if item } i \text{ is assigned to bin } k, \\ 0, & \text{otherwise.} \end{cases}$$

Although the problem seems simple to define, it is a well-known $\mathcal{NP}$-hard problem [166],while the decision if items will fit into a specified number of bins is $\mathcal{NP}$-complete problem. Therefore, different approximation solutions have been proposed for solving the BPP based on heuristic approaches. The most common online algorithms are the *Next Fit* (NF), the *First-Fit* (FF) and the *Best-Fit* (BF). While, by sorting the items in a decreasing order and applying to the sorted list of items the aforementioned heuristics, the decreasing off-line algorithms NFD, FFD and BFD are constructed. In [167] an overview of the approximation algorithms used for solving the BPP is provided.

Our bin-packing based heuristic for the resource allocation of URLLC when coexisting with eMBB services, relies to the *Refined First-Fit* (RFF) algorithm. The RFF algorithm [168] divides the set of all bins into infinite classes before

packing and it essentially performs an FF algorithm within each class of bins. The computation complexity of RFF is $N \log(N)$.

4.3.2 Resource Allocation in Heterogeneous Services

The efficient sharing of radio resources in heterogeneous services in general [43], and between eMBB and URLLC traffic specifically, has recently attracted considerable interest in the literature [169]. In this direction, two approaches have been adopted by the 3GPP. The first is based on a "puncturing" framework: according to this, eMBB traffic is scheduled initially at the beginning of the slots; upon arrival of URLLC traffic, the latter is being prioritized and dynamically overlapped at mini-slots of ongoing eMBB transmissions (which are punctured, i.e., dropped). In the second approach, known as preemptive scheduling, resources are preemptively reserved for URLLC, before the demands are placed [1, 170–172].

Based on puncturing scheduling, the studies in [42, 173–175] considered resource allocation strategies for the coexistence of URLLC and eMBB. The authors in [42] consider three types of models - threshold, linear and convex - to describe the eMBB data rate loss associated with the incoming URLLC traffic. Furthermore the authors in [174] propose a punctured scheduling approach for transmission of low latency communication (LLC) traffic multiplexed on a shared channel with eMBB. Another approach is proposed in [175], where a risk-sensitive model was introduced in order to ensure URLLC allocation but also to minimize the loss of eMBB users. However, these strategies can result in significant losses in terms of data rates for eMBB services [176] and may impact eMBB transmission reliability [177].

Alternatively, the authors in [1] studied the resource allocation of eMBB and URLLC services by preemptively reserving resources for URLLC. Such solutions ensure advantageous conditions for URLLC packets when they are generated, at the cost of wasting resources in absence of URLLC transmissions [177].

A flexible numerology and frame structure was explicitly considered in [1] by defining a time-frequency resource grid, containing different types of resource blocks of different shapes, expanding over different time spans and frequency ranges. Exploiting this flexibility to optimize the resource allocation to different

services while ensuring their QoS requirements, was shown to be an NP-hard problem. However, the resource allocation optimization over flexible numerology and frame structure while avoiding the assignment of overlapping of blocks (i.e., of puncturing), still remains a challenging task.

4.3.3 Flexible Numerology

In the above Subsection we presented a brief literature for the problem of resource allocation with heterogeneous services, focusing on puncturing and preemptive scheduling approaches. However, suggestions that admit only non-flexible numerology structures, cannot adapt well to the challenge of the high heterogeneity environment that the 5G services bring. For example, the traditional transmission time interval (TTI) of 15 kHz - 1 msec (of LTE) could hardly satisfy ultra-low latency services with time delay tolerance 0.5 msec, 0.25 msec and even lower [178].

Thus, in order to meet the vastly diverse requirements and services the concept of flexible numerology has been proposed, where the term numerology refers to the different configurations of subcarrier spacing (SCS) and cyclic prefix (CP) duration of an orthogonal frequency division multiplexing (OFDM) symbol [179]. More precisely, 5G NR Release-15 [6] defines a flexible numerology with SCS of 15, 30, and 60 kHz below 6 GHz, and 60 and 120 kHz above 6 GHz, compared to LTE which uses a fixed numerology with SCS of 15 kHz below 6 GHz. 5G NR also defines a 10 msec frame, with each frame divided into 10 subframes of 1 msec, which are further divided into one or more slots. A mini-slot comprises 14 OFDM symbols for a configuration using normal cyclic prefix, or 12 OFDM symbols in the case of an extended cyclic prefix.

In 5G NR, the mini-slot size is defined according to the symbol duration, which is inverse to the SCS, to ensure the orthogonality of the subcarriers. By using higher SCS, the symbol duration decreases and hence also the mini-slot size, which is beneficial for lower latency [150], while lower SCS numerologies are more suitable for eMBB which requires both, high data rate and significant bandwidth. URLLC traffic requires extremely low delays, often lower than 1 ms [180]. The URLLC latency requires extremely low delays ($0.25 - 10$ msec/packet) [152] and the requirements can only be satisfied if the transmission

duration and round-trip-time (RTT) are shorter than the corresponding latency constraint. Accordingly, [177] argues that the efficient support of URLLC transmissions needs either to reduce the symbol period by controlling the sub-carrier spacing, or to reduce the number of symbols in the packet TTI. The works of [181] and [154] also reach similar conclusions. More precisely, they propose that eMBB services should be scheduled with a longer TTI, while URLLC services could be scheduled on a shorter duration, to satisfy its tight latency deadline. Finally, the major challenges related to radio resource optimization for URLLC systems are described in [149].

4.3.4 Non Orthogonal Multiple Access (NOMA)

In the majority of resource allocation approaches where eMBB and URLLC traffic types coexists, OMA is applied to provide multiple users with simultaneous access to the spectrum. An orthogonal scheme allows a perfect receiver to entirely separate unwanted signals from the desired signal; signals from different users are orthogonal to each other in orthogonal schemes ensuring in the problem at hand the isolation of eMBB and URLLC traffic. As an alternative, NOMA [159] increases the spectral efficiency by employing superposition coding (SC) at the transmitter and with successive interference cancellation (SIC) at the receiver. The SC [182] is a transmission method of communicating superimposed (in the code, power or space domain) signals to multiple receivers from a single source. In this thesis we focus entirely on superposition in the power domain, i.e.; a single transmitter superimposes, with carefully selected power levels, signals intended for different users (receivers). On the other hand, SIC [183] is a technique to decode the superposed signals at each receiver; successively, starting from the weak(er) user and moving to strong(er) users in the downlink, while the order is reversed in the uplink..

NOMA has been proposed for scheduling radio resources with homogeneous requirements [184] and [185]. Recently NOMA has been applied also in heterogenuous services improving the overall performance; i.e., maximizing the eMBB throughput while meeting low-latency requirements of URLLC [186]. The authors in [187] compared the performance of OMA and NOMA for the multiplexing of eMBB and uRLLC users in the uplink using a cloud radio

access network (C-RAN) architecture. Correspondingly, [43] proposed the heterogeneous - NOMA (H-NOMA) highlighting the NOMA perspective for the eMBB allocation, when coexisting with URLLC and mMTC traffic.

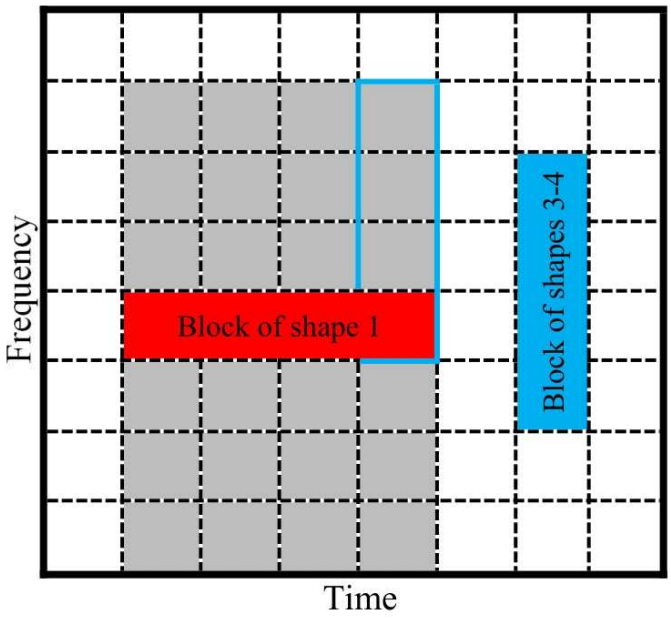
4.4 Problem Formulation

We focus in this work on downlink scheduling, with one base station (BS) servicing both throughput hungry (eMMB) and ultra-low latency users (URLLC). The objective is to find the resource allocation in the time-frequency grid that maximizes the sum throughput of the former while satisfying the throughput demands and latency constraints of the latter. Our starting point is the system model of [1]. We also utilized [188] as a tool to implement the time-frequency grid.

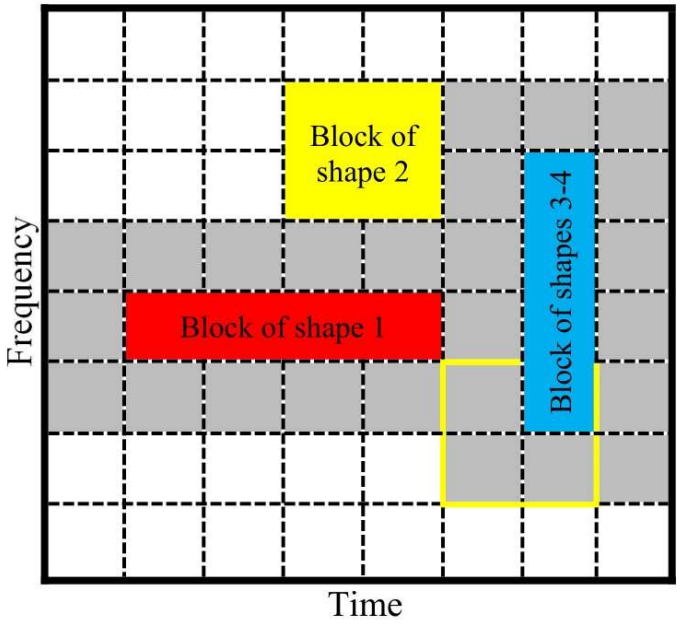
Table 4.1: The notations table

Notation	**Sets**
$\mathcal{K}^{(\ell)}$	Set of URLLC services
$\mathcal{K}^{(c)}$	Set of eMBB services
$\mathcal{K}$	Set of all services, for which $\mathcal{K} = \mathcal{K}^{(\ell)} \cup \mathcal{K}^{(c)}$
$\mathcal{B}$	Set of candidate blocks with respect to the numerology
$\mathcal{I}$	Set of all basic units of the grid, i.e., the minimum unit of resource in the time-frequency domain (mini slots)
Notation	**Parameters**
τ_k	maximum tolerant latency of service k
q_k	data demand (in bits) of service k

The terminology employed in the rest of the chapter is tabulated in Table 4.1: $\mathcal{K}$ denotes the set of all services, $\mathcal{K}^{(c)}$ the set of eMBB users, $\mathcal{K}^{(\ell)}$ the set of URLLC users, $\mathcal{B}$ is the set of all possible resource blocks according to the numerology employed and finally, $\mathcal{I}$ denotes the set of all mini-slots. We utilize the parameter $\alpha_{b,i}, b \in \mathcal{B}, i \in \mathcal{I}$ which indicates whether a block $b \in \mathcal{B}$ includes basic unit $i \in \mathcal{I}$, in which case $\alpha_{b,i} = 1$, otherwise $\alpha_{b,i} = 0$. Furthermore, we denote by $r_{b,k}, b \in \mathcal{B}, k \in \mathcal{K}$ the throughput of each resource block, under the



(a) Allocation of a candidate block of shapes 3-4, when a block of shape 1 already exists. Blue line indicates a conflicted block.



(b) Allocation of a candidate block of shape 2, when a block of shape 1 and a block of shapes 3-4 already exist. Yellow line indicates a conflicted block.

Figure 4.1: Time-frequency resource allocation, considering the flexible numerology context, with three types of resource blocks and the corresponding conflicts (grey).

constraint that the latency constraint is met. Additionally, by $x_{b,k}$ we denote a binary variable that takes the value 1 if the resource block $b \in \mathcal{B}$ is assigned to service k, otherwise $x_{b,k} = 0$.

Table 4.2: Resource blocks in flexible numerology

	Shape 1	Shape 2	Shape 3	Shape 4
TTI duration (msec)	0.5	0.25	0.125	0.125
SCS (kHz)	15	30	60	60
Symbol duration (μs)	66.7	33.3	16.7	16.7
CP (μs)	4.7	2.3	1.2	4.17
Number of Symbols	7	7	7	6

In Table 4.2 we describe the most widely utilized resource block specifications for 5G NR, depicted in Fig. 4.1(a) and (b); resource blocks of shape 1 shown in red, resource blocks of shape 2 shown in yellow and resource blocks of shapes $3-4$ shown in blue. Employing flexible numerology, $\mathcal{K}^{(c)}$ (eMBB) and $\mathcal{K}^{(\ell)}$ (URLLC) services have no restrictions and can utilize any of the given shapes. To demonstrate the concept of conflict, in Fig. 4.1(a), we illustrate in gray shade the invalid placements for shapes 3-4 when a specific placement of shape 1 has taken place, while in Fig. 4.1(b) we show the invalid placements for blocks of shape 2, when an additional placement of shape 3-4 has been decided.

A common objective in eMBB and URLLC coexistence is articulated in maximizing the sum throughput of $\mathcal{K}^{(c)}$ services under the constraint of satisfying the latency and throughput demands of $\mathcal{K}^{(\ell)}$, without any overlapping between the allocated resource blocks. In other words, our goal is to find the resource allocation that satisfies the URLLC users' demands, with minimal losses for eMBB users in terms of throughput, and, subsequently schedule all the remaining resource blocks to the eMBB services. The formal problem

formulation is given as follows:

$$[\text{P0}] \quad \max_{x_{b,k}\in\{0,1\}} \sum_{b\in\mathcal{B}} \sum_{k\in\mathcal{K}^{(c)}} r_{b,k}x_{b,k}, \tag{4.9}$$

$$\text{s.t.} \quad \sum_{b\in\mathcal{B}} r_{b,k}x_{b,k} \geq q_k, \quad k \in \mathcal{K}^{(\ell)}, \tag{4.10}$$

$$\sum_{b\in\mathcal{B}} \sum_{k\in\mathcal{K}} a_{b,i}x_{b,k} \leq 1, \quad i \in \mathcal{I}. \tag{4.11}$$

In [1] it was proven that the combinatorial problem P0 is an $\mathcal{NP}$-hard partition problem and a heuristic algorithm was proposed, referred to in following as the *baseline heuristic*. The baseline heuristic uses a utility matrix $\mathbf{u}$ with elements $u_{b,k}$ that represent the utility of a block $b \in \mathcal{B}$ assigned to a specific service $k \in \mathcal{K}$. Then, in the *first step* of the heuristic algorithm, the block b is allocated to service $k \in \mathcal{K}^{(\ell)}$ with the maximum $u_{b,k}$ while all the overlapping – to b – blocks are removed; notice that choosing the allocation that maximizes the utility without at the same time examining the "cost" of this placement in terms of generated conflict is clearly sub-optimal. The step is iterated until all the demands for $k \in \mathcal{K}^{(\ell)}$ are satisfied under the constraint (4.10). Next, in the *second step*, the placements for $k \in \mathcal{K}^{(c)}$ services are allocated, using a similar principle, until no other non-overlapping blocks have remained. Hence, the placement of the $\mathcal{K}^{(\ell)}$ and $\mathcal{K}^{(c)}$ has been treated as two separate resource allocation problems. The complexity of the baseline heuristic algorithm was shown to be $\mathcal{O}(|\mathcal{B}||\mathcal{K}|\log(|\mathcal{B}||\mathcal{K}|))$, without accounting for the computation of utility matrices.

The baseline heuristic has been extended in [1] to incorporate other utility matrices denoted by $\mathbf{u}_{LP}$, $\mathbf{u}_{LD} \in \mathbb{R}_{\mathcal{B}\times\mathcal{K}}$, where $\mathbf{u}_{LP}$ and $\mathbf{u}_{LD}$ denote the optimal solutions of the linear programming (LP) and the Lagrange dual (LD) relaxation of P0, respectively. With these two new utilities, an extension of the baseline heuristic was proposed to calculate concurrently the solution of the heuristic algorithm by adopting both $\mathbf{u}_{LP}$ and $\mathbf{u}_{LD}$ utilities and retaining the best result between them; this allowed to reach a near-optimal performance, at the cost of high computational complexity, especially considering that the dual problem P0-LD also applies a sub-gradient method.

More precisely, $\mathbf{u}_{LP}$ is the optimal solution of the LP problem, that derives

from the relaxation of $x_{b,k} \in \{0,1\}$ to $x_{b,k} \in [0,1]$ in P0. Similarly, by relaxing the constraint (4.12) that ensures no overlapping between blocks from the objective function, with the Lagrange multipliers, $\lambda_i \in \mathcal{I}$, the LD problem becomes,

$$\begin{aligned} &\min_{\lambda \geqslant 0} \left\{ \max_{x_{b,k} \in \{0,1\}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}^{(c)}} r_{b,k} x_{b,k} + \sum_{i \in \mathcal{I}} \lambda_i \left(1 - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} \alpha_{b,i} x_{b,k} \right) \right\} \\ &\text{s.t.} \quad \sum_{b \in \mathcal{B}} r_{b,k} x_{b,k} \geq q_k, \quad k \in \mathcal{K}^{(\ell)}. \end{aligned} \tag{4.12}$$

The LD problem can be decomposed for the $\mathcal{K}^{(\ell)}$ and the $\mathcal{K}^{(c)}$ services and later to be solved with a subgrandient method [189] to obtain the $\mathbf{u}_{LD}$ matrix.

Discussing the above approach, whose basic principle (with few variations) can be found in other published work, e.g., [42], we notice that despite the fact that the overall aim is to *jointly* maximize the throughput of the $\mathcal{K}^{(c)}$ services and meet the demands of the $\mathcal{K}^{(\ell)}$ services, these two interwoven goals are treated separately; in order to satisfy constraint (4.10), *first* the demands of URLLC services are met and *then* the placements of eMBB services takes place.

Such policies solve P0 by accounting only for constraint (4.10), which is suboptimal as they do not consider the *impact* of the $\mathcal{K}^{(\ell)}$ services allocation to the consequent allocation of the $\mathcal{K}^{(c)}$ services, i.e., constraint (4.11). We notice that previously proposed algorithms operate on a singe optimization target at any instance, that of maximizing first the URLLC throughout and then maximizing the eMBB throughput. Building on this observation, we will first show that the previously presented baseline heuristic can be improved, if the conflict is taken explicitly into account.

To this end, we introduce an explicit description of the impact that the assignment of any resource block to a specific service has on the feasible assignments of the remaining blocks. In other words, we account for the amount of generated conflict by any specific URLLC or eMBB resource block placement. To evaluate the impact of (4.11) explicitly, we define any conflict

(overlapping) of resource blocks as

$$c_{b,p} = \begin{cases} 1, \text{ if } \sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{B}}(\alpha_{b,i}+\alpha_{p,i}) > 1,\ i\in\mathcal{I},\ b\neq p \\ 0, \qquad \text{otherwise} \end{cases} \tag{4.13}$$

for $b, p \in \mathcal{B}$. As a next step we note that,

$$\sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}^{(c)}} r_{b,k}x_{b,k} = R_{total} - \sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{B}}\sum_{k\in\mathcal{K}} c_{b,p}x_{p,k}r_{b,k}, \tag{4.14}$$

where R_{total} denotes the maximum sum throughput of the whole resource grid with respect to $\mathcal{K}^{(c)}$ and the second triple sum represents the losses in $\mathcal{K}^{(c)}$ throughput because of the conflicts generated by the placements of all services. Given that R_{total} is constant for any particular time-frequency grid realization, the maximization of (4.9) is equivalent to the minimization of the aggregate conflict, i.e.,

$$\begin{aligned} &\max_{x_{b,k}\in\{0,1\}} \left(R_{total} - \sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{B}}\sum_{k\in\mathcal{K}} c_{b,p}x_{p,k}r_{b,k} \right) \Leftrightarrow \\ &\min_{x_{b,k}\in\ \{0,1\}} \sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{B}}\sum_{k\in\mathcal{K}} c_{b,p}x_{p,k}r_{b,k}. \end{aligned} \tag{4.15}$$

Hence, the maximization of the sum eMBB throughput may be reduced to the minimization of the potential conflicts. We also note that:

$$\mathbb{E}\left[\sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{B}}\sum_{k\in\mathcal{K}} c_{b,p}x_{p,k}r_{b,k}\right] = |\mathcal{C}|\,\bar{r}, \tag{4.16}$$

where $\mathbb{E}[\cdot]$ denotes expectation, $\mathcal{C}$ is the set of conflicts when all resource blocks have the same average throughput $\bar{r} = \mathbb{E}[r_{b,k}]$ and $|\cdot|$ denotes cardinality; i.e., from (4.15) and (4.16) it emerges that *on a large grid we need, on average, to minimize the number of conflicts.*

Considering above remarks, we propose novel heuristic algorithms for P0, focusing on minimizing the number of placements of $\mathcal{K}^{(\ell)}$ services. The first set of heuristics, dubbed in the following as conflict-aware greedy, use "conflict" enhanced variations of the utility proposed in the baseline heuristic and aim at

closing the optimality gap. The second approach is built on an interpretation of (4.15) as a bin packing optimization problem [190]; based on this approach we develop a lightweight scheduling approach that is shown to be near-optimal. It is noteworthy that this approach also allows us to find feasible solutions at very low latency, e.g., at 0.5 ms, for increasing URLLC demands.

Furthermore, as the minimization of conflicts is shown to be an equivalent optimization objective to the sum throughput maximization, we propose the use of NOMA to allow for overlapping of placements. The proposed heuristics and NOMA approaches are detailed in the next Sections.

4.5 Heuristic Algorithms and Conflict Resolution by Using NOMA

4.5.1 Conflict-aware Heuristic Solutions

We first propose extensions of the baseline heuristic, in [1], by introducing penalties in URLLC resource allocations, expressed as functions of the conflict. To this end, we introduce two metrics for the conflict induced by $\mathcal{K}^{(\ell)}$ services allocation. The aggregate conflict C_b^t,

$$C_b^t = \sum_{p \in \mathcal{B}} c_{b,p}, \;\; p, b \in \mathcal{B}, \tag{4.17}$$

that measures the total number of overlapping blocks with the block b, and, the average conflict $C_{b,k}^r$,

$$C_{b,k}^r = \sum_{p \in \mathcal{B}} \frac{c_{b,p} r_{p,k}}{C_b^t}, \;\; p, b \in \mathcal{B} \text{ and } k \in \mathcal{K}^{(\ell)} \tag{4.18}$$

that corresponds to the average throughput – for every service $k \in \mathcal{K}^{(\ell)}$ – of the blocks $p \in \mathcal{B}$ that overlap with block $b \in \mathcal{B}$. [1]

Using these new conflict measures, we propose three variations for the utility

[1] We note that despite that the average throughput - of all services $k \in \mathcal{K}$ - of the blocks $p \in \mathcal{B}$ that overlap with block $b \in \mathcal{B}$, $C_b^r = \sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{B}} \frac{c_{b,p} r_{p,k}}{C_b^t}$, should seem a more "global" conflict measure. We do not use this metric as it may be much more computational demanding, especially for a big amount of services.

matrix to be used in solving [P0]:

- In the first version the utility becomes,

$$u_{b,k}^{total} = \frac{r_{b,k}}{C_b^t}$$

- In the second variation, the utility becomes,

$$u_{b,k}^{avg} = \frac{r_{b,k}}{C_{b,k}^r}$$

- Finally, in the third variation, we use the following utility,

$$u_{b,k}^{last\ pl.} = \begin{cases} r_{b,k}, & \text{if } k = 1, \ldots, \left|\mathcal{K}^{(\ell)}\right| - 1 \\ u_{b,k}^{avg}, & \text{if } k = \{\left|\mathcal{K}^{(\ell)}\right|\}. \end{cases}$$

The utility matrix $u_{b,k}^{last\ pl.}$ is introduced to incorporate a "compromise" between the baseline and the conflict-aware approaches; notably, it considers the impact of the conflict only in the last $\mathcal{K}^{(\ell)}$ service placement, since our simulations revealed that in this last placement, usually, more blocks are required to satisfy the demands constraint.

Concluding, the set of conflict-aware heuristics allocate the URLLC services by using the $\mathbf{u}^{total}$ or $\mathbf{u}^{avg}$ or $\mathbf{u}^{last\ pl.}$ utility matrices, while for the eMBB services the utility remains unchanged. The heuristics are outlined in Algorithm 3 as pseudo-code.

4.5.2 Heuristic Inspired from Bin Packing Optimization

In the standard bin packing problem formulation, the goal is to find the optimal placement of items of different volumes in the minimum number of containers (bins) of fixed volume [190]. Although the bin packing is a combinatorial $\mathcal{NP}$-hard problem, due to it's widespread encounter in a large number of settings, various proposed heuristics have been reported in the literature with different optimality gaps. Here, we propose a novel computational efficient scheduling approach, inspired by the RFF heuristic for the standard bin packing problem.

Algorithm 3 Resource Allocation Algorithm (RA) based on [1]

Input:
; $\mathbf{u}^{(\ell)} = [u_{b,k}]$, $b \in \mathcal{B}, k \in \mathcal{K}^{(\ell)}$, utility matrix for $K^{(\ell)}$ ($\mathbf{u}^{total}$, $\mathbf{u}^{avg}$ or $\mathbf{u}^{last\ pl.}$).
; $\mathbf{u}^{(c)} = [r_{b,k}]$, $b \in \mathcal{B}, k \in \mathcal{K}^{(c)}$, utility matrix for $K^(c)$
Output: Block-service assignment $\mathbf{s}$.
repeat
    Remove from $\mathcal{B}$ the blocks in $\mathbf{s}$ and the overlapping with $\mathbf{s}$ blocks.
    $(b^{'}, k^{'}) \leftarrow \text{argmax}_{b\in\mathcal{B}, k\in\mathcal{K}^{(\ell)}}\, u^{(\ell)}_{b,k}$, $\mathbf{s} \leftarrow \mathbf{s} \cup \{(b^{'}, k^{'})\}$.
    if $q_{k'}$ is met **then**
        $\mathcal{K}^{(\ell)} \leftarrow \mathcal{K}^{(\ell)} \setminus k^{'}$.
    end if
until $\mathcal{K}^{(\ell)} = \varnothing$ **or** $\mathcal{B} = \varnothing$
if $\mathcal{K}^{(\ell)} \neq \varnothing$ **then**
    The demand of the remaining users in $\mathcal{K}^{(\ell)}$ can not be met.
end if
repeat
    Remove from $\mathcal{B}$ the blocks in $\mathbf{s}$ and the overlapping with $\mathbf{s}$ blocks.
    $(b^{'}, k^{'}) \leftarrow \text{argmax}_{b\in\mathcal{B}, k\in\mathcal{K}^{(c)}}\, u^{(c)}_{b,k}$, $\mathbf{s} \leftarrow \mathbf{s} \cup \{(b^{'}, k^{'})\}$.
until $\mathcal{B} = \varnothing$

The proposed scheduling heuristic that accounts for conflicts is summarized in Algorithm 4, jointly minimizing the number of $\mathcal{K}^{(\ell)}$ resource allocations (placements) and throughput losses for $\mathcal{K}^{(c)}$ users. Allocation of resources to $\mathcal{K}^{(\ell)}$ services and $\mathcal{K}^{(c)}$ services is treated sequentially, with the former being served first to meet the latency requirements. In the following, the vector $\mathbf{e}$ of length $|\mathcal{B}|$ has as elements the aggregated throughput losses for each allocation of a block $b \in \mathcal{B}$, i.e.,

$$e_b = \sum_{p\in\mathcal{B}} \sum_{k\in\mathcal{K}^{(c)}} c_{b,p} r_{b,k}, p, b \in \mathcal{B} \text{ and } k \in \mathcal{K}. \tag{4.19}$$

The proposed heuristic works as follows: for each $k \in \mathcal{K}^{(\ell)}$ we generate M categories (bins) with decreasing fractional sizes with respect to $q_k, k \in \mathcal{K}^{(\ell)}$, i.e., category $i \in \{1, \ldots, M\}$ is defined as the set of all resource blocks $b \in \mathcal{B}$ for which the ceiling of the ratio of the service demand over the throughput of block b is equal to i, or equivalently, category $Cat^i U^k$ contains the available resource blocks which satisfy at least $1/i$-th of the service demand q_k. Formally, we define

Algorithm 4 Bin Packing Resource Allocation Algorithm

Input:
; $\mathbf{r} = [r_{b,k}], \quad b \in \mathcal{B}, k \in \mathcal{K}$, throughput matrix
; $\mathbf{e}$, aggregated-throughput-loss vector
; $\mathbf{q}$, demand vector of URLLC services
; $\mathcal{B}$, set of all available resource blocks
Output: Block-service assignment $\mathbf{s}$.
for $k = 1$ to $|\mathbf{q}|$ **do**
    create the following categories:
    for $i = 1$ to M **do**
        Cat^iU^k = all resource blocks $b \in \mathcal{B}$ where $\lceil q_k / r_{b,k} \rceil = i$;
        Check pairwise conflicts among categorized blocks and remove the blocks with the higher aggregated-throughput-loss;
    end for
end for
► **Phase ($\mathcal{K}^{(\ell)}$ resource allocation):**
for $i = 1$ to M **do**
    select the Cat^iU^k which has the least number of blocks;
    if ($|Cat^iU^k| \geq i$ and q_k is not already met) **then**
        $\mathcal{B}' \leftarrow$ (select i number of blocks in Cat^iU^k with the least aggregated-loss-value);
        $\mathbf{s} \leftarrow \mathbf{s} \cup (b', k')$, $k' = i, \forall b' \in \mathcal{B}'$;
        Remove from $\mathcal{B}$ the blocks in $\mathbf{s}$ and those overlapping with the blocks in $\mathbf{s}$;
        if q_k is met **then**
            $\mathcal{K}^{(\ell)} \leftarrow \mathcal{K}^{(\ell)} \backslash \{k'\}$;
        end if
    end if
end for
► **Phase ($\mathcal{K}^{(c)}$ resource allocation):**
repeat
    $(b', k') \leftarrow \arg\max_{b \in \mathcal{B}, k \in \mathcal{K}^{(c)}} r_{b,k}$;
    $\mathbf{s} \leftarrow \mathbf{s} \cup (b', k')$;
    Remove from $\mathcal{B}$ the blocks in $\mathbf{s}$ and those overlapping with the blocks in $\mathbf{s}$;
until $\mathcal{B} = \emptyset$

$$Cat^iU^k = \left\{ b : \left\lceil \frac{q_k}{r_{b,k}} \right\rceil = i, \forall b \in \mathcal{B} \setminus \{Cat^jU^k\}_{j=1,\ldots,i-1} \right\},$$
$$k \in \mathcal{K}^{(\ell)}, i \in \{1, \ldots, M\}, \tag{4.20}$$

where $\lceil x \rceil$ denotes the smallest integer bigger or equal to x.

For example, Cat^1U^1 is the category of the blocks which individually satisfy the whole demand of the URLLC service $k = 1$. Therefore, the categories created for service $k \in \mathcal{K}^{(\ell)}$ range from Cat^1U^k – containing the most valuable blocks (valuable in terms of throughput $r_{b,k}$) – till Cat^MU^k, containing the least valuable blocks in order. Note that i) we need *at most* i elements from Cat^iU^k to satisfy the demand q_k of service $k \in \mathcal{K}^{(\ell)}$; ii) categories might be empty, so M needs to be defined according to the expected throughput per mini-slot as well as its variance. In our numerical results that are provided in Section 4.6 we have set $M = 10$.

Inside each category, we subsequently introduce a further minimization problem in order to select the elements from each category that incur the minimum loss to eMBB, i.e.,

$$\begin{aligned} &\min_{y_b \in \{0,1\}} \sum_b e_b y_b, \ b \in (\mathcal{K}^{(l)} \cap Cat^iU^k) \\ &\text{s.t} \quad \sum_{b \in Cat^iU^k} y_b \leq i. \end{aligned} \tag{4.21}$$

Note that if (4.21) is interpreted as a knapsack problem, each element of a given category has the same weight (equal to unity), while the values (losses in the specific instance) differ. Similar problems are encountered in different settings, e.g., the subcarrier resource allocation in [157]. Exploiting these previous results, we reproduce a simple heuristic according to which the elements of each category are *re-ordered*[2] in increasing aggregated loss $e_b, b \in \mathcal{B}$. Subsequently, the first i elements of category Cat^iU^k are allocated to URLLC.

As an example, after this step, the first element of Cat^1U^k is the resource block that can simultaneously cover the demand q_k of URLLC service k while incurring the least aggregate losses for the eMBB users. The joint minimization of the number of $\mathcal{K}^{(\ell)}$ placements and the losses due to conflicts is achieved simply by assigning to service $k \in \mathcal{K}^{(\ell)}$ the first i elements of Cat^iU^k, starting from $i = 1$, i.e., the allocation for demand q_k starts from Cat^1U^k. As explained before, the most valuable categories in terms of throughput satisfy URLLC services by using the least number of resource blocks and result in the minimum

[2]The ordering has a complexity $\mathcal{O}\big(\max_{i,k}\{|Cat^iU^k|\log(|Cat^iU^k|)\}\big)$.

number of $\mathcal{K}^{(\ell)}$ placements, that is expected on average to incur the minimum losses due to conflicts. Furthermore, having re-ordered the elements of each category in increasing eMBB loss value, we jointly account for both constraints (4.10) and (4.11) in one go. After each allocation, the allocated blocks are removed from $\mathcal{B}$ and all other categories. This procedure is repeated until the demand of all of the $\mathcal{K}^{(\ell)}$ services are satisfied or no more blocks remain in the categories.

In the last phase of the algorithm, the resource allocation to $\mathcal{K}^{(c)}$ services takes place. This is performed by selecting the block-service pairs with the highest throughput $r_{b,k}, b \in \mathcal{B}, k \in \mathcal{K}^{(c)}$) from the *remaining* available blocks. The latter have not been allocated to a URLLC service, since once a block is allocated it is removed from $\mathcal{B}$. This step is iterated until no more blocks remain available.

4.5.3 NOMA for Downlink Scheduling

In this Subsection we re-examine P0 under the assumption that it is possible to employ NOMA in the downlink to schedule different services, even at the mini-slot level [43]. In contrast to the scheduling optimization problem as formulated in P0, NOMA allows overlapping amongst the blocks, either full or partial (of some mini-slots). In light of this, P0 is reduced to an corresponding linear programming (LP) problem that we refer to as P1, in which the optimization parameter is now a real number $x_{b,k} \in [0, 1]$,

$$[\text{P1}] \quad \max_{x_{b,k} \in [0,1]} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}^{(c)}} r_{b,k} x_{b,k}, \tag{4.22}$$

$$\text{s.t.} \quad \sum_{b \in \mathcal{B}} r_{b,k} x_{b,k} \geq q_k, \quad k \in \mathcal{K}^{(\ell)}, \tag{4.23}$$

$$\sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} a_{b,i} x_{b,k} \leq \tilde{r}, \quad i \in \mathcal{I}, \tag{4.24}$$

where $\tilde{r}$ denotes the NOMA (normalized) sum throughput per block, which in the literature [191] has been shown to be superior to OMA (i.e., greater than unity); note that in P0, constraint (4.11) is upper bounded to unity. This points out a further gain in using NOMA due to the increase in per resource block utilization. However, as in this work we aim primarily at demonstrating

the gains brought about due to conflict avoidance, in the numerical results we simply use $\tilde{r} = 1$.

P1 can be efficiently solved, guaranteeing the optimal solution, by applying the simplex method, interior point methods, or the ellipsoid method [161], with respect to the infeasible solutions. In the specific problem, it is preferable to solve the dual instead of the primal problem, since the solution time increases much more rapidly with the number of constraints in the problem than with the number of variables. Moreover, the ellipsoid method and the interior point methods are mathematically iterative and need many more computing resources than the simplex algorithm for small linear programming problems. Hence, we suggest the dual simplex algorithm, that in the worst case scenario, i.e., in which an exponential number of corners exist, is able to take an exponential number of steps to find the optimal corner.

4.6 Numerical Results

In this section, we start with our numerical analysis of both OMA and NOMA schemes, under the usage of different 5G URLLC configurations and numerologies; fixed, multiple-fixed and flexible numerology. This exercise allows us to highlight the importance of flexible numerology, while motivating NOMA as a conflict mitigation approach. Here, we mainly focus on the conflicts aspect, rather than on deployment feasibility and power issues, which are important enough to deserve an independent study. We then move on to a comparative analysis of the proposed heuristic Algorithms 3 (conflict aware, CA) and 4 (bin packing based, BPB) for OMA, with a goal to support the potential of the proposed conflict aware scheduling.

We use the simulation setup given in [1], implemented based on the control channel overhead model for supporting the flexible numerology defined in [192] and considers the effect of guard band (i.e., of the cyclic prefix) on the achievable data rate by blocks, as modeled in [193]. The computation of the achieved throughput per block $r_{b,k}$ relies on the configuration of block b (see Table II), with a total number of nine multipath channel profiles [194], calculating the throughput based on the model introduced in [195]. The latter considers the intersymbol-interference (ISI) depending on CP, and approximates the

inter-channel interference (ICI) between the neighboring subbands, with the same type of numerology.

Regarding our simulation parameters, a time-frequency grid with a 2ms and 2 MHz domain, and a 16×11 shape is assumed; which indicates a set of $\mathcal{I} = \{1, \ldots, 176\}$ mini-slots. This denotes a set of $\mathcal{B} = \{1, \ldots, 549\}$ candidate blocks with respect to the numerology, where every candidate block consists of 4 elements of $\mathcal{I}$. The resource block details are given in Table II. Blocks of shape 1 (4×1), $\mathcal{B}_1 \subset \mathcal{B}$, include a multitude of $|\mathcal{B}_1| = 143$ resource blocks. Blocks of shape 2 (2×2), $\mathcal{B}_2 \subset \mathcal{B}$, include a multitude of $|\mathcal{B}_1| = 150$ resource blocks. Finally, blocks of shape 3 and 4 (1×4), $\mathcal{B}_3, \mathcal{B}_4 \subset \mathcal{B}$ include the same multitude of blocks $|\mathcal{B}_3| = |\mathcal{B}_4| = 128$. Furthermore, we consider 10 users in total, 5 URLLC and 5 eMBB, with $|\mathcal{K}^{(c)}| = |\mathcal{K}^{(\ell)}| = 5$. Moreover, the chosen latency tolerance and bit rate demands for the URRLC users are $\tau = \{0.25, 0.5, 1, 1.5, 2\}$ ms and $q = \{16, 32, 64, 128, 256, 512\}$, respectively. The latency tolerance for the eMBB users is fixed and equal to $\tau = 2$ ms. The SNR range is generated by numbers uniformly distributed in the interval $[5, 30]$ (dB). Our references to the "optimal solution" in the following text corresponds to the solution provided by the Gurobi optimization solver and is used as a benchmark for the optimality gap of the proposed heuristics. Finally, the outputs of all the simulation results are assessed over N=200 Monte Carlo simulations.

4.6.1 Performance comparison between NOMA and OMA scheduling under different numerologies

First, we compare the outcome of OMA and NOMA schemes for different numerologies. In the case of fixed numerologies, shape 1 (horizontal), shape 2 (square) and shape 3 (vertical) type of blocks are considered separately. Furthermore, capturing a common scenario in practical systems, we define as the multiple-fixed numerology the one in which eMBB uses resource blocks of shape 1 (horizontal) and URLLC of shape 3 (vertical). Finally, in the case of flexible numerology all type of shapes, given in Table II, are available to all services.

In Fig. 4.2, the sum bit rate for the eMBB services, $\mathcal{K}^{(c)}$ when applying the optimal i) NOMA and ii) OMA scheduling are shown. The NOMA sum

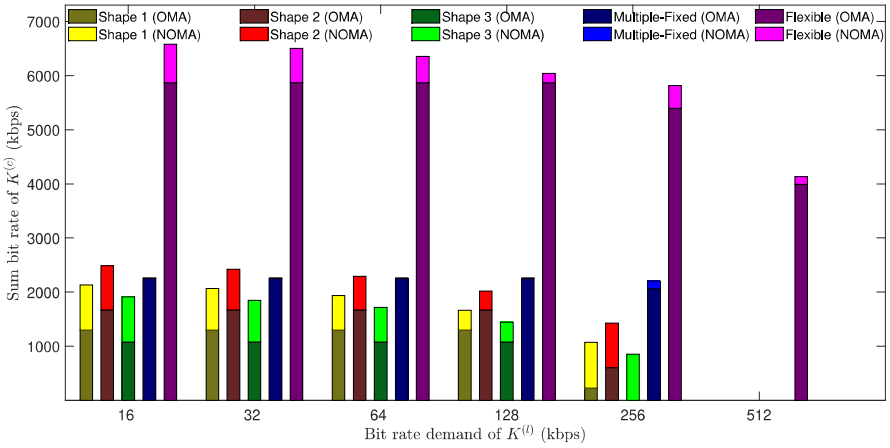


Figure 4.2: Sum bit rate for $\mathcal{K}^{(c)}$ services when employing NOMA and OMA for fixed, multiple and flexible numerology, under several q_k data demands and delay tolerance value $\tau_k = 1$ ms, $k \in \mathcal{K}^{(\ell)}$. The lighter colors depict the NOMA sum bit rate gains in comparison to the OMA. Fixed and multiple-fixed numerologies result in infeasible outputs for $q_k = 256$ kbps and $q_k = 512$ kbps, i.e., it is infeasible to satisfy all URLLC demands using these numerologies. On the other hand, flexible numerology does not suffer from infeasibility even for $q_k = 512$ kbps. The tremendous gains in using flexible numerology are consistent across all service demand scenarios. The gains in using NOMA are more accentuated in lower URLLC demands.

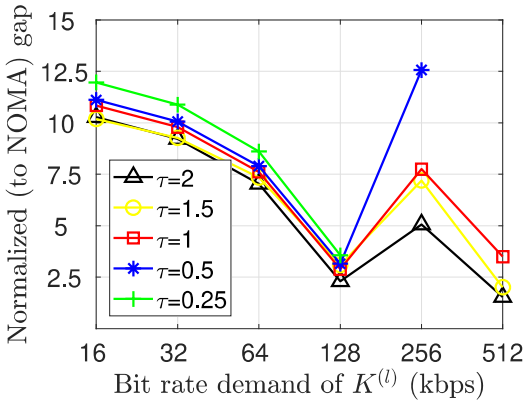
bit rate gains to the OMA are depicted with the lighter color in each bar. The latency tolerance and bit rate demands considered are $\tau = 1$ ms and $q = \{16, 32, 64, 128, 256, 512\}$ kbps, respectively, for five $\mathcal{K}^{(\ell)}$ and five $\mathcal{K}^{(c)}$ users. In all cases, as expected, flexible numerology significantly outperforms the fixed and multiple-fixed numerology. Moreover, multiple-fixed overpasses the performance of fixed numerology in the OMA case. From these results it becomes apparent that flexible numerology in combination with NOMA can offer distinct gains across varying URLLC demands. Notably, as the URLLC demands increase, flexible numerology is the only approach that avoids infeasibility issues, i.e., not covering all of URLLC demands.

Focusing on the comparison between OMA and NOMA, the NOMA consistently outperforms OMA. More precisely, NOMA based scheduling is shown to increase particularly the sum throughput of eMBB users under fixed numerology, although NOMA also improves the overall performance when using flexible numerology as well. On the other hand, NOMA does not affect the

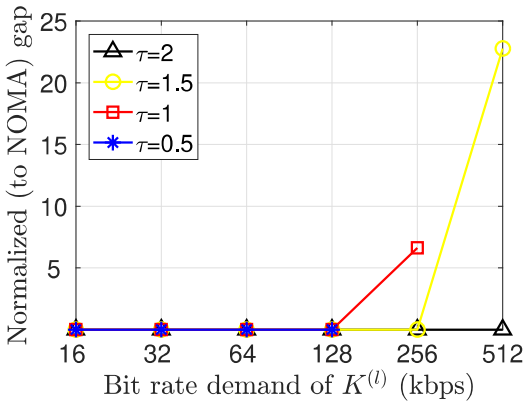
performance under multiple-fixed numerology; this is due to the fact that in the specific grid used in the simulations, overlapping of blocks is limited in the case of multi-fixed numerology.

Furthermore, in Fig. 4.3, the normalized to NOMA performance gap between OMA and NOMA (expressed as a percentage) is shown, for different numerologies (for compactness of presentation, only shape 2 is considered in the case of fixed numerology). The superiority of NOMA is reconfirmed both for fixed and flexible numerology, for different values of the URLLC latency tolerance $\tau_k = \{0.5,\ 1,\ 2\}$ ms, $k \in \mathcal{K}^{(\ell)}$. Finally, in the case of flexible numerology, the lower the delay tolerance τ_k, the higher the gains in using NOMA as opposed to OMA. The performance fluctuations, illustrated in Fig. 4.3, are strongly related to the different values of the bit rate demands $q_k,\ k \in \mathcal{K}^{(\ell)}$. More precisely, after a close inspection of the simulation outputs, we came to the conclusion that the gap between the demand of a service $k \in \mathcal{K}^{(\ell)}$ and the achievable throughput of the block, in which the service is allocated, plays an important role. A higher gap between the two corresponds to a decisive reduction of the overall available throughput for the scheduling of the $\mathcal{K}^{(c)}$ services in the OMA case, which in turn offers a crucial advantage to the NOMA scheme that allows overlaps.

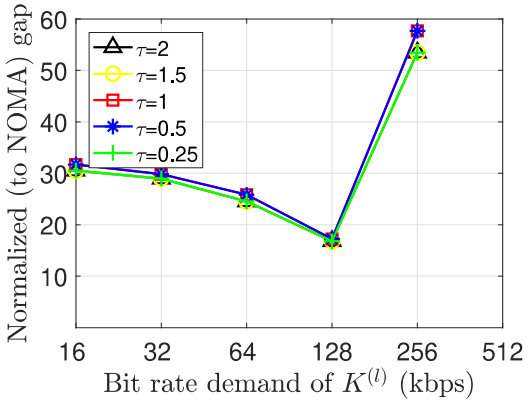
In Fig. 4.4 the overall scheduling output of all services is depicted in the case of OMA and NOMA, for $\tau_k = \{0.25, 1, 2\}$ ms and $q_k = \{32, 256, 512\}$ kbps, respectively, for all $k \in \mathcal{K}^{(\ell)}$. In the case of OMA and a small value of $q_k,\ k \in \mathcal{K}^{(\ell)}$ depicted in Fig. 4.4(a), overlapping of resource blocks is not allowed, while on the other hand in the case of NOMA, depicted in Fig. 4.4(b), the opportunity of overlapping resource blocks increases the available resource blocks for eMBB allocation; notice also that the choice of resource blocks assigned to URLLC is different. Similar outcomes are depicted in Figs. 4.4(c), (e), OMA case, and Figs. 4.4(d), (f), NOMA case, in which higher values of $q_k,\ k \in \mathcal{K}^{(\ell)}$ are considered. In this case, though, to meet the higher bit rate demands of the URLLC services, more resource blocks are allocated to them, e.g., for $q_k = 512$ kbps (Figs. 4.4(e) and (f) almost the half of the resource blocks are used to mitigate the URLLC demands.



(a) Flexible

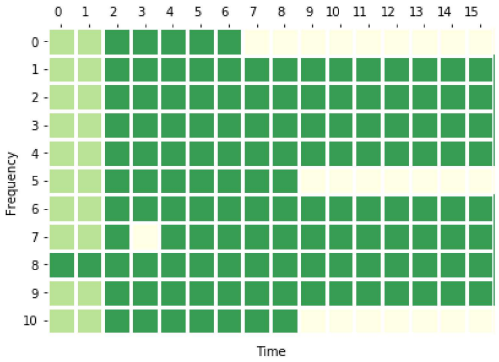


(b) Multiple-fixed

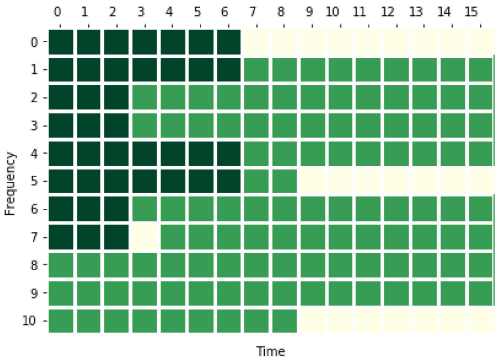


(c) Fixed (Shape 2)

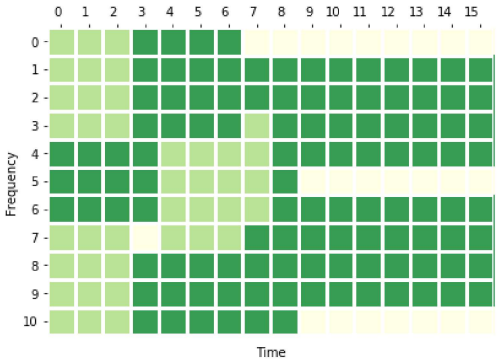
Figure 4.3: Normalized (to NOMA) gap of the sum bit rate of the $\mathcal{K}^{(c)}$ services between NOMA and OMA schemes. The y-axes measure percentages. Non existing values indicate infeasible solutions. We exclude the delay tolerance value $\tau_k = 0.25$ ms, $k \in \mathcal{K}^{(\ell)}$ from fixed and multiple-fixed numerology results, since they provide infeasible solutions for both OMA and NOMA schemes.



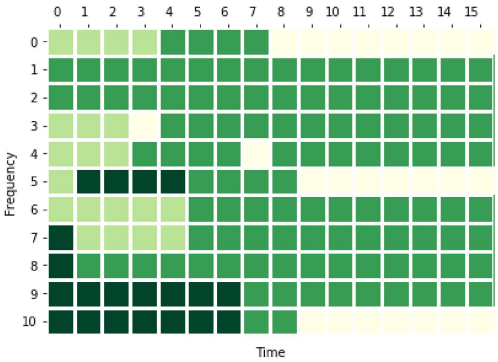
(a) OMA case, $\tau_k = 0.25$ ms and $q_k = 32$ kbps.



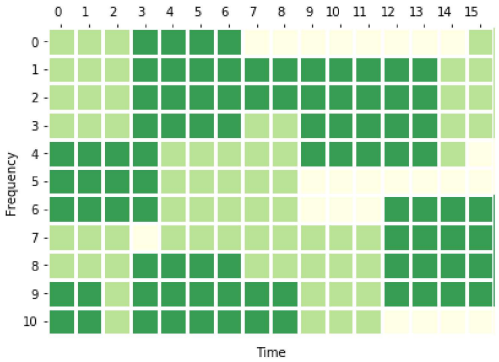
(b) NOMA case, $\tau_k = 0.25$ ms and $q_k = 32$ kbps.



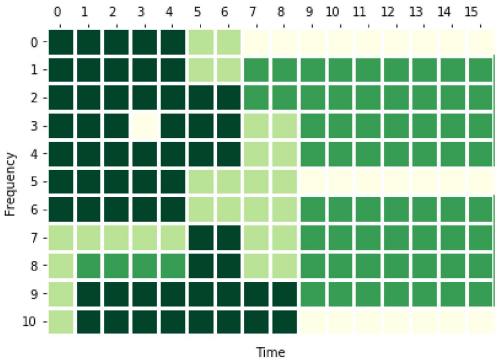
(c) OMA case, $\tau_k = 1$ ms and $q_k = 256$ kbps.



(d) NOMA case, $\tau_k = 1$ ms and $q_k = 256$ kbps.



(e) OMA case, $\tau_k = 2$ ms and $q_k = 512$ kbps.

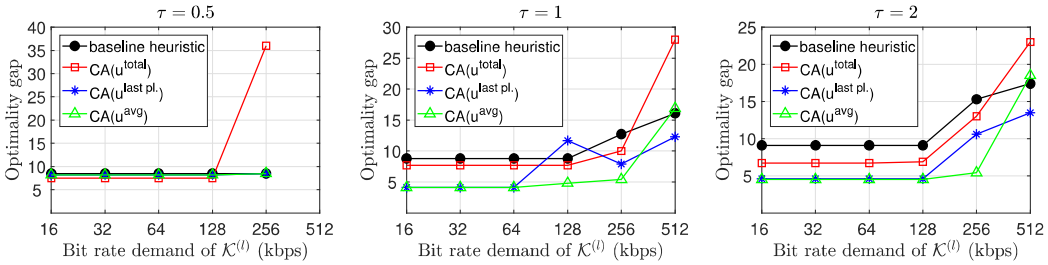


(f) NOMA case, $\tau_k = 2$ ms and $q_k = 512$ kbps.

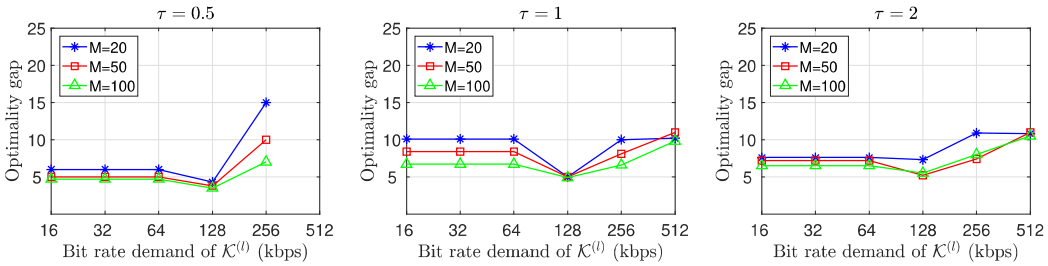
Figure 4.4: Resource allocation of URLLC (light green) and eMBB (green) services, for OMA (first column) and NOMA (second column). Light yellow denotes zero throughput mini-slots. Dark green denotes overlapping of mini-slots thanks to using NOMA.

4.6.2 Performance of proposed heuristic algorithms

First, as a validation step, we evaluate and compare the optimality gaps of the baseline heuristic (presented in [1]) and the proposed conflict aware heuristics with utilities ($\mathbf{u}^{total}$, $\mathbf{u}^{avg}$ and $\mathbf{u}^{last\ pl.}$), denoted by $CA(\cdot)$ with input one of the corresponding utility matrices, against the global optimum of P0. Then, we provide additional results with all proposed heuristics employing flexible numerology.



(a) Baseline heuristic and CA heuristics.



(b) Baseline heuristic, using the LP-LD utility matrices, for several thresholds M.

Figure 4.5: a) Optimality gaps: a) of the baseline heuristic [1] and the variations of the conflict-aware heuristic CA, and, b) of the baseline LP-LD heuristic and thresholds for the sub-gradient iterations $M = \{20,\ 50,\ 100\}$. Against the global optimum of P0, for latency tolerance values $\tau_k = \{0.5,\ 1,\ 2\}$ ms. The y-label express the relative deviation to the optimum, expressed as percentage.

Fig. 4.5 depicts the optimality gap: i) of the baseline and the variations of the conflict-aware utilities (first row), and, ii) the utilities resulting from the LP-LD relaxation of P0 (second row), for several values of maximum sub-gradient iterations, with respect to the bit rate demand and the latency tolerance of the $\mathcal{K}^{(l)}$ services. In the first column of Fig. 4.5 the conflict-aware approaches are shown, in most cases, to outperform the baseline heuristic approach for higher latency tolerance values, see Figs. 4.5(b)-(c), and to provide similar results for lower latency tolerance values, Fig. 4.5(a). $CA(u^{avg})$ and $CA(\mathbf{u}^{lastpl.})$ heuristics

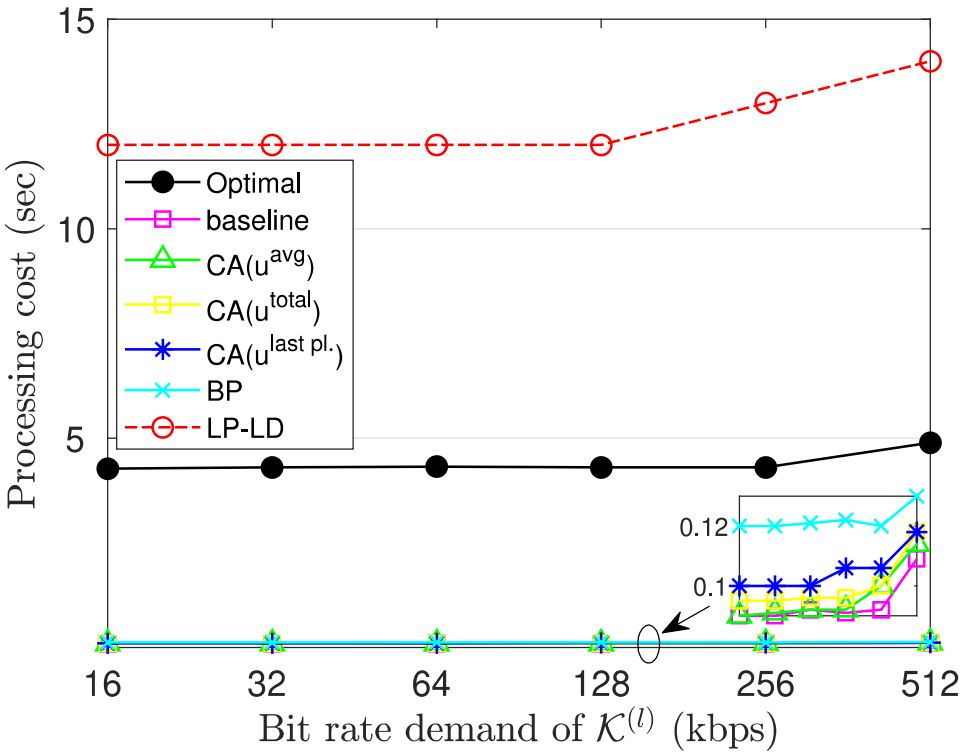


Figure 4.6: The processing cost of: i) the optimal, ii) the baseline heuristic variations, iii) the bin packing based approach, and, iv) the LP-LD (M=20), for $\tau = 1$ ms and $q_k = \{16,\ 32,\ 64,\ 128,\ 258,\ 512\}$ (kbps).

lead to clearly better results than the $CA(\mathbf{u}^{total})$, providing a gap below 5%, for users bit rate demands up to 64 kbps.

Especially the $CA(\mathbf{u}^{avg})$ maintains an optimality gap below 5% for bit rate demands up to 256 kbps, except for $\tau_k = 0.5$ ms, where the gap is constant and below 10%. However, for bit rate demands of 512 kbps, $CA(\mathbf{u}^{avg})$ and $CA(\mathbf{u}^{total})$ are inferior to $CA(\mathbf{u}^{lastpl.})$ and the baseline heuristics; note that in the case of $CA(\mathbf{u}^{total})$ the optimality gap increases to more than 23%, depending on the latency tolerance value. The above indicate that for high bit rate demands a placement policy that gives a significant weight to the overlapping blocks aspect leads to increased $\mathcal{K}^{(\ell)}$ block placements to satisfy the URLLC demands. In such cases, a more balanced (conservative) policy, like $CA(\mathbf{u}^{last\ pl.})$, seems more suitable. Overall, $CA(\mathbf{u}^{avg})$ emerges as an appropriate choice for lower bit rate demands of URLLC users (the gap is below 5%) and $CA(\mathbf{u}^{lastpl.})$ is useful for higher bit rate demands (the gap is below 13%, in all cases).

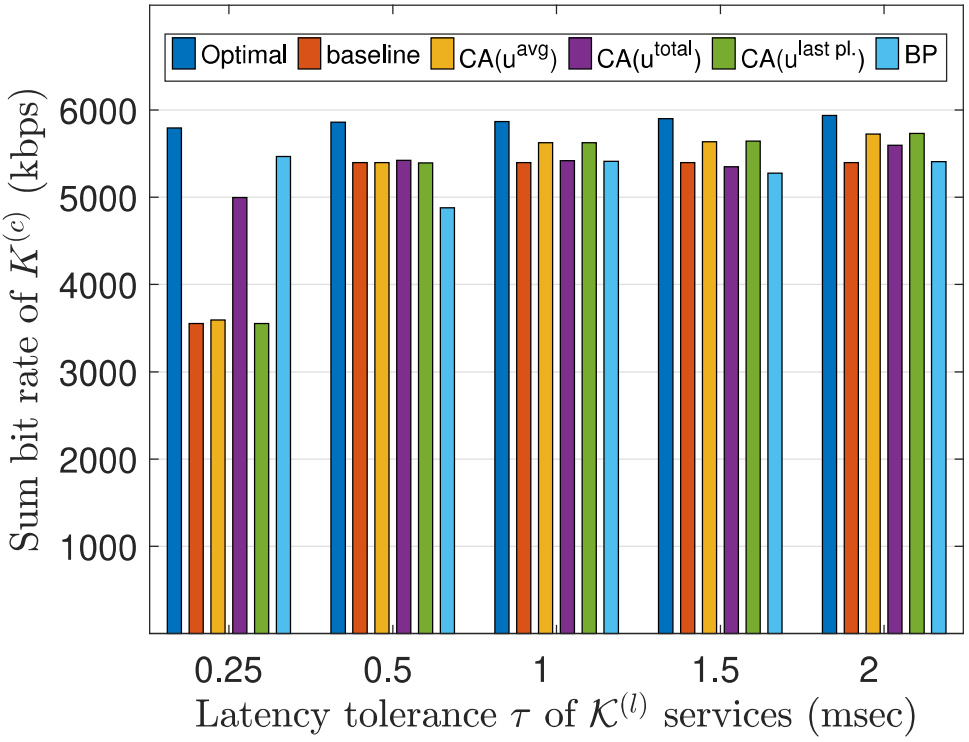


Figure 4.7: Sum bit rate of $\mathcal{K}^{(c)}$ services for various values for the latency tolerance of $\mathcal{K}^{(\ell)}$ services, when the bit rate demands of $\mathcal{K}^{(\ell)}$ users are all equal and set to 64 kbps. Similar results are produced for demands of 16 and 32 kbps.

The second row of Fig. 4.5 depicts the optimality gap of the heuristic solutions compared to the LP-LD heuristic using the relaxed P0 problem, for various threshold values $M = \{20, 50, 100\}$ for the maximum sub-gradient iterations. We do not provide the solutions coming from the incorporation of the utility matrices $\mathbf{u}_{LP}$, $\mathbf{u}_{LD} \in \mathbb{R}_{\mathcal{B}\times\mathcal{K}}$, since all variations conclude in similar results.

As it is expected, higher threshold values of M lead to a further reduce of the optimality gap, although at the cost of a higher computational time. On the other hand in all cases the heuristics are shown to maintain the optimality gap to below 10% even for $q = 512$ kbps, except for $q = 256$ kbps and $\tau = 0.5$ ms, for $M = 20$. Note that $CA(\mathbf{u}^{avg})$ heuristic's results are very close to that of the LP-LD variations for low throughput demands, as can be seen by comparing the left and the right column of Fig. 4.5. On the other hand, the

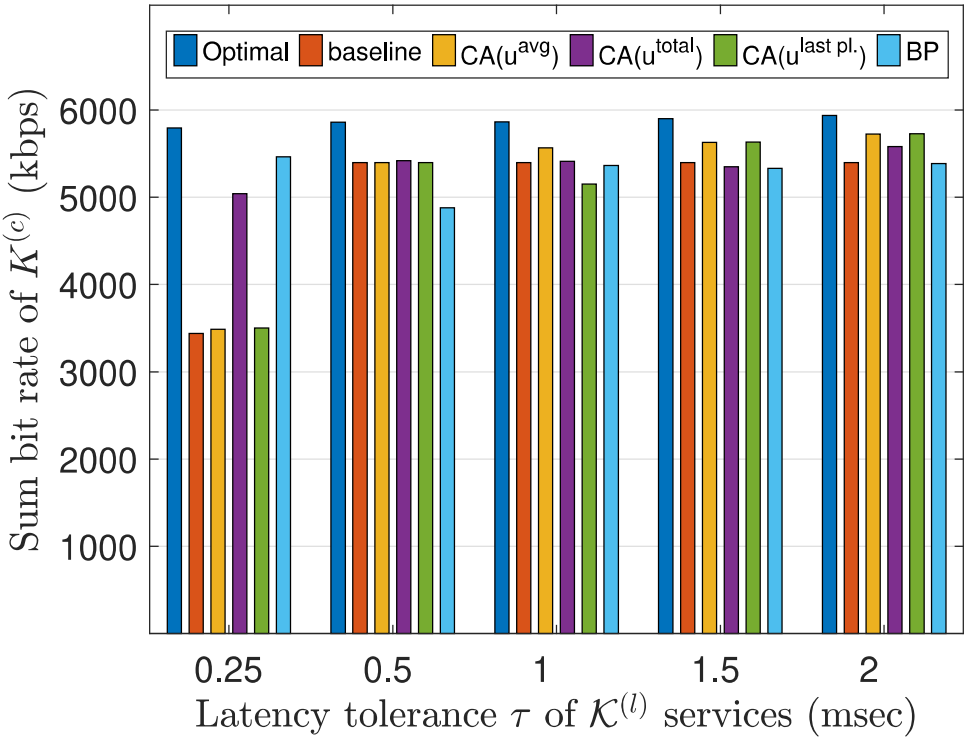


Figure 4.8: Sum bit rate of $\mathcal{K}^{(c)}$ services for various values for the latency tolerance of $\mathcal{K}^{(\ell)}$ services, when the bit rate demands of $\mathcal{K}^{(\ell)}$ users are all equal and set to 128 kbps.

reduction of the optimality gap using LP-LD utility matrices comes with a significant increase of the computational time, highlighting even more the usage of the $CA(\mathbf{u}^{avg})$ heuristic, that achieves an optimality gap of less than 5% for throughput demands up to 256 kbps.

Furthermore, we utilize our implementation to quantify the performance of the optimal and the heuristic approaches, in terms of processing cost. The computational time is measured on a Lenovo IdeaPad 510-15IKB laptop, with an Intel Core i7-7500U @ 2.70 GHz processor and 12 GB RAM. In Fig. 4.6, we depict the processing cost of: i) the optimal solution, ii) the baseline heuristic variations (without the usage of the LP-LD utilities), iii) the bin packing based approach, and, iv) the LP-LD heuristic with threshold value $M = \{20\}$, for $q_k = \{16, 32, 64, 128, 256, 512\}$ and a conventional latency tolerance value $\tau = 1$ ms. As it is depicted, the LP-LD solution is much more computational intensive than other heuristic approaches, even from the optimal solution. Note that

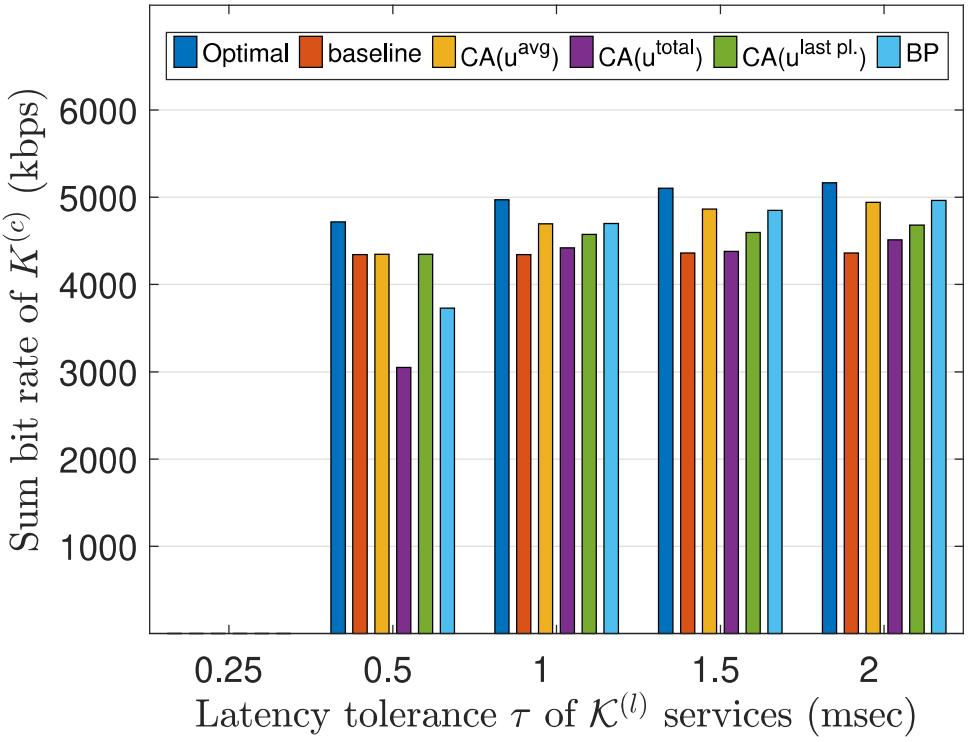


Figure 4.9: Sum bit rate of $\mathcal{K}^{(c)}$ services for various values for the latency tolerance of $\mathcal{K}^{(\ell)}$ services, when the bit rate demands of $\mathcal{K}^{(\ell)}$ users are all equal and set to 256 kbps.

higher threshold values increase drastically the processing cost, e.g., for $M = 50$ and $M = 100$ the processing cost is of 20 and 40 sec, respectively. On the other hand, the processing cost of the bin packing and the conflict-aware heuristics is between 0.9 and 0.12 sec, indicating their low computational nature; we remind that the complexity of the conflict-aware and bin packing based heuristics is of $\mathcal{O} = N \log(N)$.

Next, we compare the performance of the conflict-aware heuristic solutions (Algorithm 1), the heuristic algorithm inspired from the reformulation of the scheduling problem as a bin packing optimization (Algorithm 2), the baseline heuristic and the optimal solution. We exclude the CA approaches based on the LP-LD utility matrices from the comparison, as these come at the cost of a significantly higher complexity. Fig. 4.7 depicts a comparable performance of the heuristic algorithms to the global optimum (obtained through Gurobi solvers), while keeping the complexity very low. Note that bin packing algorithm,

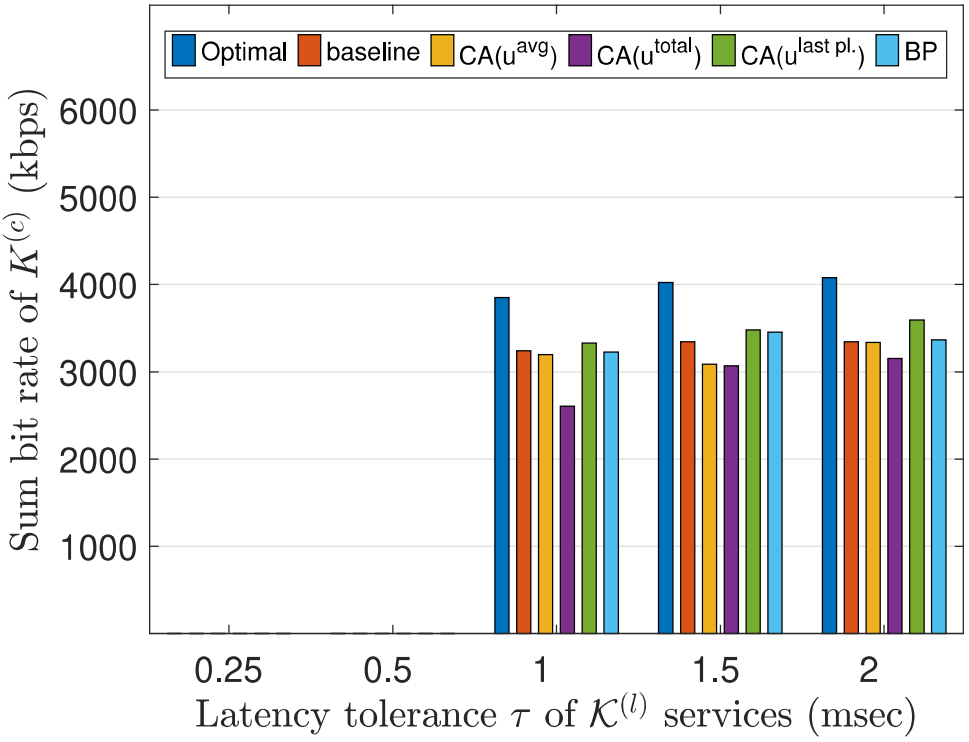


Figure 4.10: Sum bit rate of $\mathcal{K}^{(c)}$ services for various values for the latency tolerance of $\mathcal{K}^{(\ell)}$ services, when the bit rate demands of $\mathcal{K}^{(\ell)}$ users are all equal and set to 512 kbps.

with no utility computation and with near-linear complexity, achieves a near-optimal performance. This showcases that indeed, the reformulation of the optimal scheduling as a conflict minimization problem is highly pertinent and allows shedding light on how to jointly address the constraints (4.10) and (4.11) of P0. It is also noteworthy that more elaborate heuristics could be proposed in the same context, by looking at algorithms with lower optimality gaps to the optimal bin packing solution. The same conclusions can be reached in Figs. 4.8 and 4.9 for URLLC demands of 128 and 256 kbps, respectively.

Moreover, in case of higher bit rate demands for the URRLC users of 512 and 1024 kbps (Figs. 4.10 and 4.11), the bin packing based approach seems to exceed the performance of $CA(\mathbf{u}^{total})$ and $CA(\mathbf{u}^{avg})$ providing a performance similar to that of of the $CA(\mathbf{u}^{lastpl.})$ heuristic. Note that, with increasing URLLC service demands, there are many instances of infeasibility, i.e., cases in which not all of the —URLLC demands can be satisfied. Furthermore, in

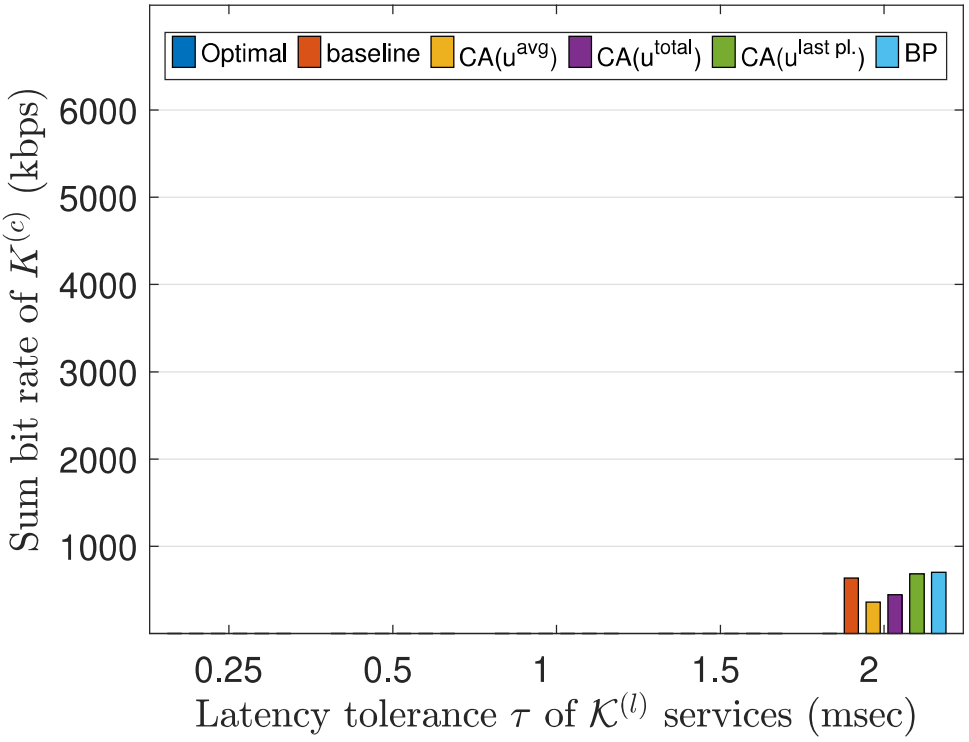


Figure 4.11: Sum bit rate of $\mathcal{K}^{(c)}$ services for various values for the latency tolerance of $\mathcal{K}^{(\ell)}$ services, when the bit rate demands of $\mathcal{K}^{(\ell)}$ users are all equal and set to 1024 kbps.

Fig. 4.11 (i.e., with $q = 1024$ kbps), the optimal solution is not depicted, because no solution was returned within the predetermined execution time limit of the chosen solver (Gurobi). Last but not least, the conflict-aware heuristic solutions based on the variations $CA(\mathbf{u}^{avg}$ and $CA(\mathbf{u}^{avg}$ exceed in all cases the performance of the baseline heuristic, as it is also shown in Fig. 4.5. Furthermore, the BP approach overpass the performance of the baseline heuristic in most cases, especially for higher bit rate demands of the $\mathcal{K}^{(\ell)}$ users.

4.7 Conclusions

In 5G and beyond networks, URLLC services will coexist with eMBB services through challenging layer 2 scheduling. To address the latter, we have reformulated the standard eMBB throughput maximization problem as an equivalent conflict minimization, which points at minimizing the overall amount of conflicts.

Building on this premise, two lightweight and efficient scheduling approaches were proposed: a family of conflict-aware heuristics that employ conflict aware utilities and a heuristic inspired by the bin packing problem.

In addition to the proposed scheduling using orthogonal multiple access (OMA), we further proposed the use of non-orthogonal multiple access (NOMA) to mitigate conflicts. We investigated the potential advantages of allowing for non-orthogonal sharing of radio resources with flexible numerology and frame structure. The intuition for NOMA's superior performance, as a result of alleviating conflicts, was demonstrated to hold; importantly, NOMA can potentially offer significant advantages particularly in the case of ultra-low latency constraints for the URLLC users.

Extensive simulations were performed for URLLC services with different QoS requirements both for OMA and NOMA scenarios. The simulation results showed that i) all of the proposed heuristics have near-optimal performance, demonstrating that conflict minimization is indeed key to layer 2 scheduling, and, ii) there are significant gains in terms of resource utilization when employing NOMA.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This final Chapter concludes the thesis, presenting and summarizing its scientific contributions, and describing relevant future research directions. In brief, we examined problems related to aspects of 5G and beyond networks, which we studied considering algorithms from the realms of statistical and optimization analysis.

Our first contribution is the real time change point detector (RCPD), a novel algorithm for the real-time detection of changes in the mean value of content popularity. Approaching the problem statistically, we efficiently combined off-line and on-line non-parametric cumulative sum (CUSUM) procedures to avoid restrictive assumptions for content popularity behavior and to reduce the overall computational cost. We divided the algorithm in two phases. The first phase was an extended retrospective (off-line) procedure with a modified binary segmentation (BS) algorithm and was used to adjust on-line parameters, based on historical data of the particular video. The second phase integrated one of two alternative trend indicators to the sequential (on-line) procedure, to reveal the direction of a detected change. We provided extensive simulations, using synthetic and real data, that demonstrated the performance of the proposed algorithm for the successful identification of content popularity changes in real-time.

We also extended the RCPD algorithm for the detection of changes in the variance of a time series. In accordance with RCPD, we combined an off-line

approach, i.e., during which algorithmic and model parameters were learned, and an on-line part, i.e, where changes in the variance of the time series were identified using a stopping time procedure. Whenever the value of the test statistic surpassed a predefined critical value, a change was declared. We proposed three different approaches for the test statistic: i) a non-parametric, ii) a parametric using an autoregressive moving average (ARMA) model, and, iii) a parametric using a nonlinear generalized autoregressive conditional heteroskedasticity (GARCH) model. Our studies using synthetic data indicated that the ARMA parametric approach did not generalize well. Due to this fact, we only performed experiments on real data using the non-parametric and the GARCH approaches. We concluded that both can equally well identify large deviations in the variance and that, in the general case, the non-parametric approach was able to provide quicker detection of CPs in the datasets studied in this work.

Applications that incorporate RCPD were also considered in this thesis. In this context, we proposed the unikernel-based content distribution network (UNIC) platform, an elastic content distribution facility motivated by the unique requirements of 5G networks evolution (e.g., ultra low-delays and flexibility), incorporating mechanisms for content popularity detection. Finally, we provided proof-of-concept results validating the efficient and elasticity behavior of UNIC, in terms of content delivery performance and server resource utilization. We also performed experiments for two software defined wireless sensor networks (SDWSN) distributed denial-of-service (DDoS) attacks, in topologies of 36 and 100 nodes. The attacks implemented were the false data flow forward (FDFF) and the false neighbor information (FNI). Our results showed that it is feasible to detect those attacks by monitoring either the data packets delivery rate or control packets metrics. However, targeting the quickest detection possible, far superior detection performance was achieved for the FDFF, when monitoring the control packets overhead. Conversely, results showed a significantly better performance in detecting the FNI attack, when monitoring the data packets delivery rate. In either cases, the agility of detection was noteworthy, with either attack identified within 3-10 samples from its launch.

Furthermore, layer 2 resource allocation scheduling considering ultra reliable low latency communications (URLLC) and enhanced mobile broadband (eMBB)

coexistence is a known challenging task. To address this, we reformulated the standard eMBB throughput maximization problem as an equivalent conflict minimization. Building on this premise, two conflict-aware heuristics were proposed. The first was based on novel, conflict-aware utilities while the second was inspired by a simple heuristic for bin packing problems.

In addition to the proposed scheduling using an orthogonal multiple access (OMA) approach, NR evolution also considers non-orthogonal multiple access (NOMA). We investigated the potential advantages of allowing for non-orthogonal sharing of radio resources with flexible numerology and frame structure. The intuition for NOMA's superior performance, as a result of alleviating conflicts, was demonstrated to hold; importantly, NOMA can potentially offer significant resource allocation efficiency advantages, particularly in the case of ultra-low latency constraints for the URLLC users.

Extensive simulations were performed for URLLC services with different QoS requirements both for OMA and NOMA scenarios. The simulation results showed that: i) the proposed conflict-aware heuristic algorithm surpassed in terms of eMBB throughput gains, non conflict-aware existing approaches, ii) the bin packing based heuristic could provide a near optimal solution with very lower computational complexity, and, iii) there were significant gains in terms of resource utilization when employing NOMA.

Concluding, we present our ideas for future work, related to the research subjects investigated in the thesis. Firstly, regarding the RCPD algorithm, we plan to:

- Evaluate it using multi-dimensional time-series to capture more accurately the dynamics of content popularity better (e.g., incorporate additional dimensions with the number of likes, comments, etc.) and in different contexts, such as on the real-time resource utilization of servers;

- Investigate and further extend the algorithm's scalability properties, i.e., estimate the number of videos that can be analyzed in parallel;

- Conduct real large-scale CDN experiments utilizing a distributed architecture with multiple content popularity analyzers, monitoring in real-time clusters of videos, at a minimum overall processing cost.

Lastly, regarding the joint scheduling of URLLC and eMBB traffic in 5G wireless networks, we plan to extend the heuristic solutions in an online manner, i.e., when the mini-slot throughput values of the resource grid becomes available in real-time, sequentially. Furthermore, focusing on NOMA schemes, we will consider: i) applying heuristic solutions to reduce the solutions' computation complexity, and, ii) adding a new constraint for the allowed overlaps of every resource block, considering the symbol interference cancellation (SIC) complexity.

References

[1] Lei You, Qi Liao, Nikolaos Pappas, and Di Yuan. Resource optimization with flexible numerology and frame structure for heterogeneous services. *IEEE Commun. Lett.*, 22(12):2579–2582, 2018.

[2] M. Series. IMT Vision - Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond. Tech. Rep. Recommendation ITU-R M.2083-0, 2015.

[3] Shao-Yu Lien, Shin-Lin Shieh, Yenming Huang, Borching Su, Yung-Lin Hsu, and Hung-Yu Wei. 5G New Radio: Waveform, Frame Structure, Multiple Access, and Initial Access. *IEEE Commun. Mag.*, 55(6):64–71, 2017.

[4] Petar Popovski. Ultra-Reliable Communication in 5G Wireless Systems. In *Proc. Int. Conf. 5G Ubiq. Connect.*, pages 146–151. IEEE, 2014.

[5] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead. 2020. *arXiv:2005.10027*, [Online]. Available: https://arxiv.org/abs/2005.10027.

[6] 3GPP. NR; Physical channels and modulation. Technical Specification (TS) 38.211, 3rd Generation Partnership Project (3GPP), 03 2020. Version 16.1.0.

[7] ICT Facts and Figures - The world in 2015. International Telecommunication Union (ITU), Geneva, Switzerland, 2015.

[8] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile Edge Computing — A Key Technology Towards 5G. ETSI, Sophia Antipolis, France, White paper, 2015.

[9] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge Computing: Vision and Challenges. *IEEE Internet Things J.*, 3(5):637–646, 2016.

[10] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu. Edge Computing in 5G: A review. *IEEE Access*, 7:127276–127289, 2019.

[11] Ejaz Ahmed, Arif Ahmed, Ibrar Yaqoob, Junaid Shuja, Abdullah Gani, Muhammad Imran, and Muhammad Shoaib. Bringing computation closer toward the user network: is Edge Computing the Solution? *IEEE Commun. Mag.*, 55(11):138–144, 2017.

[12] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge Computing: a Survey. *Future Gener. Comput. Syst.*, 97:219–235, 2019.

[13] Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. Network Function Virtualization: Challenges and Opportunities for Innovations. *IEEE Commun. Mag.*, 53(2):90–97, 2015.

[14] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, et al. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar. 2008.

[15] Stuart Clayman, Lefteris Mamatas, and Alex Galis. Experimenting with Control Operations in Software-defined Infrastructures. In *Proc. NetSoft Conf. Workshops (NetSoft)*, pages 390–396. IEEE, 2016.

[16] Gaurav Banga, Peter Druschel, and Jeffrey C Mogul. Resource Containers: a New Facility for Resource Management in Server Systems. In *Proc. 3rd Symp. Operating Syst. Des. Implementation (OSDI)*, volume 99, pages 45–58, 1999.

[17] Anil Madhavapeddy and David J Scott. Unikernels: Rise of the Virtual Library Operating System. *Queue*, 11(11):30–44, 2013.

[18] Mukaddim Pathan, Rajkumar Buyya, and Athena Vakali. Content Delivery Networks: State of the Art, Insights, and Imperatives. In *Content Delivery Networks*, pages 3–32. Springer, 2008.

[19] Polychronis Valsamas, Sotiris Skaperas, and Lefteris Mamatas. Elastic Content Distribution Based on Unikernels and Change-Point Analysis. In *Proc. 24th Eur. Wireless Conf. (EW)*, pages 1–7. VDE, Catania, Italy, 2018.

[20] Polychronis Valsamas, Panagiotis Papadimitriou, Ilias Sakellariou, Sophia Petridou, Lefteris Mamatas, Stuart Clayman, Francesco Tusa, and Alex Galis. Multi-PoP Network Slice Deployment: a Feasibility Study. In *Proc. 8th Int. Conf. Cloud Netw. (CloudNet)*, pages 1–6. IEEE, 2019.

[21] Polychronis Valsamas, Sotiris Skaperas, George Violettas, Tryfon Theodorou, Sophia Petridou, Dimitris Vardalis, Antonios Tsioukas, and Lefteris Mamatas. Multi Access Edge Computing for Efficient Content Distribution and IOT Services. In *Wireless Commun. Netw. Conf. (WCNC)*. IEEE, 2019.

[22] CISCO Visual Networking. CISCO Global Cloud Index: Forecast and Methodology, 2015-2020. San Jose, CA, USA, CISCO, Tech. Rep., 2017.

[23] Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo. Machine Learning Paradigms for Next-generation Wireless Networks. *IEEE Wireless Commun.*, 24(2):98–105, 2016.

[24] Rongpeng Li, Zhifeng Zhao, Xuan Zhou, Guoru Ding, Yan Chen, Zhongyao Wang, and Honggang Zhang. Intelligent 5G: when Cellular Networks meet Artificial Intelligence. *IEEE Wireless Commun.*, 24(5):175–183, 2017.

[25] David M Gutierrez-Estevez, Marco Gramaglia, Antonio De Domenico, Nicola Di Pietro, Sina Khatibi, Kunjan Shah, Dimitris Tsolkas, Paul Arnold, and Pablo Serrano. The Path Towards Resource Elasticity for

5G Network Architecture. In *Proc. Wireless Commun. Netw. Conf. Workshops (WCNCW)*, pages 214–219. IEEE, 2018.

[26] Muhammad Rehan Raza, Carlos Natalino, Peter Öhlen, Lena Wosinska, and Paolo Monti. A Slice Admission Policy Based on Reinforcement Learning for a 5G Flexible RAN. In *Proc. Eur. Conf. Opt. Commun. (ECOC)*, pages 1–3. IEEE, 2018.

[27] Suoheng Li, Jie Xu, Mihaela Van Der Schaar, and Weiping Li. Popularity-Driven Content Caching. In *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pages 1–9. IEEE, San Francisco, CA, USA, Apr. 2016.

[28] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: a Survey. *ACM Comput. Surveys (CSUR)*, 41(3):1–58, Sept. 2009.

[29] Angelos K Marnerides, Alberto Schaeffer-Filho, and Andreas Mauthe. Traffic Anomaly Diagnosis in Internet Backbone Networks: A Survey. *Comput. Netw.*, 73:224–243, 2014.

[30] Ido Nevat, Dinil Mon Divakaran, Sai Ganesh Nagarajan, Pengfei Zhang, Le Su, Li Ling Ko, and Vrizlynn LL Thing. Anomaly Detection and Attribution in Networks With Temporally Correlated Traffic. *IEEE/ACM Trans. Netw. (ToN)*, 26(1):131–144, Feb. 2018.

[31] Seong Soo Kim and AL Narasimha Reddy. Statistical Techniques for Detecting Traffic Anomalies through Packet Header Data. *IEEE/ACM Trans. Netw. (ToN)*, 16(3):562–575, 2008.

[32] Gautam Thatte, Urbashi Mitra, and John Heidemann. Parametric Methods for Anomaly Detection in Aggregate Traffic. *IEEE/ACM Trans. Netw. (ToN)*, 19(2):512–525, 2010.

[33] Daniel Turner, Kirill Levchenko, Alex C Snoeren, and Stefan Savage. California Fault Lines: Understanding the Causes and Impact of Network Failures. In *Proc. ACM SIGCOMM Conf.*, pages 315–326, 2010.

[34] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. Internet of Things Security: a Survey. *J. Network Comput. Appl.*, 88:10–28, 2017.

[35] Jeffrey G Andrews, Stefano Buzzi, Wan Choi, Stephen V Hanly, Angel Lozano, Anthony CK Soong, and Jianzhong Charlie Zhang. What will 5G be? *IEEE J. Sel. Areas Commun.*, 32(6):1065–1082, 2014.

[36] Jianli Pan and James McElhannon. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.*, 5(1):439–449, 2017.

[37] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke. A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements. *IEEE Access*, 5:1872–1899, 2017.

[38] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov. Security in Software Defined Networks: A Survey. *IEEE Commun. Surveys Tuts.*, 17(4):2317–2346, 2015.

[39] Zhaogang Shu, Jiafu Wan, Di Li, Jiaxiang Lin, Athanasios V. Vasilakos, and Muhammad Imran. Security in Software-Defined Networking: Threats and Countermeasures. *Mobile Netw. and Appl.*, 21(5):764–776, Oct 2016.

[40] A. Akhunzada, E. Ahmed, A. Gani, M. K. Khan, M. Imran, and S. Guizani. Securing Software Defined Networks: Taxonomy, Requirements, and Open Issues. *IEEE Commun. Mag.*, 53(4):36–44, April 2015.

[41] Da Yin, Lianming Zhang, and Kun Yang. A DDoS Attack Detection and Mitigation with Software-defined Internet of Things Framework. *IEEE Access*, 6:24694–24705, 2018.

[42] Arjun Anand, Gustavo De Veciana, and Sanjay Shakkottai. Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks. *IEEE/ACM Trans. Netw. (ToN)*, 28(2):477–490, 2020.

[43] Petar Popovski, Kasper Fløe Trillingsgaard, Osvaldo Simeone, and Giuseppe Durisi. 5G wireless network slicing for eMBB, URLLC, and

mMTC: A communication-theoretic view. *IEEE Access*, 6:55765–55779, 2018.

[44] Necos project: Towards lightweight slicing of cloud federated infrastructures. https://intrig.dca.fee.unicamp.br/2017/09/05/necos-2-year-eu-brazil-collaborative-project-starting-in-nov2017/.

[45] Alexandru Tatar, Marcelo Dias De Amorim, Serge Fdida, and Panayotis Antoniadis. A Survey on Predicting the Popularity of Web Content. *J. Internet Services Appl.*, 5(1):8, Dec. 2014.

[46] Gabor Szabo and Bernardo A Huberman. Predicting the Popularity of Online Content. *Commun. ACM*, 53(8):80–88, Aug. 2010.

[47] Tom Goethals, Merlijn Sebrechts, Ankita Atrey, Bruno Volckaert, and Filip De Turck. Unikernels vs Containers: An In-Depth Benchmarking Study in the Context of Microservice Applications. In *Proc. Int. Symp. Cloud Service Comput. (SC2)*. IEEE, Nov. 2018.

[48] Joao Martins, Ahmed Mohamed, Costin Raiciu, and Felipe Huici. Enabling Fast, Dynamic Networking Processing with ClickOS. In *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, pages 67–72, 2013.

[49] Gonca Gürsun, Mark Crovella, and Ibrahim Matta. Describing and Forecasting Video Access Patterns. In *Proc. Int. Conf. Comput. Commun. (INFOCOM)*, pages 16–20. IEEE, Shanghai, China, Apr. 2011.

[50] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can Cascades be Predicted? In *Proc. 23rd Int. Conf. World Wide Web (WWW)*, pages 925–936. ACM, Seoul, Republic of Korea, Apr. 2014.

[51] Stefan Fremdt. Asymptotic Distribution of the Delay Time in Page's Sequential Procedure. *J. Statist. Planning Inference*, 145:74–91, Feb. 2014.

[52] Yannick Hoga. Monitoring multivariate time series. *J. Multivariate Anal.*, 155:105–121, Mar. 2017.

[53] E Brodsky and Boris S Darkhovsky. *Nonparametric methods in change point problems.* Dordrecht, The Netherlands: Kluwer, 2013.

[54] Donald J Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proc. AAAI Workshop Knowl. Disc. Databases (KDD)*, volume 10, pages 359–370, Seattle, USA, Aug. 1994.

[55] Rohit J Kate. Using Dynamic Time Warping Distances as Features for Improved Time Series Classification. *Data Mining Know. Discovery*, 30:283–312, Mar. 2016.

[56] Peter J Brockwell, Richard A Davis, and Matthew V Calder. *Introduction to time series and forecasting.* 2th Ed. New York, NY, USA: Springer, 2002.

[57] Peter Hall and Christopher C Heyde. *Martingale limit theory and its application.* Academic Press, 2014.

[58] Jérôme Dedecker, Paul Doukhan, Gabriel Lang, León R José Rafael, Sana Louhichi, and Clémentine Prieur. Weak dependence: with Examples and Applications. In *Lecture Notes in Statistics*, volume 190, pages 9–20. Springer, New York, 2007.

[59] Wei Biao Wu et al. Strong Invariance Principles for Dependent Random Variables. *Ann. Probability*, 35(6):2294–2320, 2007.

[60] Michèle Basseville and Igor V Nikiforov. *Detection of Abrupt Changes: Theory and Application*, volume 104. Englewood Cliffs, New Jersey, USA: Prentice Hall , 1993.

[61] Alexander Aue and Lajos Horváth. Structural Breaks in Time Series. *J. Time Series Anal.*, 34(1):1–16, Jan. 2013.

[62] Miklós Csörgö and Lajos Horváth. *Limit theorems in change-point analysis*, volume 18. Wiley Series in Probability and Statistics. Chichester: John Wiley & Sons Inc, 1997.

[63] Tobias Berens, Gregor NF Weiß, and Dominik Wied. Testing for Structural Breaks in Correlations: Does it Improve Value-at-Risk Forecasting? *J. Empirical Finance*, 32:135–152, 2015.

[64] Gabriel Tsechpenakis, Dimitris N Metaxas, Carol Neidle, and Olympia Hadjiliadis. Robust Online Change-point Detection in Video Sequences. In *Proc. Comput. Vis. Patt. Recog. Work.*, page 155, New York, NY, USA, June 2006.

[65] Victor Konev and Sergey Vorobeychikov. Quickest Detection of Parameter Changes in Stochastic Regression: Nonparametric CUSUM. *IEEE Trans. Inf. Theory*, 63(9):5588–5602, Sept. 2017.

[66] Alexander G Tartakovsky, Boris L Rozovskii, Rudolf B Blazek, and Hongjoong Kim. A Novel Approach to Detection of Intrusions in Computer Networks via Adaptive Sequential and Batch-sequential Change-point Detection Methods. *IEEE Trans. Signal Process.*, 54(9):3372–3382, Sept. 2006.

[67] Haining Wang, Danlu Zhang, and Kang G Shin. Change-point Monitoring for the Detection of DoS Attacks. *IEEE Trans. Depend. Sec. Comput.*, 1(4):193–208, Oct.-Dec. 2004.

[68] Alexander G Tartakovsky, Aleksey S Polunchenko, and Grigory Sokolov. Efficient Computer Network Anomaly Detection by Change Point Detection Methods. *IEEE J. Sel. Topics Signal Process.*, 7(1):4–11, Feb. 2013.

[69] Augustin Soule, Kavé Salamatian, and Nina Taft. Combining Filtering and Statistical Methods for Anomaly Detection. In *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, pages 1–14. ACM, New York, NY, USA, Oct. 2005.

[70] Venkata Jandhyala, Stergios Fotopoulos, Ian MacNeill, and Pengyu Liu. Inference for Single and Multiple Change-points in Time Series. *J. Time Ser. Anal.*, 34(4):423–446, June 2013.

[71] Sangyeol Lee, Jeongcheol Ha, Okyoung Na, and Seongryong Na. The CUSUM Test for Parameter Change in Time Series Models. *Scand. J. Statist.*, 30(4):781–796, Oct. 2003.

[72] István Berkes, Edit Gombay, Lajos Horváth, and Piotr Kokoszka. Sequential Change-point Detection in GARCH (p, q) Models. *Econometric Theory*, 20(6):1140–1167, Dec. 2004.

[73] Chia-Shang James Chu, Maxwell Stinchcombe, and Halbert White. Monitoring Structural Change. *Econometrica: J. Econometric Soc.*, pages 1045–1065, 1996.

[74] Alexander Aue, Lajos Horváth, Marie Hušková, and Piotr Kokoszka. Change-point Monitoring in Linear Models. *Econometrics J.*, 9(3):373–403, 2006.

[75] Lajos Horváth, Marie Hušková, Piotr Kokoszka, and Josef Steinebach. Monitoring Changes in Linear Models. *J. Statist. Planning Inference*, 126(1):225–251, 2004.

[76] Edit Gombay and Daniel Serban. Monitoring Parameter Change in AR (p) Time Series Models. *J. Multivariate Anal.*, 100(4):715–725, 2009.

[77] Christian Callegari, Angelo Coluccia, Alessandro D'Alconzo, Wendy Ellens, Stefano Giordano, Michel Mandjes, Michele Pagano, Teresa Pepe, Fabio Ricciato, and Piotr Zuraniewski. A Methodological Overview on Anomaly Detection. In *Data Traffic Monitoring and Analysis*, pages 148–183. Springer, Berlin, Germany: Springer-Verlag, 2013.

[78] Samaneh Aminikhanghahi and Diane J Cook. A Survey of Methods for Time Series Change Point Detection. *Knowl. Inf. Syst.*, 51(2):339–367, May 2017.

[79] Dhruv Choudhary, Arun Kejariwal, and Francois Orsini. On the Runtime-Efficacy Trade-off of Anomaly Detection Techniques for Real-Time Streaming Data. Oct. 2017, arXiv:1710.04735. [Online]. Available: https://arxiv.org/abs/1710.04735.

[80] Sotiris Skaperas, Lefteris Mamatas, and Arsenia Chorti. Early Video Content Popularity Detection with Change Point Analysis. In *Proc. Int. Global Commun. Conf. (GLOBECOM)*, pages 1–7. IEEE, Abu Dhabi, UAE, Dec. 2018.

[81] Sotiris Skaperas, Lefteris Mamatas, and Arsenia Chorti. Real-Time Algorithms for the Detection of Changes in the Variance of Video Content Popularity. *IEEE Access*, 8:30445–30457, 2020.

[82] S. Skaperas, L. Mamatas, and A. Chorti. Real-Time Video Content Popularity Detection Based on Mean Change Point Analysis. *IEEE Access*, 7:142246–142260, 2019.

[83] Youmna Borghol, Siddharth Mitra, Sebastien Ardon, Niklas Carlsson, Derek Eager, and Anirban Mahanti. Characterizing and Modelling Popularity of User-Generated Videos. *Perform. Eval.*, 68(11):1037–1055, Nov. 2011.

[84] Jie Xu, Mihaela Van Der Schaar, Jiangchuan Liu, and Haitao Li. Forecasting Popularity of Videos Using Social Media. *IEEE J. Sel. Topics Signal Process.*, 9(2):330–343, Mar. 2015.

[85] Henrique Pinto, Jussara M Almeida, and Marcos A Gonçalves. Using Early View Patterns to Predict the Popularity of YouTube Videos. In *Proc. 6th ACM Int. Conf. Web Search and Data Mining (WSDM)*, pages 365–374. ACM, Rome, Italy, Feb. 2013.

[86] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. YouTube Traffic Characterization: a View from the Edge. In *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, pages 15–28. ACM, San Diego, California, USA, 2007.

[87] Savera Tanwir and Harry Perros. A Survey of VBR Video Traffic Models. *IEEE Commun. Surveys Tuts.*, 15(4):1778–1802, Jan. 2013.

[88] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. News Comments: Exploring, Modeling, and Online Prediction. In *Proc. Eur.*

Conf. Inform. Retrieval, pages 191–203. Springer, Milton Keynes, UK, Mar. 2010.

[89] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. Analyzing the Video Popularity Characteristics of Large-scale User Generated Content Systems. *IEEE/ACM Trans. Netw. (ToN)*, 17(5):1357–1370, Oct. 2009.

[90] Tomasz Trzciński and Przemysław Rokita. Predicting Popularity of Online Videos Using Support Vector Regression. *IEEE Trans. Multimedia*, 19(11):2561–2570, Nov. 2017.

[91] Sam Romano and Hala ElAarag. A Neural Network Proxy Cache Replacement Strategy and its Implementation in the Squid Proxy Server. *Neural Comput. & Appl.*, 20(1):59–78, Sept. 2011.

[92] Waleed Ali and Siti Mariyam Shamsuddin. Intelligent Client-Side Web Caching Scheme Based on Least Recently Used Algorithm and Neuro-Fuzzy System. In *Proc. Int. Symp. Neural Netw. (ISNN)*, pages 70–79. Springer, Wuhan, China, May 2009.

[93] W-X. Liu, J. Zhang, Z-W. Liang, L-X. Peng, and J. Cai. Content Popularity Prediction and Caching for ICN: A Deep Learning Approach With SDN. *IEEE Access*, 6:5075–5089, 2018.

[94] Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. In *Proc. 17th Int. Conf. World Wide Web (WWW)*, pages 247–256. ACM, Beijing, China, Apr. 2008.

[95] Jiongjiong Song, Min Sheng, Tony Q. S. Quek, Chao Xu, and Xijun Wang. Learning-Based Content Caching and Sharing for Wireless Networks. *IEEE Trans. Commun.*, 65(10):4309–4324, Oct. 2019.

[96] Fan Jiang, Zeng Yuan, Changyin Sun, and Junxuan Wang. Deep Q-Learning-Based Content Caching with Update Strategy for Fog Radio Access Networks. *IEEE Access*, 7:97505–97514, 2019.

[97] Xiaobo Zhou and Cheng-Zhong Xu. Optimal Video Replication and Placement on a Cluster of Video-on-Demand Servers. In *Proc. Int. Conf. Parallel Process. (ICPP)*, pages 547–555. IEEE, Vancouver, Canada, Aug. 2002.

[98] Wenting Tang, Yun Fu, Ludmila Cherkasova, and Amin Vahdat. Modeling and Generating Realistic Streaming Media Server Workloads. *Comput. Netw.*, 51(1):336–356, Jan. 2007.

[99] Rabindra N Bhattacharya, Vijay K Gupta, and Ed Waymire. The Hurst Effect Under Trends. *J. Appl. Probability*, pages 649–662, 1983.

[100] Thomas Karagiannis, Michalis Faloutsos, and Rudolf H Riedi. Long-Range Dependence: Now you See it, Now you Don't! In *Proc. Int. Global Commun. Conf. (GLOBECOM)*, volume 3, pages 2165–2169. IEEE, 2002.

[101] Donald WK Andrews. Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica: J. Econometric Soc.*, 59:817–858, May 1991.

[102] Dominik Wied. A Nonparametric Test for a Constant Correlation Matrix. *Econometric Rev.*, 36(10):1157–1172, Apr. 2017.

[103] Marc Lavielle and Gilles Teyssiere. Adaptive Detection of Multiple Change-points in Asset Price Volatility. In *Long Memory in Economics*, pages 129–156. Springer, G. Teyssiere and A. Kirkman, Eds. Berlin, Germany: Springer–Verlag, 2007.

[104] Daniele Angelosante and Georgios B Giannakis. Sparse Graphical Modeling of Piecewise-stationary Time Series. In *Proc. Int. Conf. Acoust., Speech and Signal Process (ICASSP)*, pages 1960–1963. IEEE, Prague, Czech Republic, May 2011.

[105] Carla Inclan and George C Tiao. Use of Cumulative Sums of Squares for Retrospective Detection of Changes of Variance. *J. Amer. Statist. Assoc.*, 89(427):913–923, Sept. 1994.

[106] Huang Kai, Qi Zhengwei, and Liu Bo. Network Anomaly Detection Based on Statistical Approach and Time Series Analysis. In *Proc. Int. Conf.*

Advanced Inform. Netw. Appl. (WAINA) Workshops, pages 205–211. IEEE, Bradford, UK, May 2009.

[107] Nesrine Ben Hassine, Ruben Milocco, and Pascale Minet. ARMA Based Popularity Prediction for Caching in Content Delivery Networks. In *Proc. Wireless Days*, pages 113–120. IEEE, Porto, Portugal, Mar. 2017.

[108] Dominik Wied and Pedro Galeano. Monitoring Correlation Change in a Sequence of Random Variables. *J Statist. Planning Inference*, 143(1):186–196, Jan. 2013.

[109] Mattia Zeni, Daniele Miorandi, and Francesco De Pellegrini. YOUStatAnalyzer: a Tool for Analysing the Dynamics of YouTube Content Popularity. In *Proc. 7th Int. Conf. Perform. Eval. Methodol. Tools*, pages 286–289. ICST, Torino, Italy, Dec. 2013.

[110] Richard G Clegg. A Practical Guide to Measuring the Hurst Parameter. *Int. J. Simul. Syst. Sci. Technol.*, 7(2):3–14, Nov. 2006.

[111] Jean-Marc Bardet, Gabriel Lang, Georges Oppenheim, Anne Philippe, Stilian Stoev, and Murad S Taqqu. Semi-parametric Estimation of the Long-range Dependence Parameter: a Survey. *Theory and Applications of Long-range Dependence*, pages 557–577, 2003.

[112] Fabrice Guillemin, Bruno Kauffmann, Stephanie Moteau, and Alain Simonian. Experimental Analysis of Caching Efficiency for YouTube Traffic in an ISP Network. In *Proc. 25th Int. Teletraffic Congr. (ITC)*, pages 1–9. IEEE, Shanghai, China, Sept. 2013.

[113] NGMN Alliance. 5G White Paper. *Next Generation Mobile Networks, White Paper*, 2015.

[114] A. Wang, Z. Zha, Y. Guo, and S. Chen. Software-Defined Networking Enhanced Edge Computing: A Network-Centric Survey. *Proc. IEEE*, 107:1500–1519, Aug. 2019.

[115] Kideok Cho, Munyoung Lee, Kunwoo Park, Ted Taekyoung Kwon, Yanghee Choi, and Sangheon Pack. WAVE: Popularity-Based and Collaborative in-Network Caching for Content-Oriented Networks. In *Proc.*

Int. Conf. Comput. Commun. (INFOCOM) Workshops, pages 316–321. IEEE, 2012.

[116] Xiaofei Wang, Min Chen, Tarik Taleb, Adlen Ksentini, and Victor CM Leung. Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems. *IEEE Commun. Mag.*, 52(2):131–139, 2014.

[117] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE*, 103(1):14–76, Jan 2015.

[118] R. C. A. Alves, D. A. G. Oliveira, G. A. Nunez Segura, and C. B. Margi. The Cost of Software-Defining Things: A Scalability Study of Software-Defined Sensor Networks. *IEEE Access*, 7:115093–115108, Aug 2019.

[119] Di Niu, Zimu Liu, Baochun Li, and Shuqiao Zhao. Demand Forecast and Performance Prediction in Peer-Assisted On-Demand Streaming Systems. In *Proc. Int. Conf. Comput. Commun. (INFOCOM)*, pages 421–425. IEEE, Shanghai, China, Apr. 2011.

[120] Christos Katris and Sophia Daskalaki. Generation of Synthetic Video Traffic Using Time Series. *Simul. Model. Pract. Theory*, 75:127–145, June 2017.

[121] Bo Zhou, Dan He, Zhili Sun, and Wee Hock Ng. Network Traffic Modeling and Prediction with ARIMA/GARCH. In *Proc. HET-NETs Conf.*, pages 1–10, IIkley, UK, July 2005.

[122] Christos Katris and Sophia Daskalaki. Dynamic Bandwidth Allocation for Video Traffic Using FARIMA-Based Forecasting Models. *J. Netw. Syst. Manag.*, 27(1):39–65, Jan. 2019.

[123] Di Niu, Hong Xu, and Baochun Li. Resource Auto-Scaling and Sparse Content Replication for Video Storage Systems. *ACM Trans. Modeling Perform. Eval. Comput. Syst. (TOMPECS)*, 2(4):19, Dec. 2017.

[124] Fabio Lopez-Pires and Benjamin Baran. Virtual Machine Placement Literature Review, 2015. *arXiv:1506.01509*, [Online]. Available: https://arxiv.org/abs/1506.01509.

[125] Giuseppe Siracusano, Roberto Bifulco, Martino Trevisan, Tobias Jacobs, Simon Kuenzer, Stefano Salsano, Nicola Blefari-Melazzi, and Felipe Huici. Re-Designing Dynamic Content Delivery in the Light of a Virtualized Infrastructure. *IEEE J. Sel. Areas Commun.*, 35(11):2574–2585, 2017.

[126] Yanxiang Jiang, Miaoli Ma, Mehdi Bennis, Fuchun Zheng, and Xiaohu You. A Novel Caching Policy with Content Popularity Prediction and User Preference Learning in Fog-RAN. In *Proc. Int. Global Commun. Conf. (GLOBECOM) Workshops*, pages 1–6. IEEE, 2017.

[127] S. S. Bhunia and M. Gurusamy. Dynamic attack detection and mitigation in IoT using SDN. In *27th Int. Telecommun. Netw. and Appl. Conf. (ITNAC)*, pages 1–6, Nov 2017.

[128] Rui Wang, Zhiyong Zhang, Zhiwei Zhang, and Zhiping Jia. ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs. *Comput. Netw.*, 139:119 – 135, 2018.

[129] G. A. Nunez Segura, Sotiris Skaperas, Arsenia Chorti, Lefteris Mamatas, and C. B. Margi. Denial of Service Attacks Detection in Software-Defined Wireless Sensor Networks. In *Proc. IEEE Int. Conf. Commun. (ICC) Worskhop on SDN Security, Dublin UK*, 2020.

[130] Alexander Aue, Siegfried Hörmann, Lajos Horváth, and Matthew Reimherr. Break Detection in the Covariance Structure of Multivariate Time Series Models. *Ann. Statist.*, 37(6B):4046–4087, Dec. 2009.

[131] K. Pape, D. Wied, and P. Galeano. Monitoring Multivariate Variance Changes. *J. Empirical Finance*, 75A:54–68, Dec. 2016.

[132] Alexander Aue, Christopher Dienes, Stefan Fremdt, Josef Steinebach, et al. Reaction Times of Monitoring Schemes for ARMA Time Series. *Bernoulli*, 21(2):1238–1259, May 2015.

[133] Youngmi Lee Na, Okyoung and Sangyeol Lee. Monitoring Parameter Change in Time Series Models. *Statist. Methods & Appl.*, 20(2):171–199, June 2011.

[134] James Davidson. *Stochastic limit theory: An introduction for econometricians.* OUP Oxford, New York, USA: Oxford University Press (OUP), 1994.

[135] D. Wied, M. Arnold, N. Bissantz, et al. A New Fluctuation Test for Constant Variances with Applications to Finance. *Metrica*, 75(8):1111–1127, Nov. 2012.

[136] Christian Francq and Jean-Michel Zakoian. Maximum Likelihood Estimation of Pure GARCH and ARMA-GARCH Processes. *Bernoulli*, 10(4):605–637, Aug. 2004.

[137] Wai Leong Ng Leung, Sze Him and Chun Yip Yau. Sequential Change-point Detection in Time Series Models Based on pairwise Likelihood. *Statistica Sinica*, 27(2):575–605, Apr. 2017.

[138] Node-RED: http://nodered.org/.

[139] Ansible:https://www.ansible.com.

[140] Httperf HTTP load generator: https://github.com/httperf/httperf.

[141] Collectd The System Statistics Collection Daemon: https://collectd.org/download.

[142] InfluxData (InfluxDB) — Time Series Database Monitoring & Analytics: https://www.influxdata.com/developers/.

[143] Grafana - The Open Platform for Analytics and Monitoring: https://grafana.com/.

[144] Gustavo A Nunez Segura, Cintia Borges Margi, and Arsenia Chorti. Understanding the performance of software defined wireless sensor networks under denial of service attack. *Open J. Internet Things (OJIOT)*, 5(1):58–68, 2019.

[145] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-Level Sensor Network Simulation with COOJA. In *Proc. Conf. Local Comput. Netw. (LCN)*, pages 641–648, Nov 2006.

[146] Yalcin Sadi, Serhat Erkucuk, and Erdal Panayirci. Flexible physical layer based resource allocation for machine type communications towards 6G. In *Proc. IEEE 2nd 6G Wireless Summit (6G SUMMIT)*, pages 1–5, Virtual, Mar. 2020.

[147] Anique Akhtar and Hüseyin Arslan. Downlink resource allocation and packet scheduling in multi-numerology wireless systems. In *Proc. IEEE Wireless Commun. Netw. Conf. Workshop (WCNCW)*, pages 362–367, Barcelona, Spain, Apr. 2018.

[148] Ljiljana Marijanovic, Stefan Schwarz, and Markus Rupp. Multi-user resource allocation for low latency communications based on mixed numerology. In *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, pages 1–7, Honolulu, Hawaii, USA, Sept. 2019.

[149] Changyang She, Chenyang Yang, and Tony QS Quek. Radio resource management for ultra-reliable and low-latency communications. *IEEE Commun. Mag.*, 55(6):72–78, 2017.

[150] Omid Semiari, Walid Saad, Mehdi Bennis, and Merouane Debbah. Integrated millimeter wave and sub-6 GHz wireless networks: A roadmap for joint mobile broadband and ultra-reliable low-latency communications. *IEEE Wireless Commun.*, 26(2):109–115, 2019.

[151] He Chen, Rana Abbas, Peng Cheng, Mahyar Shirvanimoghaddam, Wibowo Hardjawana, Wei Bao, Yonghui Li, and Branka Vucetic. Ultra-reliable low latency cellular networks: Use cases, challenges and approaches. *IEEE Commun. Mag.*, 56(12):119–125, 2018.

[152] Joachim Sachs, Gustav Wikstrom, Torsten Dudda, Robert Baldemair, and Kittipong Kittichokechai. 5G radio network design for ultra-reliable low-latency communication. *IEEE Netw.*, 32(2):24–31, 2018.

[153] Kangjie Zhang, Xiaodong Xu, Jingxuan Zhang, Bufang Zhang, Xiaofeng Tao, and Yuantao Zhang. Dynamic Multiconnectivity Based Joint Scheduling of eMBB and uRLLC in 5G Networks. *IEEE Syst. J.*, early access, Apr. 2020.

[154] Guillermo Pocovi, Klaus I Pedersen, and Preben Mogensen. Joint link adaptation and scheduling for 5g ultra-reliable low-latency communications. *IEEE Access*, 6:28912–28922, 2018.

[155] Praveenkumar Korrai, Eva Lagunas, Shree Krishna Sharma, Symeon Chatzinotas, Ashok Bandi, and Björn Ottersten. A RAN resource slicing mechanism for multiplexing of eMBB and URLLC Services in OFDMA Based 5G wireless networks. *IEEE Access*, 8:45674–45688, 2020.

[156] Praveen Kumar Korrai, Eva Lagunas, Shree Krishna Sharma, Symeon Chatzinotas, and Björn Ottersten. Slicing based resource allocation for multiplexing of eMBB and URLLC services in 5G wireless networks. In *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw. (CAMAD)*, pages 1–5, Limassol, Cyprus, Sep. 2019.

[157] Miroslav Mitev, Arsenia Chorti, Martin Reed, and Leila Musavian. Authenticated secret key generation in delay-constrained wireless systems. *EURASIP J. Wireless Commun. Netw.*, 2020(1):1–29, 2020.

[158] Lingyang Song, Yonghui Li, Zhiguo Ding, and H Vincent Poor. Resource management in non-orthogonal multiple access networks for 5G and beyond. *IEEE Netw.*, 31(4):8–14, 2017.

[159] SM Riazul Islam, Nurilla Avazov, Octavia A Dobre, and Kyung-Sup Kwak. Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges. *IEEE Commun. Surveys Tuts.*, 19(2):721–742, 2016.

[160] Ayman T Abusabah and Huseyin Arslan. NOMA for multinumerology OFDM systems. *Wireless Commun. Mobile Comput.*, 2018:1–9, 2018.

[161] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex Optimization*. Cambridge, U.K: Cambridge Univ. Press, 2004.

[162] George B Dantzig. Discrete-Variable Extremum Problems. *Oper. Res.*, 5(2):266–288, 1957.

[163] Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ronald L. Rivest, and Robert Endre Tarjan. Time Bounds for Selection. *J. Comput. Syst. Sci.*, 7(4):448–461, 1973.

[164] David P Williamson and David B Shmoys. *The Design of Approximation Algorithms*. Cambridge, U.K: Cambridge Univ. Press, 2011.

[165] Silvano Martello. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley-Interscience Ser. Discrete Math. Optim., John Wiley & Sons Ltd., 1990.

[166] Michael R Gary and David S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. WH Freeman & Co., New York, 1979.

[167] Edward G Coffman, János Csirik, Gábor Galambos, Silvano Martello, and Daniele Vigo. *Bin Packing Approximation Algorithms: Survey and Classification*. 2nd Ed. Handbook of combinatorial Optim., Springer New York, 2013.

[168] Andrew Chi-Chih Yao. New Algorithms for Bin Packing. *Journal of the ACM (JACM)*, 27(2):207–227, 1980.

[169] Jianhua Tang, Byonghyo Shim, and Tony QS Quek. Service Multiplexing and Revenue Maximization in Sliced C-RAN Incorporated with URLLC and Multicast eMBB. *IEEE J. Sel. Areas Commun.*, 37(4):881–895, 2019.

[170] Ali A Esswie, Klaus I Pedersen, and Preben E Mogensen. Preemption-aware rank offloading scheduling for latency critical communications in 5g networks. In *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, pages 1–6. IEEE, Kuala Lumpur, Malaysia, May 2019.

[171] Ali A Esswie and Klaus I Pedersen. Opportunistic spatial preemptive scheduling for URLLC and eMBB coexistence in multi-user 5G networks. *IEEE Access*, 6:38451–38463, 2018.

[172] Guillermo Pocovi, Hamidreza Shariatmadari, Gilberto Berardinelli, Klaus Pedersen, Jens Steiner, and Zexian Li. Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements. *IEEE Netw.*, 32(2):8–15, 2018.

[173] Annapurna Pradhan and Susmita Das. Joint preference metric for efficient resource allocation in co-existence of eMBB and URLLC. In *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, pages 897–899, Bengaluru, India, Jan. 2020.

[174] Klaus I Pedersen, Guillermo Pocovi, Jens Steiner, and Saeed R Khosravirad. Punctured scheduling for critical low latency data on a shared channel with mobile broadband. In *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, pages 1–6, Toronto, Canada, Sept. 2017.

[175] Madyan Alsenwi, Nguyen H Tran, Mehdi Bennis, Anupam Kumar Bairagi, and Choong Seon Hong. eMBB-URLLC resource slicing: A risk-sensitive approach. *IEEE Commun. Lett.*, 23(4):740–743, 2019.

[176] Jing Li and Xing Zhang. Deep Reinforcement Learning Based Joint Scheduling of eMBB and URLLC in 5G Networks. *IEEE Wireless Commun. Lett.*, 2020.

[177] Madyan Alsenwi, Nguyen H Tran, Mehdi Bennis, Shashi Raj Pandey, Anupam Kumar Bairagi, and Choong Seon Hong. Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach. 2020, *arXiv:2003.07651*. [Online]. Available: http://arxiv.org/ abs/2003.07651.

[178] Chao Tang, Xin Chen, Ying Chen, and Zhuo Li. Dynamic Resource Optimization Based on Flexible Numerology and Markov Decision Process for Heterogeneous Services. In *Proc. 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, pages 610–617. IEEE, 2019.

[179] Xiaoying Zhang, Lei Zhang, Pei Xiao, Dongtang Ma, Jibo Wei, and Yu Xin. Mixed Numerologies Interference Analysis and Inter-Numerology Interference Cancellation for Windowed OFDM Systems. *IEEE Trans. Veh. Technol.*, 67(8):7047–7061, 2018.

[180] Gordon J Sutton et al. Enabling technologies for ultra-reliable and low latency communications: From PHY and MAC layer perspectives. *IEEE Commun. Surveys Tuts.*, 21(3):2488–2524, 2019.

[181] Klaus Pedersen, Guillermo Pocovi, Jens Steiner, and Andreas Maeder. Agile 5G Scheduler for Improved E2E Performance and Flexibility for Different Network Implementations. *IEEE Commun. Mag.*, 56(3):210–217, 2018.

[182] Sundaram Vanka, Sunil Srinivasa, Zhenhua Gong, Peter Vizi, Kostas Stamatiou, and Martin Haenggi. Superposition Coding Strategies: Design and Experimental Evaluation. *IEEE Trans. Wireless Commun.*, 11(7):2628–2639, 2012.

[183] Nikolaos I Miridakis and Dimitrios D Vergados. A Survey on the Successive Interference Cancellation Performance for Single-Antenna and Multiple-Antenna OFDM Systems. *IEEE Commun. Surveys & Tuts.*, 15(1):312–335, 2012.

[184] Xiaofang Sun, Shihao Yan, Nan Yang, Zhiguo Ding, Chao Shen, and Zhangdui Zhong. Downlink NOMA Transmission for Low-Latency Short-Packet Communications. In *Proc. Int. Conf. Commun. (ICC) Workshops*, pages 1–6. IEEE, 2018.

[185] Zhiguo Ding, Jie Xu, Octavia A Dobre, and H Vincent Poor. Joint Power and Time Allocation for NOMA–MEC Offloading. *IEEE Trans. Veh. Technol.*, 68(6):6207–6211, 2019.

[186] Rahif Kassab, Osvaldo Simeone, Petar Popovski, and Toufiqul Islam. Non-Orthogonal Multiplexing of Ultra-Reliable and Broadband Services in Fog-Radio Architectures. *IEEE Access*, 7:13035–13049, 2019.

[187] Rahif Kassab, Osvaldo Simeone, and Petar Popovski. Coexistence of URLLC and eMBB Services in the C-RAN Uplink: An Information-Theoretic Study. In *Proc. Int. Global Commun. Conf. (GLOBECOM)*, pages 1–6. IEEE, 2018.

[188] Lei You, Qi Liao, Nikolaos Pappas, and Di Yuan. 2D Resource Allocation. *IEEE Dataport*, 2018, doi:10.21227/ch8e-x385.

[189] Daniel Pérez Palomar and Mung Chiang. A Tutorial on Decomposition Methods for Network Utility Maximization. *IEEE J. Sel. Areas Commun.*, 24(8):1439–1451, 2006.

[190] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. 5th Ed. Springer Publishing Company, Incorporated, 2006.

[191] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. 2nd Ed. Wiley-Interscience, Wiley&Sons Inc, 2006.

[192] Honglei Miao and Michael Faerber. Physical downlink control channel for 5G new radio. In *Proc. IEEE Eur. Conf. Netw. Commun. (EuCNC), Oulu, Finland*, pages 1–5, June 2017.

[193] Ahmet Yazar and Hüseyin Arslan. A flexibility metric and optimization methods for mixed numerologies in 5G and beyond. *IEEE Access*, 6:3755–3764, 2018.

[194] Release 14, technical specification (ts) 36.101 v 14.3.0, 3rd generation partnership project (3gpp), *3GPP Evolved universal terrestrial radio access (E-UTRA); User equipment (UE) radio transmission and reception*, 2017.

[195] Mickael Batariere, Kevin Baum, and Thomas P Krauss. Cyclic prefix length analysis for 4G OFDM systems. In *Proc. IEEE 60th Veh. Technol. Conf. (VTC-Fall)*, pages 543–547, Los Angeles, CA, USA, Sep. 2004.