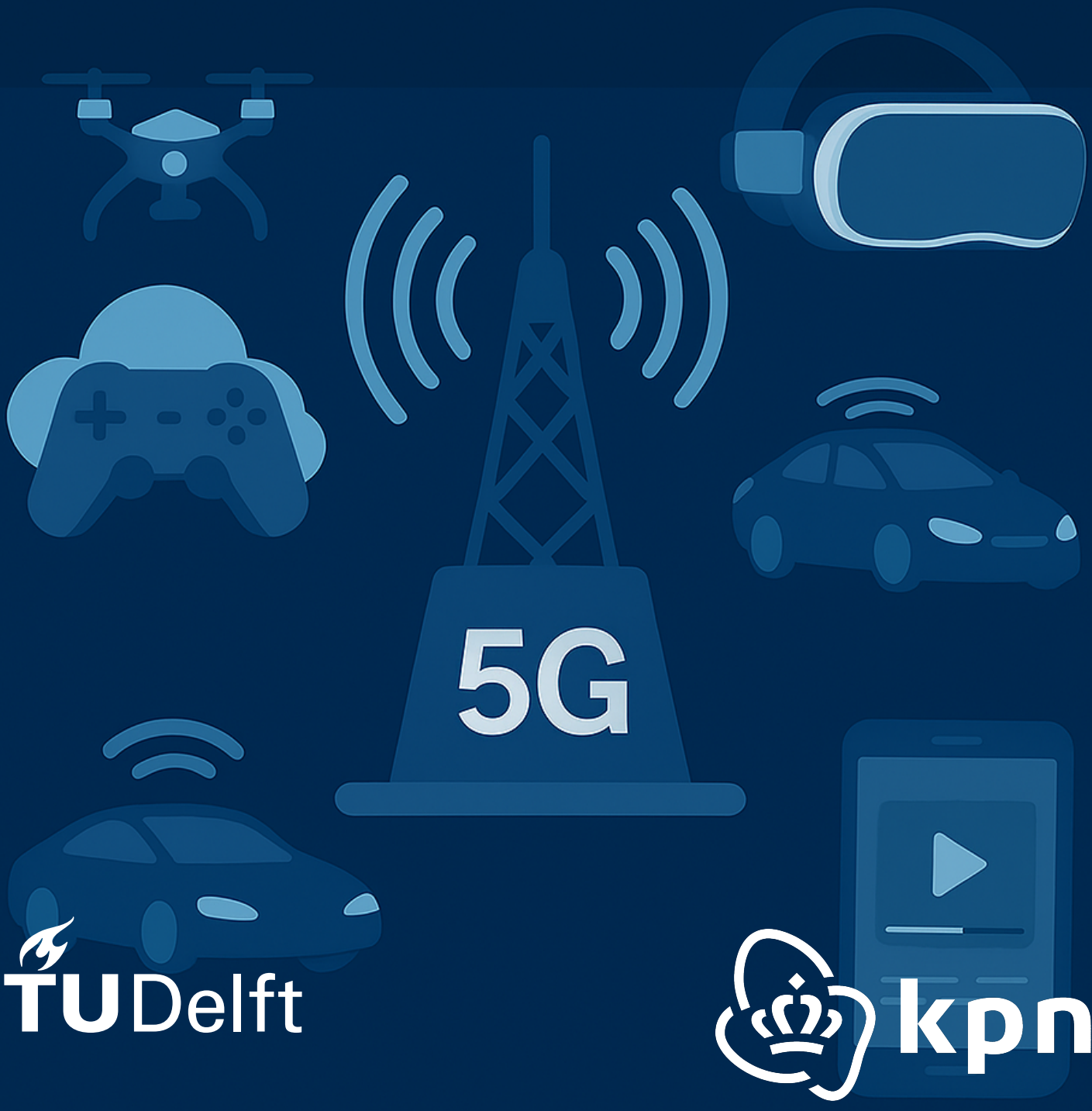
The Impact of Rate Adaptation Mechanisms in Mobile Networks for QoS Enhancement

A Comparative Analysis of L4S and ANBR

Anieze Ikedionwu



TU Delft

kpn

The Impact of Rate Adaptation Mechanisms in Mobile Networks for QoS Enhancement

A Comparative Analysis of L4S and ANBR

MSc Thesis report

by

Anieze Ikedionwu

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on August 29, 2025 at 10:00

Thesis committee:

Chair:	Dr. ir. Eric Smeitink
Daily supervisors:	Dr. ing. Edgar van Boven
	Ir. Rogier Noldus
Company supervisor:	Ing. Paul Schilperoort
External examiner:	Dr. Qing Wang
Place:	Faculty of Electrical Engineering, Delft
Project Duration:	November, 2024 - August, 2025
Student number:	5928273

TU Delft

kpn

Acknowledgment

I would like to use this section to express my heartfelt gratitude to everyone who contributed to this thesis. First and foremost, I thank my Chair, Eric Smeitink, for his support, advice, and valuable feedback throughout the process.

I am also grateful to my daily supervisors, Edgar van Boven and Rogier Noldus. Edgar not only connected me with KPN but also provided feedback and support throughout my thesis. Rogier, an engineer at Ericsson, also supported me and enabled the use of the Ericsson gNodeB at KPN for testing. My appreciation also goes to Qing Wang for kindly agreeing to be part of my thesis committee.

A special thank you to my company supervisor, Paul Schilperoort, for welcoming me to KPN, providing this exciting topic, and generously sharing his expertise. I truly appreciate the many learning opportunities and your continuous support.

To my manager at KPN, Anne van Otterlo, thank you for your warm welcome, valuable insights, and constructive feedback. I also extend my appreciation to the End-to-End team for welcoming me.

Many people played a key role in setting up the test network at KPN's test center and supported me during the testing phase. Anouska Bongers, Jan Rompelberg, Jochem Westerduin, and Robert Nicoderm helped with integrating the radio into the test network. Special thanks to Anouska for her kindness and for helping to organize the process. Vincent Weijers supported me with troubleshooting the smartphones used during testing when they were unable to connect to the network. Billy Tiang helped me to provision the SIM cards I used for for the tests. Inti Risseeuw provided me with the server used for the tests. Lichang Zhang guided me in configuring the APN in the core network, and Muthanna Oqiali helped to enable the connection of the Ericsson gNodeB to the secure gateway. I am truly grateful to each of you.

I would also like to acknowledge Eric Verdaasdonk, the Ericsson engineer who helped install the base station at the KPN test center. Thank you so much for your support.

To those who patiently answered my many questions, Rob Hendriks, Faizel Salimi, Rob Paats, Martin de Vreugd and Eric Oosterndorp, thank you. I truly appreciate your time and help.

Everyone took time out of their busy schedules to assist me, and for that, I am sincerely grateful. To all other colleagues at KPN who welcomed me and contributed in various ways, even if not mentioned by name; thank you! Your support has meant a lot to me.

Anieze Ikedionwu
Delft, August 2025

Abstract

The increasing demand for real-time applications such as cloud gaming, augmented/virtual reality (AR/VR), remote control, and industrial automation, has placed stringent requirements on mobile networks to deliver ultra-low latency and high reliability. As 5G networks evolve, ensuring consistently low delays, even during congestion periods, is critical for these real-time applications.

This thesis investigates two network-assisted rate adaptation mechanisms: Low Latency Low Loss Scalable Throughput (L4S) and Access Network Bitrate Recommendation (ANBR). Both mechanisms aim to reduce latency and packet loss while maximizing throughput during periods of congestion. L4S, standardized by 3GPP and IETF, uses Explicit Congestion Notification (ECN) marking in the IP header of the packets, where the base station marks packets to signal early signs of congestion. This allows the sender to react promptly and adjust its transmission rate using a scalable congestion control algorithm. ANBR, also standardized by 3GPP, takes a different approach by providing rate recommendations from the base station to the user equipment (UE) using MAC layer messages.

While both technologies share similar goals, L4S has seen significant industry interest in recent times, whereas ANBR remains relatively underexplored. Despite their potential and similarities, the coexistence of these two technologies and suitability for different scenarios have not been thoroughly investigated.

This research done in collaboration with KPN, addresses this gap by evaluating the comparative performance, suitability, and coexistence of L4S and ANBR for different network scenarios. The research combines theoretical and practical analysis. The units of research include literature and standards reviews, simulations using ns-3, and practical experiments conducted at KPN's test lab. Latency, packet loss, and throughput are analyzed for each experiment.

The findings provide insights into the advantages and disadvantages of L4S and ANBR, and highlight the applications for which they are most suitable. Based on the findings, recommendations are proposed to guide the effective adoption and integration of L4S and/or ANBR in KPN. A key finding from the research is that L4S is better suited for applications requiring ultra-low latency, while ANBR is more appropriate for applications with higher throughput sensitivity. With L4S, telecom operators can have better control over latency and define queueing thresholds at which rate adaptation should begin for the applications, enabling them to better ensure that the Quality of Service (QoS) requirements of each application are met. In contrast, ANBR does not directly target queueing delay; instead, it uses a window mechanism to send rate recommendations to the UE, which limits its ability to control latency.

Contents

Acronyms

3GPP	Third Generation Partnership Project
5G	fifth generation
5G MS	5G Media Streaming
5G-MAG	5G Media Action Group
5GC	5G Core
5QI	5G QoS Identifier
ABR	Adaptive Bitrate
ACK	Acknowledgment
AMF	Access and Mobility Management Function
ANBR	Access Network Bitrate Recommendation
ANBRQ	Access Network Bitrate Recommendation Query
AQM	Active Queue Management
AR	Augmented Reality
AR/VR	Augmented Reality/Virtual Reality
ARP	Allocation and Retention Priority
AVC	Advanced Video Coding
BBR	Bottleneck Bandwidth and Round-trip propagation time
CE	Congestion Experienced
CUPS	Control and User Plane Separation
DASH	Dynamic Adaptive Streaming over HTTP
DCTCP	Data Center TCP
DL	Downlink
DN	Data Networks
DRB	Data Radio Bearer
ECE	ECN Echo
ECN	Explicit Congestion Notification
ECT	ECN Capable Transport
eMBB	enhanced Mobile Broadband

GBR	Guaranteed Bitrate
GCC	Google Congestion Control
HEVC	High Efficiency Video Coding
IETF	Internet Engineering Task Force
L4S	Low Latency Low Loss Scalable Throughput
MAC	Media Access Control
MAC CE	Media Access Control Control Element
mMTC	massive Machine-Type Communications
NEF	Network Exposure Function
NFs	Network Functions
NG-RAN	Next Generation Radio Access Network
NR	New Radio
PCC	Policy and Charging Control
PCF	Policy Control Function
PDCP	Packet Data Convergence Protocol
PDR	Packet Detection Rule
PDU	Protocol Data Unit
PHY	Physical layer
QER	QoS Enforcement Rule
QFI	Qos Flow Identifier
QoE	Quality of Experience
QOS	Quality of Service
QUIC	Quick UDP Internet Connections
RAN	Radio Access Network
RED	Random Early Detection
RFC	Request for Comments
RLC	Radio Link Control
ROHC	Robust Header Compression
RTCP	Real-time Transport Control Protocol
RTP	Real-Time Protocol
RTT	Round trip time

SBA	Service Based Architecture
SCReAM	Self-Clocked Rate Adaptation for Multimedia
SDAP	Service Data Adaptation Protocol
SMF	Session Management Function
TCP	Transmission Control Protocol
ToS	Type of Service
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communications
ViLTE	Video over LTE
ViNR	Video over NR
VoIP	Voice over IP
VoLTE	Voice over LTE
VoNR	Voice over NR
VR	Virtual Reality
vRTT	Virtual Round Trip Time
XR	Extended Reality

List of Figures

List of Tables

1

Introduction

1.1 Background

Over the past decade, mobile communication networks have undergone transformative changes driven by the rise of latency-sensitive and high-bandwidth applications. The next generation of services, ranging from cloud gaming and immersive Augmented Reality/Virtual Reality (AR/VR) experiences to real-time industrial control systems and telemedicine, places unprecedented demands on mobile network infrastructure. While some primarily demand stringent latency guarantees, others require both high throughput and low latency to ensure acceptable performance.

The evolution of 5G networks has been largely motivated by these emerging requirements. Unlike its predecessors, 5G aims to support a wide range of service categories, including enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC) as shown in Figure 1.1 [1]. Among these, URLLC targets end-to-end latencies in the order of milliseconds or sub-milliseconds, which is critical for applications where any delay can compromise safety, functionality, or user experience.

As the digital ecosystem advances, the scope and scale of real-time interactions are expanding. Technologies such as digital twins for industrial systems, AR-enabled smart glasses, autonomous driving, and remote robotic surgery are transitioning from concept to deployment. These applications rely heavily on low-latency data exchange, where even minimal delays on the order of tens of milliseconds can lead to failures, safety risks, or significant degradation in Quality of Experience (QoE).

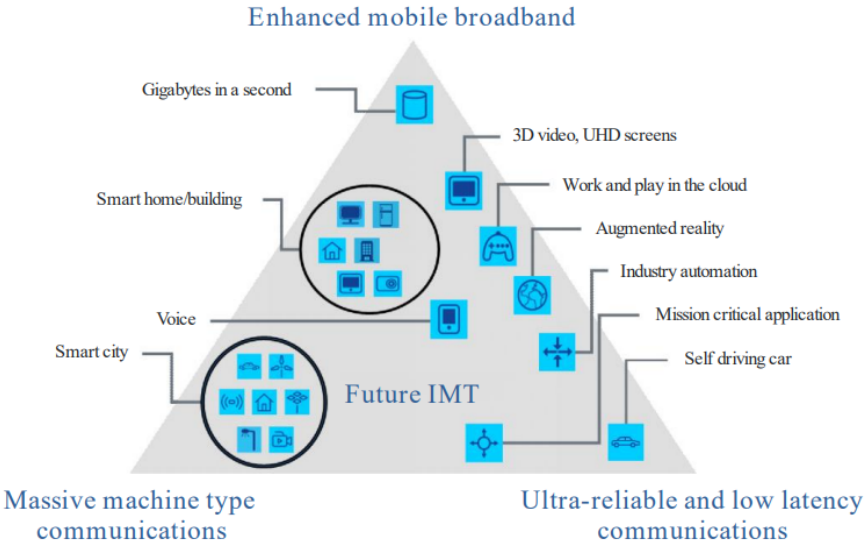


Figure 1.1: 5G applications

1.2 Latency challenges and sources

Achieving ultra-low latency in mobile networks involves addressing delays that occur at various layers and components of the system. As shown in Figure 1.2 [2], these sources of delays include transmission delays (caused by the time required to send the packet unto the link), propagation delays (due to the physical distance between the sender and the receiver, and the nature of the signal), processing delays at intermediate nodes, and queueing delays that arise at bottleneck nodes which typically occurs when there is congestion.

To meet the requirements of low-latency applications, substantial efforts have been made within the 5G architecture to reduce latency. These include innovations such as edge computing to shorten the distance between user and application servers, network slicing for dedicated resource allocation, mini-slot-based scheduling in the RAN for faster transmission opportunities, and Control and User Plane Separation (CUPS) to streamline packet handling [3]. Despite these advancements, one of the persistent challenges remains the delays caused by congestion, particularly in the Radio Access Network (RAN) due to capacity limitations. When traffic exceeds available capacity at a given point in the network, queues begin to build up. When packets are queued, latency is increased and in severe cases, packet loss occurs. In TCP-based flows, lost packets trigger retransmissions, which further increase end-to-end latency and consume additional resources. In UDP-based applications, the packets are not retransmitted, leading to a direct reduction in quality, where there could be frozen or skipped video frames in a streaming situation.

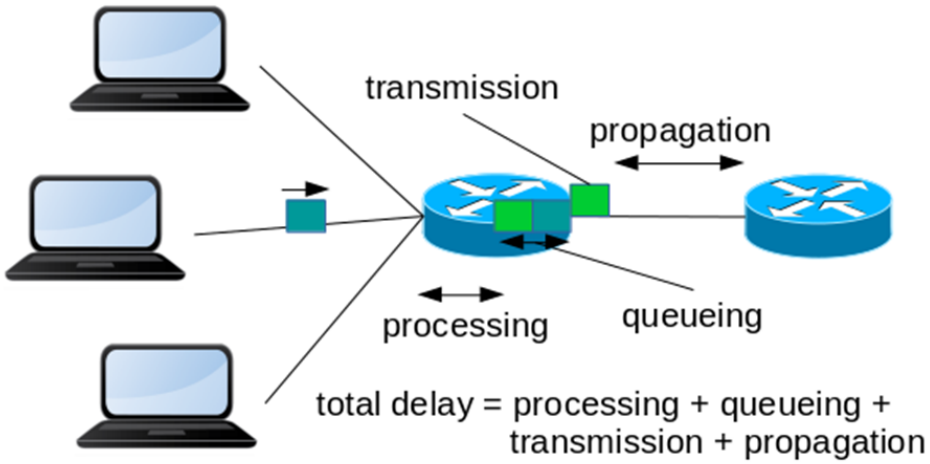


Figure 1.2: Sources of latency in the network

This thesis investigates two techniques designed to reduce delays caused by insufficient capacity and congestion at bottleneck nodes; network nodes where the volume of incoming traffic exceeds the available capacity.

1.3 Traditional congestion control and rate adaptation

A common approach to managing congestion-related delays is to adapt the sending rate at the application to match the available network capacity. To do this effectively, the sender needs timely and accurate feedback about the state of the network. Based on this feedback, the application reduces its transmission rate during periods of congestion and subsequently increases it according to the congestion control algorithm.

Traditional TCP congestion control mechanisms, such as TCP Reno and TCP Cubic, infer congestion through indirect signals such as packet loss or increased round-trip time (RTT). These algorithms typically respond to loss events by reducing the sending rate by a fixed proportion (50% in TCP Reno and 30% in TCP Cubic) [4]. While this method improves latency, prevents further packet losses and prevents network collapse, it also leads to slow recovery of the data rate, under-utilization of available bandwidth, and high variability in throughput. To mitigate packet loss and maintain high utilization, network devices often use large buffers. However, this introduces a well-known trade-off: large buffers reduce packet loss but increase latency due to longer queueing delays, whereas small buffers reduce delay but may cause frequent packet drops and under-use of link capacity.

Rate adaptation mechanisms such as Bottleneck Bandwidth and Round-trip propagation time (BBR) infer congestion by estimating the available bandwidth and RTT to determine the optimal sending rate that maximizes throughput while maintaining a minimal RTT. Similarly, Google Congestion Control (GCC), which is designed for real-time communication, relies on measurements of one-way delay to detect congestion periods and adjust its sending rate accordingly.

Essentially, current mechanisms used in mobile networks rely on their estimates of available

capacity, using packet loss patterns, RTT measurements, or heuristic-based models. However, without explicit feedback from the network, these estimations can lead to suboptimal rate choices that may either overload the network or fail to utilize its full capacity. This is especially important for mobile networks that have highly dynamic wireless channel conditions.

1.4 Network-Assisted Rate Adaptation - L4S and ANBR

To improve the accuracy of congestion detection and rate adaptation, explicit feedback from the network is highly beneficial. Network-assisted mechanisms provide more reliable congestion signals, allowing applications to make better-informed rate adjustments. Two emerging solutions that take this approach are Low Latency Low Loss Scalable Throughput (L4S) and Access Network Bitrate Recommendation (ANBR). These techniques aim to enhance performance by enabling a more collaborative interaction between the application and the network, thereby reducing latency, packet loss and preventing congestion buildup, while aiming to maximize throughput.

L4S is a network mechanism standardized by the Internet Engineering Task Force (IETF) in RFC 9330 [5], RFC 9331 [6] and RFC 9332 [7], and standardized in Release 18 of the 3GPP TS 23 501 document (which is the main architecture document for 5G) [8]. L4S uses Explicit Congestion Notification (ECN) to provide early congestion signals to senders without relying on packet loss. This enables applications to achieve low queueing delays while maximizing throughput. Specifically, L4S uses Active Queue Management (AQM) mechanisms at bottleneck nodes to mark packets in the IP header when queues exceed very low thresholds. Receivers detect these ECN marks and relay the ECN feedback to the sender, which then adjusts its transmission rate in proportion to the amount of ECN marks by using scalable congestion control algorithms such as TCP Prague [9], UDP Prague [10] or the L4S-adapted Self-Clocked Rate Adaptation for Multimedia (SCReAM) [11]. Unlike traditional TCP congestion control algorithms that reduce their transmission rate by a fixed proportion and require time to increase their rates again, scalable congestion control algorithms adjust the sending rate based on the extent of congestion, thus also ensuring that throughput is maximized. Operating at the IP layer, L4S can be applied in both fixed and mobile networks.

In contrast, ANBR offers an alternative approach in which the RAN evaluates network conditions and recommends optimal bitrates to the User Equipment (UE) through Media Access Control Control Element (MAC CE). The UE then communicates these recommendations to the application server, enabling rate adaptation that reflects real-time RAN conditions. ANBR is standardized by 3GPP for multimedia telephony [12] and 5G Media Streaming (5G MS) [13].

1.5 Research gap

L4S and ANBR are both network-assisted mechanisms aimed at reducing latency and packet loss while maximizing throughput. Although they share similar goals, L4S and ANBR differ in their technical design and how they are integrated into the network. L4S has gained significant attention from both industry and academia in recent years, with multiple real-

world implementations and trials already in progress. In contrast, ANBR remains relatively unexplored, with limited documentation, minimal academic evaluation, and minimal deployments or practical trials.

Despite the similarity and potential in the two mechanisms, there is a noticeable lack of comparative research that explores their performance, trade-offs, and interaction. Additionally, there is limited understanding of how L4S and ANBR could perform when deployed in combination. It is also unclear under which scenarios one mechanism may offer advantages over the other. Exploring their joint deployment is also important, as a well-designed combination could potentially bring greater gains than using each of the technologies independently by taking advantage of their strengths.

Addressing these gaps is important for stakeholders across the telecommunications ecosystem, including mobile network operators, equipment vendors, and application developers. This will ensure that these mechanisms are deployed effectively and tailored to the requirements of applications. It will also guide the development of more effective rate adaptation strategies by ensuring that latency is minimized while throughput is maximized.

This thesis aims to address this research gap by evaluating the performance of L4S and ANBR, both individually and in combination.

1.6 Research questions

The main question this research aims to address is:

How can a mobile network operator integrate L4S and/or ANBR into its network?

Based on this, the following sub-questions are derived:

SQ1: How do L4S and ANBR affect key network performance indicators such as latency, packet loss, and throughput?

SQ2: What are the comparative advantages and disadvantages of L4S and ANBR?

SQ3: Which application types or network scenarios stand to benefit most from the use of L4S and/or ANBR?

SQ4: What deployment strategies should be considered by mobile network operators (with a particular focus on KPN), equipment vendors, and application developers to use these mechanisms most effectively?

1.7 Approach

This section presents the approach used for this thesis. It consists of three main aspects: theoretical analysis, practical analysis, and interpretation of the results.

Theoretical Analysis
Review of Literature and Standards: A review of the IETF and 3GPP specifications for L4S

and ANBR is conducted alongside academic publications to establish the design principles, architecture, implementation, and experimental approaches.

Design comparison: L4S and ANBR are compared, and their advantages and disadvantages are highlighted.

Practical Analysis
Simulation using ns-3: A custom 5G simulation environment is developed using the ns-3 simulator to evaluate the performance of L4S, ANBR, and their combination. The evaluation is carried out using UDP traffic, with a particular focus on low-latency, high-throughput applications. In particular, the simulation was modeled using traffic characteristics of AR/VR use cases, which have low-latency and high throughput requirements.

Testlab Evaluation: Real-world validation of L4S for UDP-based high-rate applications was carried out in the KPN 5G Standalone (5G SA) test network. Since Huawei, the radio equipment vendor for KPN, did not support L4S, an Ericsson base station was installed in the KPN test network to enable the evaluation.

ANBR supported by Huawei, was not evaluated in the test network due to the lack of a UE equipment vendor supporting ANBR for high-rate applications. Although a UE vendor with ANBR support for Voice over LTE was available, additional constraints prevented usable results from being obtained from this test.

Results and conclusion
Results from both the simulation and the testlab were analyzed to identify performance trends, validate theoretical assumptions, and assess the impact of L4S and ANBR.

Based on the findings, recommendations were provided for KPN.

1.8 Document structure

The remainder of the thesis is organized as follows. Chapter 2 reviews the standardization efforts related to both L4S and ANBR, examines the available literature, and explores their current adoption in the industry. Chapter 3 provides a detailed explanation of the principles and operational processes of L4S and ANBR, starting with the foundational knowledge required to understand these mechanisms, followed by a description of their processes. A comparative analysis is then presented to highlight their advantages and disadvantages, and a combined use of L4S and ANBR is proposed. Chapter 4 describes the simulation and testbed setups used for the experiments, while Chapter 5 presents and discusses the experimental results obtained from both setups. Finally, Chapter 6 concludes the thesis by addressing the research (sub)questions posed in Chapter 1 and giving recommendations for future work.

2

Review of Literature and Standards

This chapter presents a review of the current state of standardization, industry adoption, and research related to L4S and ANBR. The goal is to summarize the state-of-the-art for both mechanisms and provide a reference point for the comparative analysis and experimental work presented in later chapters.

2.1 Standardization of L4S

As mentioned in Chapter 1, L4S has been standardized by the IETF through a set of RFCs: RFC 9330 [5], RFC 9331 [6], and RFC 9332 [7]. These RFCs collectively define the architecture, requirements, and operational guidelines for deploying L4S in IP networks.

- RFC 9330 defines the overall architecture of L4S, including its deployment model and recommended deployment practices.
- RFC 9331 specifies the use of the ECN field in the IP header of the data packets to support L4S in order to provide congestion signals to the application.
- RFC 9332 proposes a Dual Queue Coupled AQM mechanism for IP networks that enables L4S flow isolation and coexistence between non-L4S and L4S traffic.

As noted in Chapter 1, L4S has been adopted into mobile network standards for Extended Reality (XR) services through 3GPP Release 18. 3GPP TS 23.501 [8] specifies that the ECN marking logic for L4S can be applied separately to a specific QoS Flow in both the Uplink (UL) and Downlink (DL), with congestion detection done by the RAN and ECN marking performed either by the RAN or the User Plane Function (UPF). The actual AQM behavior and the criteria for detecting congestion, such as how and when to apply ECN markings, are left to vendor-specific implementations.

Scalable congestion control mechanisms are key to L4S. The IETF has outlined specific requirements for these algorithms termed the "Prague Congestion Control Algorithm" [9]. These requirements ensure that a congestion control algorithm is suitable for L4S by highlighting properties such as responsiveness to ECN feedback, proportional rate reduction upon ECN feedback, controlled rate increase during low periods of congestion, and so on. Algorithms such as TCP Prague [9], UDP Prague [10] and the L4S variant of SCReAM (SCReAMv2) [11] are examples that adhere to these principles.

2.2 Industry adoption of L4S

The adoption of L4S across the telecommunications industry is gaining momentum, with a broad range of stakeholders - including UE manufacturers, application developers, network equipment vendors, and network operators - actively demonstrating its practical benefits through trials and early-stage deployments. The following paragraphs highlight key parts of the ecosystem and notable efforts related to L4S implementation and evaluation in the telecoms industry.

Device and Application Support: Apple has incorporated L4S support into iOS 17, iPadOS 17, macOS Sonoma, and tvOS 17 , aiming to improve the performance of latency-sensitive applications such as FaceTime [14]–[16].

NVIDIA has integrated L4S support into its GeForce NOW cloud gaming platform. Users can enable L4S through the streaming quality settings to benefit from reduced latency and packet loss during gameplay, thereby enhancing the overall gaming experience [15], [17].

Network Equipment Manufacturer Support: Nokia has played a key role in driving L4S adoption across both broadband and mobile networks. Its white paper [18] outlines how L4S can deliver ultra-low latency with minimal impact on throughput, making it suitable for real-time applications like cloud gaming, XR, and industrial automation. Nokia has also collaborated with operators such as Vodafone [19] and Elisa [20] to evaluate L4S performance in real-world trial environments.

Ericsson has introduced L4S support in its RAN nodes to enable time-critical 5G applications, with the capability to steer L4S traffic to a dedicated QoS Flow and apply ECN marking using the 5QI associated with that flow, as defined in the standards. Its white paper [3] highlights the importance of L4S in achieving ultra-low latency for use cases including XR and cloud gaming. Ericsson has also conducted trials and experiments in collaboration with Etisalat[1] [21] to assess the practical impact of L4S in operational networks.

Network Operator Support: Vodafone, in collaboration with Nokia, conducted a successful L4S trial over residential broadband using Passive Optical Network (PON) technology. The results demonstrated reductions in latency [19].

Etisalat, in collaboration with Ericsson, implemented L4S in a live 5G standalone (SA) network. The trial reported over a 50% reduction in latency for cloud gaming applications, reinforcing the value of L4S in delivering real-time performance [21].

Elisa and Nokia showcased L4S in a live 5G SA setup at the Nokia Arena in Tampere, Finland. During a real-time video streaming demonstration under congested conditions, the L4S-enabled smartphone was able to stream content instantly, whereas the non-L4S device experienced noticeable buffering [20].

[1] Etisalat rebranded to "e&" in 2022

2.3 Review of work related to L4S

Several experimental and simulation-based studies have demonstrated the potential of L4S to reduce latency while maximizing throughput. This section reviews selected works that are relevant to the scope of this thesis.

The most relevant study for the L4S component of this thesis is the work by Brunello et al. [4], which investigated the integration of L4S into 5G networks for Augmented Reality gaming. Their study identified key deployment challenges, in particular, the limitations caused by encryption below the Packet Data Convergence Protocol (PDCP) layer, which would make it difficult to mark packets at the Radio Link Control (RLC) layer where packet queueing takes place. To address this issue, the authors proposed either moving the queue to the PDCP layer or measuring queueing delay at the RLC layer and applying ECN marking at the PDCP layer prior to encryption. Since moving the queue would require costly hardware modifications, they proceeded with measuring delay at the RLC layer and marking packets at the PDCP layer. SCReAM was used as the scalable congestion control mechanism in their evaluation. Simulations, carried out using a proprietary tool, showed significant reductions in latency when L4S was used, compared to scenarios where L4S was not used. Furthermore, the study demonstrated that changing the ECN marking queueing thresholds had a clear impact on performance. Increasing the threshold range led to higher throughput but also had more delay, while reducing the threshold range resulted in lower latency at the cost of some throughput.

The work by Pan et al. [22] examined L4S standardization under 3GPP Release 18 and proposed a roadmap for implementation in XR services. Key L4S challenges identified include effective ECN marking strategies, congestion detection strategies, scalable congestion control design, and maintaining optimal performance even when there is user mobility. Using a testbed with Real-Time Protocol (RTP)-based WebRTC transmission and a Dual-Queue Coupled AQM, L4S was shown to outperform Google Congestion Control (GCC) in bandwidth efficiency and video stalling. Although a slight drop in video quality was observed with L4S, the improvements in latency and playback stability were significant. The study also evaluated a combined approach using both L4S and GCC, which achieved better bandwidth utilization than L4S alone. This is because L4S depends on ECN feedback for congestion control, which can lead to a slow recovery of bitrate post congestion, while GCC also uses additional information such as the change in the one-way delay to accelerate bitrate increase.

Son et al. [23] developed a WebRTC congestion control system for 5G video streaming. The system included an L4S-enabled queue management module, ECN feedback through Real-time Transport Control Protocol (RTCP), and sender-side modules for encoding, pacing, and rate adaptation. The setup demonstrated better delays and link utilization compared to GCC, although with temporary rate reductions in L4S due to L4S's early reaction to congestion.

Another study by Srivastav et al. [24] evaluated TCP Prague (a scalable congestion control algorithm for L4S) over 60 GHz mmWave links using traces emulated on a CloudLab testbed, which is a National Science Foundation funded cloud-based platform for networking research. The results confirmed that TCP Prague achieves lower delays than TCP Cubic and BBR. TCP Cubic which is based on packet loss fills the bottleneck buffer and had the highest

delay. BBR tries to find a point with the minimum RTT and highest bandwidth. To achieve this, it periodically enters probe phases where it reduces its sending rate to a great extent to re-estimate the minimum RTT, which in turn causes huge degradations in throughput during those periods. L4S, on the other hand, reduces its sending rate in accordance to the level of ECN marking, thereby avoiding the drastic rate reductions. However, the study also highlighted that L4S had fairness issues where some L4S flows were starved of bandwidth, which was made worse by sudden capacity drops caused by mmWave fading.

Monteiro et al. [25] evaluated L4S in a private 5G standalone network designed for industrial use cases. They used a virtualized testbed with programmable P4 switches and implemented a Dual-Queue Coupled AQM to differentiate between L4S and non-L4S traffic. TCP Prague was also used as the scalable congestion control algorithm to handle L4S flows. The results showed that L4S significantly reduced packet loss and improved video quality for real-time video in congested scenarios compared to a case where L4S was not used.

2.4 Standardization of ANBR

ANBR is not a new concept but has been evolving within 3GPP standards across multiple releases. It was first introduced in 3GPP TS 26.114 [12] and 3GPP TS 26.300 [26] in Release 14 as a mechanism for adaptive codec bitrate control in multimedia telephony services such as Voice over LTE (VoLTE) and Video over LTE (ViLTE) . In Release 15, this support was extended to Voice over NR (VoNR) and Video over NR (ViNR).

ANBR was also discussed in 3GPP TR 26.919 [27] as a more reliable alternative to ECN-based rate adaptation. The report emphasizes that ANBR provides explicit bitrate recommendations to the application which would prevent over or underestimation of the capacity. It was also emphasized that ECN packets could get dropped by intermediate nodes [27].

The control signaling structure for ANBR was formally defined in the MAC layer through MAC CEs. This was first specified for LTE in 3GPP TS 36.321 (Release 14) [28] and subsequently for 5G NR in 3GPP TS 38.321 (Release 15) [29].

In the context of 5G Media Streaming (5GMS), ANBR has been reinforced as a key enabler for tight integration between the application and the network. Both 3GPP TS 26.501 (Release 16) [30] and 3GPP TS 26.510 (Release 18) [13] emphasize it's role in enabling dynamic rate adaptation of streaming quality based on network conditions.

2.5 Industry adoption of ANBR

While ANBR is a standardized feature introduced in 3GPP Release 14 (4G), its commercial deployment remains limited. Nonetheless, a few industry stakeholders have begun higlighting ANBR and integrating it into their platforms, especially within IMS-based voice services and emerging 5G media architectures.

Device and Equipment Manufacturer Support: Google's Pixel 7 and newer devices support ANBR for Voice over LTE (VoLTE) as part of Release 14. However, widespread deployment and operator-side enablement are still lacking.

In a white paper, MediaTek highlights the use of ANBR for VoNR [31]. They highlight its use in adjusting codec settings dynamically, thereby improving call quality under challenging radio conditions. The same mechanism is noted as applicable to high data-rate services such as VR streaming.

5G Media Streaming: The 5G Media Action Group (5G-MAG) is actively working on standard-compliant implementations of the 5G MS architecture, based on 3GPP Release 17 [32]. One of the key goals of 5G MS is to enable tight collaboration between the application and the 5G network to enhance streaming services. ANBR is one of the features in this architecture, allowing the network to provide real-time bitrate recommendations to media applications. This would improve the adaptability of streaming clients, particularly for XR and immersive video content, by responding to changing network conditions more effectively.

2.6 Review of work related to ANBR

Unlike L4S, the concept of ANBR has received relatively limited attention in academic literature. Only a few studies have referenced ANBR, primarily in the context of VoLTE and VoNR scenarios. In these cases, rate adaptation for IMS services does not primarily target reducing queueing delays, but rather focuses on maintaining acceptable voice quality under varying radio conditions.

At the cell edge, poor signal quality leads to fewer allocated resource blocks and smaller Transport Block Sizes (TBS). High-bitrate codecs may struggle to transmit reliably in such conditions, increasing the risk of packet loss and degraded voice quality. By lowering the codec bitrate, fewer resources and transport block sizes are required per packet, which reduces the likelihood of packet loss and helps maintain reliable voice service.

Prasad et al. [33] propose an adaptive rate switching technique for Enhanced Voice Services (EVS) codecs over VoLTE. Their rate adaptation method allows the UE to dynamically switch between EVS 24.4 kbps and a more robust EVS 13.2 kbps Channel-Aware Mode (CAM) during poor network conditions, based on the UE's observed RTP packet loss, which was achieved by varying the Reference Signal Received Power (RSRP). This approach aims to maintain acceptable voice quality even under degraded conditions. Although the paper does not explicitly evaluate ANBR, it does mention ANBR and ECN as network-based alternatives to the UE based method. They emphasize that with the UE based approach, UE vendors are not reliant on network operators (through the RAN, Evolved Packet Core (EPC), and IP Multimedia Subsystem (IMS)) for the rate switching. Results showed that codec rate adaptive switching outperformed static codec configurations in Mean Opinion Score (MOS), as lower rates improved resilience to packet loss under poor RF conditions.

Xu et al. [34] investigate the effectiveness of ANBR compared to handover strategies in maintaining QoE at the cell edge for VoNR. Their study evaluates two codec modes: EVS and AMR-WB. In their approach, the RAN sends codec rate recommendations to the UE based on metrics such as Block Error Rate (BLER) and downlink RSRP, enabling the UE to adapt its codec rate accordingly. To emulate cell center and cell edge conditions, the authors varied the RSRP and assessed voice quality using the Mean Opinion Score. The study found that handover-based strategies remained more effective than ANBR/rate switching in cell-edge scenarios. Specifically, when RSRP fell below -125 dBm, both EVS and AMR-WB showed a

rapid decline in MOS across all their respective codec rates. Furthermore, they observed that lowering the codec rate did not yield a significant improvement in speech quality when RSRP was below −127 dBm. Given that handover thresholds are typically set below −127 dBm, the authors concluded that handover is generally preferable in such conditions, as reducing the codec rate alone can significantly degrade speech quality at the cell edge.

Karjee et al. [35] references ANBR as part of a broader cross-layer rate adaptation (CLRA) mechanism for Voice Over IP (VoIP) and other application traffic in 5G NR. Although they do not explicitly evaluate ANBR in isolation, their proposed architecture incorporates bitrate recommendations from the gNodeB and uses reinforcement learning at the UE to select optimal codec rates for VoIP. Additionally, the UE uses the recommended bitrate to estimate available throughput and dynamically allocates it between foreground applications (Hotstar streaming, web browsing) and background applications (Play Store, Gmail) according to the predicted demand of the applications. Their framework provides an end-to-end view of how ANBR signaling from the RAN can be utilized within the UE.

2.7 Summary

The literature shows that L4S has been widely studied in both academic and industry contexts. Research has highlighted its integration into mobile networks, its impact on latency and throughput, and its benefits for low-latency applications like XR and cloud gaming. One study showed that using SCReAM scalable congestion control with L4S reduces latency compared to SCReAM without L4S. While throughput was slightly lower than SCReAM without L4S capability, it remained acceptable. L4S also improves playback stability compared to Google Congestion Control, with only small trade-offs in video quality and temporary rate reductions. Furthermore, when using TCP Prague, L4S achieves the best latency performance compared to TCP Cubic and BBR, while avoiding the throughput reductions that BBR experiences when estimating the minimum round-trip time.

In contrast, ANBR has received far less attention in academic research. Although it is standardized by 3GPP and is gradually being adopted in commercial systems, most studies focus on its application to voice services such as VoLTE and VoNR, where it adapts codec rates based on radio conditions. Comprehensive evaluations of ANBR are limited, and its potential use cases beyond voice services, particularly for low-latency, high-bitrate applications, remain largely unexplored. Moreover, the existing literature presents conflicting views on rate adaptation for VoLTE and VoNR. One author reports that adaptive codec rate switching improves mean opinion score, while another finds that codec rate adaptation at the cell edge can degrade speech quality, with handover-based approaches performing more effectively.

While this thesis does not evaluate ANBR features for VoLTE and VoNR, future research could investigate whether ANBR codec rate adaptation can improve voice quality for cell-edge users, particularly given the conflicting findings in the literature. Moreover, the effectiveness of handover-based approaches may be limited in high-traffic scenarios where the neighboring cells may be congested.

As mentioned in Chapter 1, this thesis examines the advantages and disadvantages of L4S and ANBR in terms of design and operation, and explores how they can co-exist and be used in mobile networks. One of the goals is to understand how combining their strengths could

enhance bitrate adaptation and improve overall quality of service and experience for end users.

3

Theoretical Framework

This chapter provides an overview of the technical foundations relevant to this thesis. It begins by describing the end-to-end data flow, followed by an outline of the 5G system architecture, covering both the core network and the radio access network. The chapter then presents the operational principles and architectural components of L4S and ANBR. A comparative analysis of L4S and ANBR is conducted to highlight their respective strengths and limitations. Finally, a hybrid mechanism is proposed that combines both techniques.

3.1 End-to-End Data Flow

To understand the role of L4S and ANBR in modern networks, it is useful to examine on high level, the broader context of data transfer from the application server to the UE. This includes an overview of the application, transport, and network layers.

Today, the majority of internet traffic is made up of audio, video, and image content [36]. This traffic comes from a variety of services, including video-on-demand (e.g., Netflix, YouTube), conversational applications (e.g., VoIP (Microsoft Teams, FaceTime), VoLTE, VoNR), and live streaming (e.g., cloud gaming, AR/VR). In addition to media content, some applications also transmit real-time control or pose data; for example, in cloud gaming, AR/VR, or remote control systems, where the client sends control input and the server responds accordingly [37].

As stated in Chapter 1, these applications are sensitive to latency, often requiring round-trip delays on the order of milliseconds. High latency can result in delayed interactions, frozen frames, or degraded user experience, especially in conversational and interactive scenarios. At the same time, some of these applications demand high bitrates to deliver high-quality media content. Bitrate refers to the number of bits transmitted per second; higher bitrates generally enable better visual or audio quality. For example, a High-Definition (HD) video requires a higher bitrate than a low-resolution video.

Understanding how this data moves through the network is important. The following paragraphs provide a high-level overview of the flow of data from the application layer down through the transport and network layers.

Application Layer: Video Encoding and Compression

At the application layer, the raw video content is encoded and compressed to reduce bandwidth usage while preserving visual quality. A video is composed of a sequence of frames displayed at a specific frame rate, creating the perception of motion. Each frame consists of a grid of pixels; for instance, a 1080p HD frame contains 1920 x 1080 pixels. If each pixel uses 24 bits (8 bits per Red Green Blue (RGB) channel), then a single uncompressed frame would require roughly 50 Mbps. Streaming 24 such frames per second would result in a data rate of over 1 Gbps, which is not practical for real-time transmission [36].

To address this, video compression is used. Encoding refers to the process of converting raw video data into a compressed digital format suitable for network transmission, storage, and playback. Modern video codecs such as H.264 AVC and H.265 HEVC significantly reduce data size by removing redundancy within and between frames, while maintaining acceptable quality. At the receiver side, the video is decoded and decompressed before being rendered for playback.

Transport Layer Protocols: TCP and UDP

After video compression, the resulting data is divided into smaller segments and encapsulated into transport-layer packets, with a transport-layer header added using either TCP or UDP, depending on the application's requirements.

- **TCP:** provides a reliable, connection-oriented service. It ensures that data is delivered in order, without errors or losses, by using acknowledgments, retransmissions, and built-in congestion control mechanisms. Video data is segmented into packets, each appended with a TCP header that contains sequencing and error-checking information. These features make TCP suitable for video-on-demand or file transfers, where reliability is more important than delay.
- **UDP:** In contrast to TCP, UDP provides a connectionless, best-effort delivery service with minimal protocol overhead. It does not guarantee the delivery or ordering of packets, transmitting them immediately as they are generated according to the application's rate. Its low overhead and absence of retransmissions make it suitable for real-time applications such as VoIP, cloud gaming, and live video streaming, where minimizing latency is more important than ensuring perfect reliability. However, due to the lack of built-in congestion control mechanisms, it is the responsibility of the application layer to implement its own strategies for congestion control or rate adaptation.

Network Layer and Below

At the network layer, the Internet Protocol (IP) encapsulates transport-layer packets into IP packets, adding IP headers that include source and destination addresses for routing across the internet. These packets are then handed down to the data link and physical layers, where they are encapsulated into frames and transmitted over physical media such as fiber, copper, or wireless channels.

Congestion Control and Rate Adaptation

Congestion control and rate adaptation play a critical role in maintaining a stable streaming performance. It helps manage network bandwidth efficiently, reduce latency and packet loss, avoid congestion collapse, and ensure fair bandwidth sharing among multiple users and applications.

TCP-based Congestion Control: uses algorithms like slow start, congestion avoidance, fast retransmit, and fast recovery to dynamically adjust the sending rate based on network feedback. Traditional TCP algorithms such as TCP Reno and TCP Cubic reduce their transmission rates by a fixed amount in response to congestion signals, usually detected through packet loss. For example, TCP Reno halves its congestion window when loss is detected, while TCP Cubic reduces it by about 30%. After this reduction, the sending rate slowly ramps up again (Additive Increase Multiplicative Decrease), which can take a long time to fully utilize throughput again because it reduces by a fixed rate each time there is congestion. TCP Prague is the TCP-based algorithm for L4S. Other examples of TCP algorithms include BBR, and New Reno.

One way mobile network operators aim to improve TCP performance is through TCP Optimization (also known as TCP acceleration), which gives them greater control over TCP connections. In TCP optimization, a buffer is introduced within the network environment to temporarily hold data arriving from the application server [38].

In this setup, the mobile network sends TCP acknowledgments on behalf of the UE to the application server even before the UE has actually received the corresponding packets. This allows the application server to continue to transmit at a full rate, while the network manages the adaptation of the transmission rate. As a result, the buffer fills quickly. This mechanism not only accelerates the TCP slow-start phase, leading to a faster ramp-up of the transmission rate towards the RAN and quicker utilization of the available bandwidth, but also enables faster retransmission of lost packets.

Unlike traditional TCP congestion control in mobile networks, which generally interprets packet loss as a sign of congestion, TCP Optimization can use radio-aware insights to differentiate between congestion-related and non-congestion-related losses. This enables the operator to adapt the transmission rate more efficiently based on network conditions, radio-aware bandwidth estimation, and congestion control algorithms.

Application-Level Congestion Control for UDP: is necessary because UDP itself does not provide congestion control. Real-time multimedia applications that use UDP often implement custom mechanisms to adapt their bitrate, frame rate, or resolution in response to network conditions. These adaptations are typically based on observed throughput, packet loss, delay, or buffer occupancy. One example is SCReAM, developed by Ericsson, which uses Real-time Transport Control Protocol (RTCP) feedback and delay estimates from the receiver to dynamically adjust the sending rate. SCReAM also has an L4S variant as discussed in previous sections [11]. Another example is UDP Prague, the UDP congestion control algorithm for L4S.

3.2 Overview of the 5G System Architecture

The 5G system consists of two major components: the 5G Core (5GC) and the Next Generation Radio Access Network (NG-RAN). The 5GC is a cloud-native, Service Based Architecture (SBA), where Network Functions (NFs) interact through APIs. The NG-RAN connects the UE to the core network via gNodeBs, which implement the radio access protocol stack and perform scheduling, radio resource management, and link adaptation.

3.2.1 5G Core Network Functions

The 5GC consists of a set of modular, Service Based Architecture NFs. Each NF performs different functions, and interacts through well-defined interfaces [39]. Figure 3.1 [40], illustrates the key NFs and their interconnections.

This subsection summarizes the major NFs that are relevant to session management, mobility, and policy enforcement.

- **User Plane Function (UPF):** The UPF handles the user plane and acts as the gateway to the data network. It handles traffic forwarding, inspection, and usage reporting. It is controlled by the Session Management Function (SMF) and may be deployed at the network edge or in centralized data centers depending on traffic optimization requirements.
- **Access and Mobility Management Function (AMF):** The AMF manages the signaling between the UE and the 5GC, and interfaces with the RAN over the N2 interface and directly with the UE through the N1 interface as seen in Figure 3.1. It plays a central role in UE registration, authentication, and mobility management of the UE.
- **Session Management Function (SMF):** The SMF handles the establishment, modification, and release of Protocol Data Unit (PDU) sessions, which connect the UE to Data Networks (DN) through the UPF. It also performs IP address allocation for IP-type PDU sessions and determines how user-plane traffic should be steered through selected UPFs. In collaboration with the Policy Control Function (PCF), the SMF enforces QoS and charging rules.
- **Policy Control Function (PCF):** The PCF is the central function for policy and QoS control in the 5GC. It provides authorization for QoS priority levels and charging control per session. It interacts with the SMF and AMF to enforce access, mobility, and session-related policies. The PCF also communicates with UEs (through the AMF) to distribute policies such as network slice selection, non-3GPP access rules, and data network name selection.
- **Network Exposure Function (NEF):** The NEF provides a standardized interface to expose network capabilities and events to internal and external applications. Similar to the Service Capability Exposure Function (SCEF) in 4G Core, it allows authorized applications to access certain information from the network. Additionally, the NEF enables applications to request specific QoS treatments.

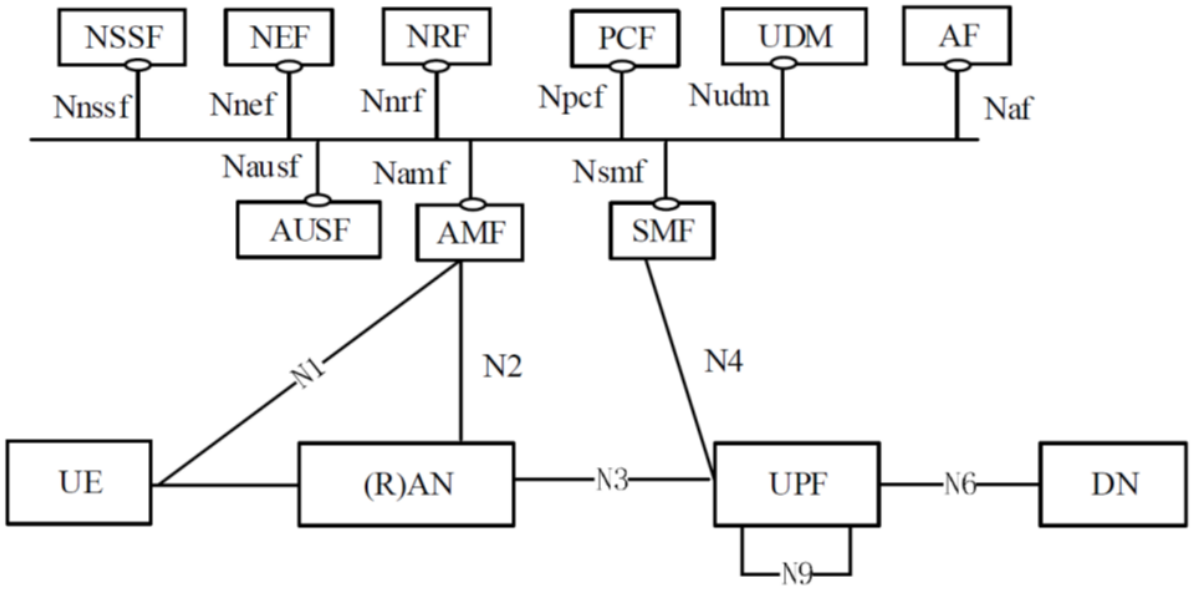


Figure 3.1: 5G Architecture showing the Core Network Functions

3.2.2 The QoS Framework

A PDU session is the logical connection between a UE and the UPF in the 5G system architecture. The SMF initiates the establishment of the PDU session by allocating an IP address to the UE and defining the QoS parameters. As seen in Figure 3.2 [39], each session may contain multiple QoS flows, which are the smallest unit of QoS differentiation. These flows are uniquely identified by a Qos Flow Identifier (QFI), and are then mapped to one or more Data Radio Bearers (DRBs) in the RAN, depending on the RAN configuration and service requirements.

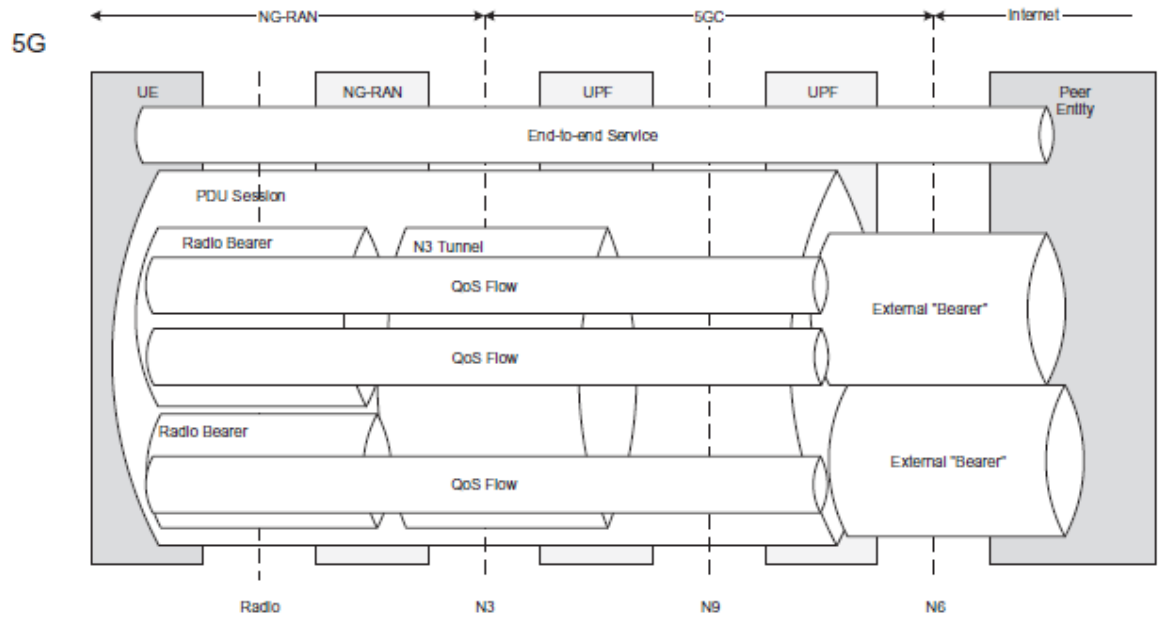


Figure 3.2: 5G QoS Framework

The QoS framework in 5G is flow-based and supports both Guaranteed Bitrate (GBR) and non-GBR flows. During PDU session establishment, QoS rules and QFIs are delivered to the UE through Non-Access Stratum (NAS) signaling via the AMF, and are used to control uplink traffic marking and handling. These QoS flows ensure consistent treatment across the 5G core and the RAN.

When a data packet arrives at the UPF, particularly in the downlink, it is subject to classification and enforcement of QoS policies. The UPF uses Packet Detection Rule (PDR) installed by the SMF to classify the packet into a corresponding QoS flow, which is identified by a QFI. The classification is based on packet attributes such as source/destination IP, port numbers, protocol or ECN-Capable Transport Codepoint (in the case of L4S).

Each QFI is associated with a QoS Enforcement Rule (QER), which defines parameters such as:

- Maximum and Guaranteed Bitrates (MBR, GBR) for UL and DL
- Priority level, delay budget, and packet error rate parameters (i.e; 5QI values)

The QFI is inserted into the GPRS Tunnelling Protocol - User Plane (GTP-U) header over the N3 interface and delivered to the gNodeB. At the gNodeB, the Service Data Adaptation Protocol (SDAP) maps the QoS flows to the appropriate DRB according to the established policies and configurations.

Role of the PCF and SMF in QoS Control: The PCF is a central policy decision point, which coordinates with the SMF and UPF to maintain the subscriber-specific service requirements throughout the session. The SMF interacts with the PCF to get the policy-based QoS configurations. These are provided:

1. During the initial PDU session establishment, based on subscription data from the Unified Data Management (UDM).
2. When a new QoS Flow is requested (e.g., due to demand from an application or an Application Function (AF) request).

The PCF responds with Policy and Charging Control (PCC) Rules, which include:

- QFI and associated QoS characteristics such as 5QI, Allocation and Retention Priority (ARP), GBR/MBR.
- Packet filters and precedence values for traffic classification.
- Session-AMBR and other operator-specific policy parameters.

3.2.3 Radio protocol stack in the NG-RAN

The NG-RAN is responsible for the radio interface between the UE and the 5GC. It has a multi-layered protocol stack. Each protocol layer as shown in Figure 3.3 [41], performs specific functions, and together they enable dynamic resource management, error correction, security, and quality of service enforcement [41]. This subsection provides an overview of the various radio protocol layers.

- **Physical Layer (PHY)**: The lowest layer of the radio protocol stack, the PHY layer handles the physical transmission and reception of radio signals. It performs functions such as modulation and demodulation, channel coding and decoding, and Multiple-Input, Multiple-Output antenna processing (MIMO).
- **Medium Access Control (MAC)**: The MAC layer coordinates the access to the shared radio channel. It is responsible for dynamic scheduling of uplink and downlink transmissions, Hybrid Automatic Repeat Request (HARQ) processes, multiplexing and demultiplexing of logical channels, and prioritization of traffic. It also handles Buffer Status Reporting (BSR) to report how much data is waiting in the RLC buffers.
- **Radio Link Control (RLC)**: The RLC layer provides data transfer services through three operational modes:
  - Acknowledged Mode (AM) for reliable, error-corrected transmission with retransmissions.
  - Unacknowledged Mode (UM) for real-time traffic where low latency is prioritized over reliability.
  - Transparent Mode (TM) for control plane data with minimal overhead.

  It also handles segmentation and reassembly of data units, ensuring size compatibility with underlying layers. Typically, packets are queued in the RLC layer before being scheduled by the MAC layer, with each user having its own separate queue [4].
- **Packet Data Convergence Protocol (PDCP)**: The PDCP layer provides higher-level functions such as header compression using schemes like Robust Header Compression (ROHC), ciphering and integrity protection of user and control plane data, and in-sequence delivery of packets.

- **Service Data Adaptation Protocol (SDAP)**: The SDAP layer enables the mapping of QoS Flows (identified by QFIs) to DRBs. It ensures that traffic with different QoS requirements is appropriately handled. SDAP can either multiplex multiple QFIs onto a single DRB or assign a dedicated DRB to a single QFI.

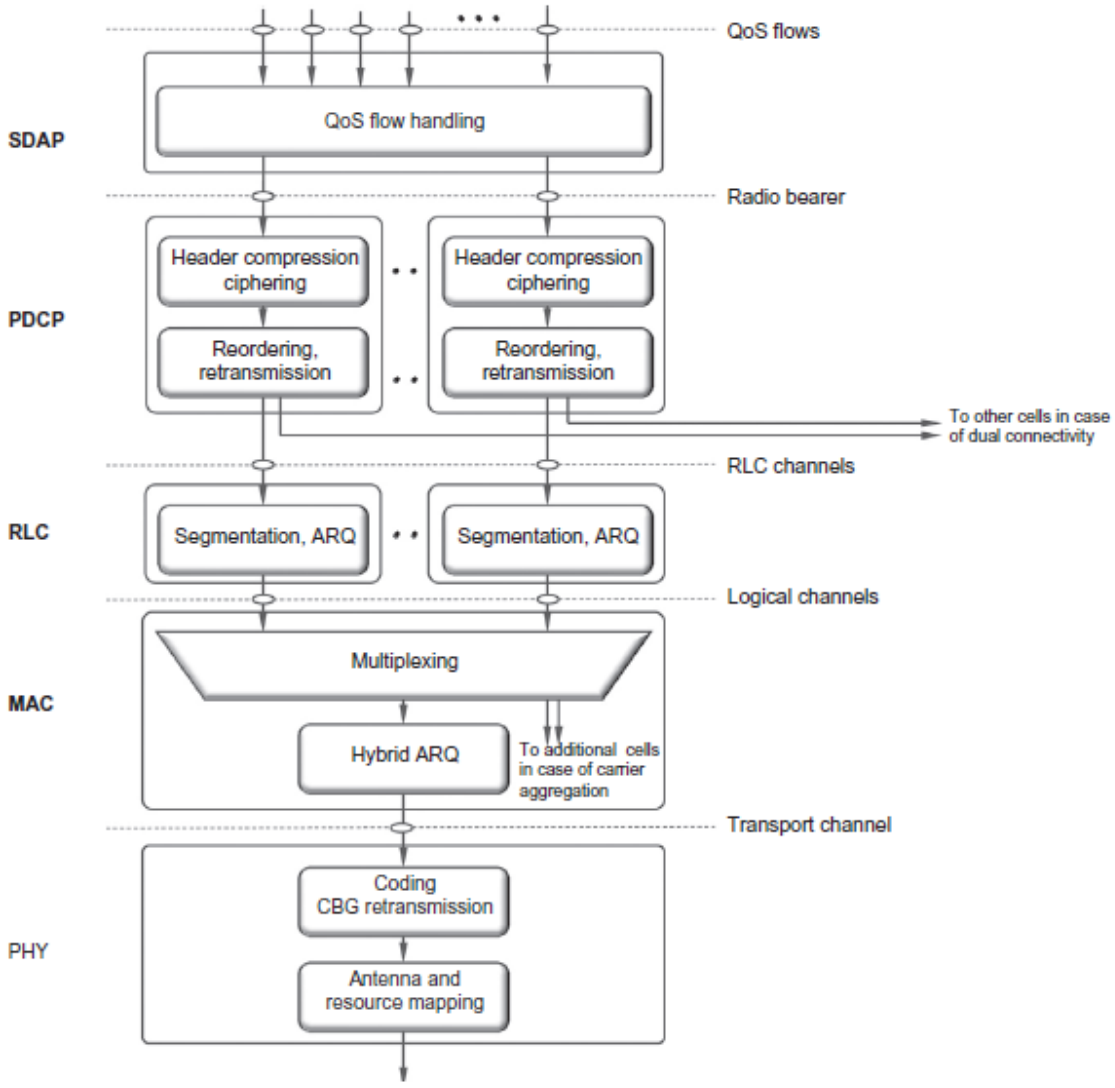


Figure 3.3: NR downlink user-plane protocol architecture

3.3 Detailed overview of L4S

As stated in Chapter 1, L4S uses ECN marking in IP headers at the base station based on very low queueing delays to enable early signals of congestion to the application as shown in Figure 3.4. The uplink procedure is indicated by the dotted lines, while the downlink procedure is indicated by the solid lines. In this context, the uplink procedure refers to the UE sending user data to an application server, such as during a live video stream when the UE

is uploading content. The downlink procedure refers to the UE receiving user data from the application server, for example, when the UE is viewing a live stream.

Although ECN is not a new concept for IP networks and was introduced in RFC 3168 [42], the L4S architecture redefines its usage to enable low latency while maintaining high throughput, through the use of scalable congestion control algorithms. This section provides a comprehensive overview of how L4S operates across the end-to-end data path, with emphasis on its integration in TCP and UDP-based systems within 5G networks.

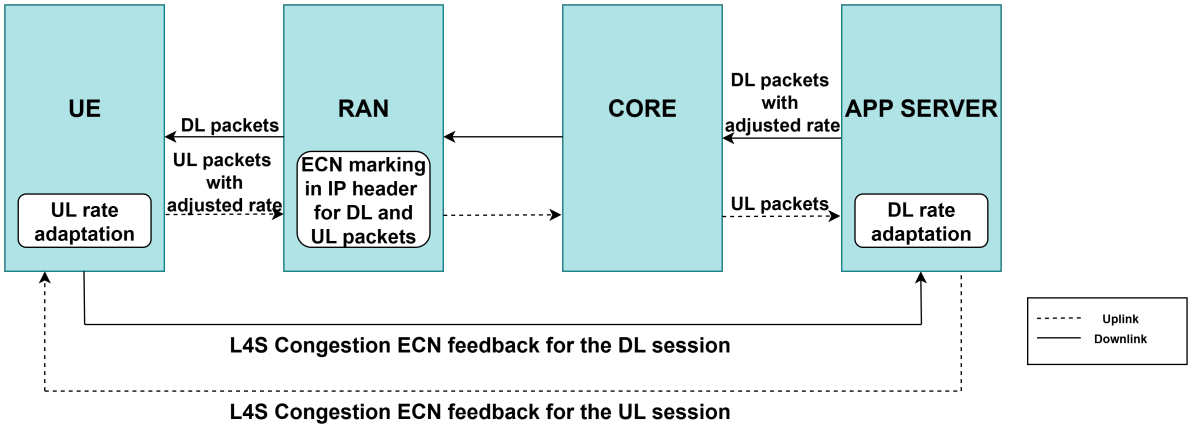


Figure 3.4: High Level Overview of L4S

3.3.1 Active Queue Management

Traditional queueing mechanisms in routers and switches, commonly known as tail drop (where packets at the end of the queue are dropped), only signal congestion when buffers overflow and packets are dropped. This reactive strategy leads to high latency and high losses. To address this, modern networks implement AQM techniques. AQM algorithms aim to proactively manage congestion by detecting early signs of buffer build-up and signaling congestion before queues are full. AQM techniques operate by measuring queue length or delay, and generate early congestion signals (either through packet losses or ECN marking), as opposed to waiting for tail drops. This early detection allows senders to slow down in a controlled and timely manner, resulting in lower latency and reduced packet loss. Example of AQM algorithms include Random Early Detection (RED), Controlled Delay (CoDel) or FlowQueue Controlled Delay (FQ CoDel).

3.3.2 Explicit Congestion Notification

ECN is a complementary mechanism to AQM algorithms that allows for marking packets at network queues to signal congestion.

ECN operates by using two bits in the Type of Service (ToS) field of the IP header, specifically designated for ECN functionality and has four possible values:

- **Not-ECN-Capable Transport (Not-ECT)**: Represented by the bits "00". This indicates that the packet is not capable of using ECN marks for congestion signaling. This is set

by the packet sender (that is; the application that generates the traffic).

- **ECN-Capable Transport, codepoint 0 (ECT(0))**: Represented by the bits "10". This is also set by the sender to indicate that the packet is ECN-capable and is used for classic ECN congestion control.
- **ECN-Capable Transport, codepoint 1 (ECT(1))**: Represented by the bits "01". Similar to ECT(0), but used specifically to identify L4S-capable traffic (RFC 9331). This is also set by the sender of the packets.
- **Congestion Experienced (CE)**: Represented by the bits "11". Set by a router or switch experiencing congestion (based on its AQM algorithm). Instead of dropping the packet, it is marked with CE and forwarded, signaling congestion to the receiver.

3.3.3 L4S procedure in 5G

This subsection provides an overview of the L4S procedure in 5G based on the reviewed standards and literature. In the L4S procedure, end-hosts (the application servers and UEs) mark packets as ECN-capable using ECT(1) (represented as 01) in the IP header. This indicates that the flow is L4S-aware and can react to congestion. The UPF, according to 3GPP TS 23 501 [8], in cooperation with the PCF and SMF, identifies and classifies such L4S flows using fields such as:

- Source and destination IP addresses
- Source and destination Port numbers
- Transport protocol (e.g., TCP or UDP)
- ECN field (e.g. match ECT(1) or CE codepoints)

Once identified, L4S traffic is associated with a dedicated QoS Flow, typically with a dynamic or predefined 5QI and marked for L4S-based congestion control.

According to 3GPP TS 23 501 [8], ECN marking for L4S can be performed either by the RAN or the UPF, depending on operator policy. In RAN-based marking, the SMF provides configuration indications to the gNodeB during QoS Flow setup, to enable the gNodeB to mark packets. For UPF-based marking, the UPF uses congestion reports from the gNodeB (sent through GTP-U header extensions) to infer congestion and apply CE marking accordingly. The marking behavior of the RAN and UPF is implementation-specific [8]. However, performing marking at the UPF introduces additional delay for downlink traffic and requires extra signaling between the RAN and the UPF [18].

At the network bottleneck (that is, the RAN), L4S relies on delay-sensitive AQM algorithms to mark packets with the CE codepoint (represented as 11).

In 5G networks, packets are typically queued at the RLC layer before being scheduled by the MAC layer, as mentioned in Section 3.2.3. However, encryption is applied at the PDCP layer, which limits visibility into packet contents at lower layers. As a result, ECN marking decisions can be based on congestion information calculated at the RLC layer, but the actual marking then applied at the PDCP layer. This separation introduces additional latency, as it can delay congestion feedback to the sender [4]. Although applying ECN marking directly at the RLC

layer would improve congestion feedback, doing so would require hardware modifications, which increases complexity and cost [4].

The choice of AQM mechanism is left to the implementer. IETF RFC 9332 [7] specifies the DualQ Coupled AQM as a recommended approach for L4S, although other implementations are possible. Since each UE maintains a separate queue at the gNodeB for each QoS flow, ECN marking decisions can be applied on a per-QoS flow basis. Each UE may have multiple QoS flows, each identified by different 5QIs. For example, if 5QI 3 is configured with L4S enabled, then all UEs with QoS flows assigned with 5QI 3 will have L4S applied to those flows.

As suggested in [4], the ECN marking in mobile networks can be based on the following criteria:

- If delay < Min Threshold: No marking
- If Min Threshold ≤ delay < Max Threshold: Probabilistic marking is done. The probability of marking the packets increases linearly as the delay increases. The marking probability is given by:

$$P = \frac{\text{Delay} - \text{Min Threshold}}{\text{Max Threshold} - \text{Min Threshold}} \tag{3.1}$$

- If delay ≥ Max Threshold: All packets are marked

According to [9], smoothing of the ECN markings by the bottleneck node is not required, as scalable congestion control algorithms at the sender are specifically designed to smoothen the ECN feedbacks.

At the receiver:

- For TCP, ECN CE marks are echoed back to the sender through TCP Acknowledgment (ACK) packets with the Accurate ECN flag in the TCP header [43].
- For UDP/RTP, CE marks are sent back to the sender by using RTCP feedback messages [44] or custom application messages.

3.3.4 Sender-Side Congestion Control

Upon receiving the CE feedback from the receiver, the sender performs scalable congestion control to adjust its sending rate. The IETF has defined a set of guidelines, that make a scalable congestion control algorithm L4S compliant which they have termed, the "Prague Congestion Control". This specifies the behavior for a congestion control algorithm to be compliant with L4S [9]. These requirements apply across transport protocols, including TCP, UDP, and other protocols such as Quick UDP Internet Connections (QUIC). Several implementations have been developed to meet these criteria, such as TCP Prague [9], UDP Prague [9], [10], and the L4S-compatible variant of SCReAM [11]. While they differ in protocol context, they share a common underlying mechanism designed to react promptly and proportionally to ECN feedback. This subsection discusses the main concept of the Prague Congestion Control as explained in [9].

Prague Congestion Control builds on the foundational principles of Data Center TCP (DCTCP), but has been extended to support the Internet. The paragraphs below describe the Prague algorithm.

ECN Feedback and Alpha Calculation: The algorithm maintains a moving average of alpha (α), which represents the fraction (frac) of acknowledged (ACKed) packets that were marked with ECN CE in their IP header over the previous virtual RTT (vRTT). That is;

$$\text{frac} = \frac{\text{ACKed packets with ECN marks}}{\text{Total ACKed packets}} \tag{3.2}$$

EWMA Update of Alpha: The value of α is updated once per virtual RTT(vRTT) using an Exponentially Weighted Moving Average (EWMA):

$$\alpha = (1-g)\,\alpha + g\,\text{frac} \tag{3.3}$$

where g is the EWMA gain factor or smoothening parameter (typically $g = \frac{1}{16}$) that dictates how much weight is given to the alpha calculation. A higher gain factor gives more weight to the new calculated "frac", thus making the EWMA more responsive to new changes, while a lower gain factor gives more weight to the previous calculated alpha, resulting in a smoother average. A gain factor of 1 makes alpha equal to frac. This smoothing prevents overreaction to short-lived congestion bursts and oscillations between rates.

The Virtual Round Trip Time (vRTT) is designed to address RTT bias and improve fairness when coexisting with classic congestion control algorithms such as TCP Reno. RTT bias occurs because the sending rate of a flow is inversely proportional to its RTT; thus flows with lower RTTs get ACKs faster and can increase their transmission rates more quickly, often at the expense of flows with higher RTTs. When a Prague-enabled application with a low RTT shares a link with classic congestion control algorithms like TCP Reno, it becomes a problem. In such cases, the Prague flow can dominate the available bandwidth if its RTT is very low. To address this, vRTT sets a lower bound for the RTT value used in certain Prague equations. By doing so, it prevents the Prague flow from gaining an unfair throughput advantage due to its low RTT. vRTT is derived from the smoothed RTT (sRTT) as:

$$\text{vRTT} = \max(\text{sRTT}, 25\text{ ms}) \tag{3.4}$$

This ensures that for RTTs higher than 25ms, vRTT is the sRTT, while for RTTs lower than 25ms, the vRTT used for the Prague calculations would be 25ms.

The sRTT denotes the smoothed RTT, which is the Exponential Weighted Moving Average (EWMA) of the RTT, calculated as [10];

$$\text{sRTT} = (1-\beta)\cdot\text{sRTT} + \beta\cdot\text{RTT} \tag{3.5}$$

where β is the EWMA gain factor, typically set to $\beta = 1/8$ [10].

Rate Reduction: When ACKs carrying ECN marks are received, Prague reduces its congestion window (cwnd) based on the computed congestion estimate, α (Equation 3.3) and then

enters the Congestion Window Reduced (CWR) state. The cwnd represents the maximum amount of data that can be in flight, i.e., sent into the network but not yet acknowledged by the receiver, and serves as a mechanism to control the sending rate in response to network congestion.

In the CWR state, the congestion window is reduced only once per virtual RTT (vRTT) upon receipt of ECN-marked ACKs. After this single reduction, no further decreases are applied for one round-trip, even if additional ECN-marked ACKs are received. This is done to prevent consecutive repeated window reductions.

The Congestion Window is decreased as:

$$\text{cwnd}_{\text{new}} = \text{cwnd} * \left(1 - \frac{\alpha}{2}\right) \tag{3.6}$$

A higher α corresponds to greater congestion, resulting in a more significant reduction of the cwnd.

Rate increase: Once the congestion window has been reduced, it is allowed to grow again even though it receives ACKed packets with ECN CE markings. It grows based on the number of bytes that have been ACKed without ECN CE markings as shown in the equation below:

$$\text{cwnd}_{\text{new}} = \text{cwnd} + \frac{(\text{acked} - \text{ece}) * \text{sRTT}^2 * MSS}{\text{cwnd} * \text{vRTT}^2} \tag{3.7}$$

Here, **acked** is the total acknowledged bytes, and **ece** is the subset of the acknowledged bytes that have ECN CE markings. Thus, **acked - ece** represents the subset of the ACKed bytes without ECN CE markings. MSS is the maximum segment size.

Pacing/Transmission rate: After calculation of the congestion window, prague uses packet pacing to send data smoothly into the network and to avoid sending data in bursts. For streaming applications with a single pacing rate, this rate is the target bitrate or the encoding rate.

The pacing/transmission rate is:

$$\text{pacing\_rate} = \frac{\text{cwnd}}{\text{sRTT}} \tag{3.8}$$

End-to-End L4S Workflow

The overall end-to-end workflow of L4S for downlink is summarized in Figure 3.5 and described in the steps below:

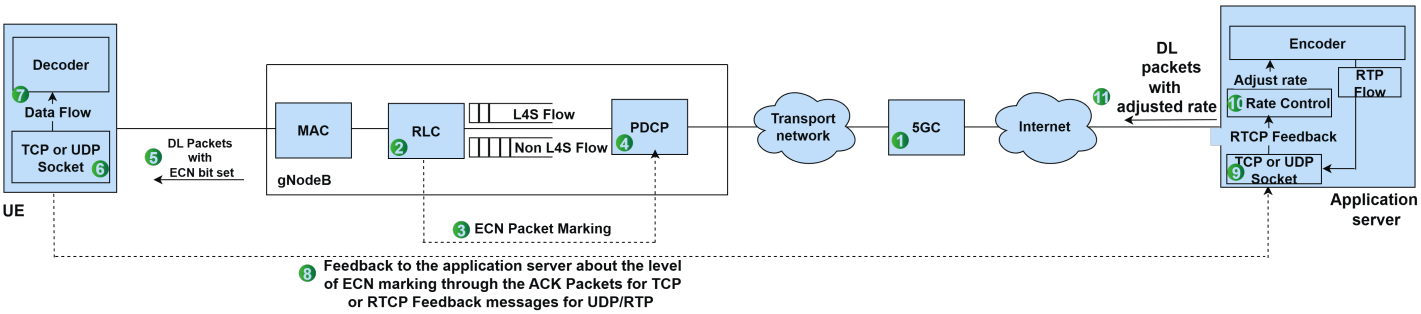


Figure 3.5: L4S Procedure - Downlink

1. **Traffic Classification:** The PCF or the SMF identifies L4S traffic (e.g. through ECT(1), Source and Destination IP/port numbers) and steers it into the L4S QoS flow.

2-3. **Queue Monitoring and marking logic:** The gNodeB monitors queue delay and applies marking thresholds. CE probability increases linearly with delay between the minimum threshold and the maximum threshold; all packets are marked if delay exceeds the maximum threshold.

4. **Marking Location:** ECN CE marking is done at the PDCP layer.

5. **Packet delivery to UE:** The marked packets are delivered to the UE with the CE bits set in the IP header of the packets.

6- 8. **Decoder and Feedback Channel:** The packets are decoded at the receiver and the ECN CE marks are fed back to the sender.

- TCP: CE marks are echoed in the TCP header of ACK packets.
- UDP: RTCP feedback or custom application messages are used.

9. **Congestion Response:**

- TCP: In TCP-based systems, congestion control is integrated into the operating system. Hence, the transport layer handles the pacing of the packets using the TCP Prague algorithm.
- UDP: For real-time applications running over UDP/RTP, congestion control is handled at the application layer. The application adapts the rate of the packets based on the CE information (e.g., in SCReAM or UDP Prague). This requires adaptive encoders and feedback mechanisms such as RTCP (as mentioned in step 6-8).

10. **Encoder Adaptation:** The encoder dynamically adjusts its bitrate based on the feedback and the rate calculations.

11. **Next Packet Transmission:** The server sends the next set of packets with the adjusted bitrate.

In the uplink direction, the L4S procedure differs slightly from the downlink. Although comprehensive information about the uplink process for L4S is not yet available in literature, the

general approach is outlined below. A guideline is also presented in 3GPP TR 23 700-60 [45].

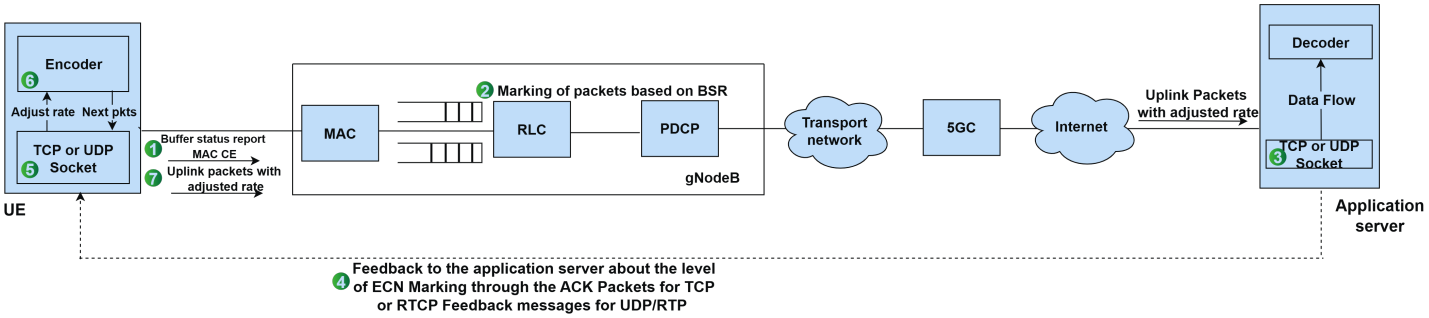


Figure 3.6: L4S Procedure - Uplink

As shown in Figure 3.6, in the uplink, the UE acts as the encoder, while the application server acts as the decoder. Queue monitoring occurs at the UE, and congestion feedback flows in the reverse path compared to the downlink. The uplink procedure can be summarized as follows:

1. **Queue Status Reporting:** The UE monitors its uplink buffer and sends a Buffer Status Report (BSR) MAC CE to the gNodeB, which indicates the level of queueing at the UE RLC buffers [45].

2. **Marking Decision:** The gNodeB uses the reported buffer information to estimate the queueing delay and applies ECN CE marks.

3 - 4. **Feedback to UE:** The application server receives the ECN CE-marked packets, extracts the marking information, and sends this ECN CE information to the UE.

5. **Congestion Response:** For TCP-based applications, the UE computes a new pacing rate using the TCP Prague algorithm. For UDP-based applications, the feedback is forwarded to the media encoder to adjust the encoding bitrate accordingly (e.g. as in SCReAM or UDP Prague).

6. **Rate Adaptation:** The encoder updates its encoding bitrate based on the feedback to match the available uplink capacity.

7. **Packet Transmission:** The UE transmits the next set of packets with the adjusted bitrate to the gNodeB.

3.4 Detailed overview of ANBR

As mentioned in Chapter 1, ANBR is a 3GPP-standardized mechanism that enables the access network (e.g. enodeB or gNodeB) to recommend an optimal bitrate for media transmission, allowing real-time adaptation to radio conditions. The high level procedure is shown in Figure 3.7. The uplink procedure is indicated by the dotted lines, while the downlink procedure is indicated by the solid lines.

The ANBR mechanism is applicable in both uplink and downlink media sessions.

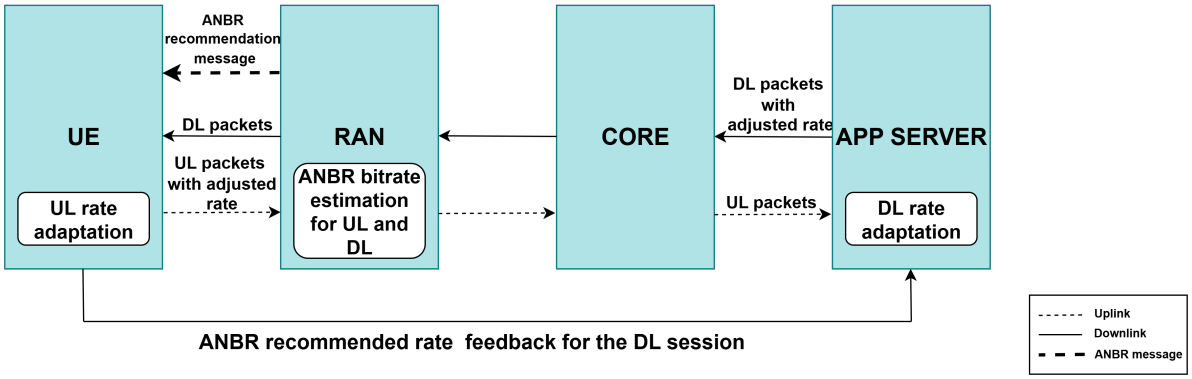


Figure 3.7: High level overview of ANBR

3.4.1 ANBR procedure in 5G

ANBR involves three primary components, which would be detailed in subsequent subsections. These components are:

1. **Network (RAN)** - The gNodeB calculates the optimal physical layer bitrate for the application and sends this to the UE through MAC CEs.
2. **User Equipment (UE)** - Receives the bitrate recommendation and forwards it to the application layer.
3. **Application Server** - Uses the recommendation to decide on the appropriate bitrate to use for the next packets it sends.

Bitrate Estimation and Signaling

Bitrate estimation for ANBR is vendor-specific; however, 3GPP TS 26.114 [12], 3GPP TS 36 321 [28] and 3GPP TS 38 321 [29] provides general guidelines. The gNodeB or eNodeB estimates the optimal bitrate over a defined averaging window, and transmits this information using the Recommended Bit Rate MAC CE to the UE's MAC entity. The ANBR message suggests a physical-layer bitrate for a specific logical channel and direction (uplink or downlink). To map this physical-layer recommendation to the application bitrate, the UE and the gNodeB must account for any protocol overheads, including IP, UDP/TCP, RTP, RLC, PDCP, and MAC headers. In addition, any header compression techniques, such as Robust Header Compression (ROHC), should be considered when estimating the effective bitrate of the application [12].

Bitrate Handling at the UE

Upon receiving the MAC CE, the UE will identify the logical channel and direction (uplink or downlink) and take one of two actions;

- **Uplink:** The UE application directly adapts its sending rate according to the recommendation.

- **Downlink:** The UE forwards the recommendation to the application server. The method of forwarding is implementation-specific and not standardized.

The application then uses this information to determine the appropriate bitrate for subsequent packets.

Bitrate Query by the UE

The UE may also initiate a bitrate recommendation query using the Access Network Bitrate Recommendation Query (ANBRQ) MAC CE. In this case, the UE sends a MAC CE to the gNodeB containing the requested bitrate. If the query timer that is set by the gNodeB to prevent a bitrate query over a certain time interval (bitRateQueryProhibitTimer) is not active, the UE sends this ANBRQ MAC CE to the gNodeB. Upon receiving this, the gNodeB responds with an updated recommended bitrate MAC CE. This UE-initiated flow enables the UE to request an increase in bitrate at intervals, especially when the application experiences buffer underruns.

3.4.2 Recommended Bitrate MAC CE format

Defined in 3GPP TS 38.321 [29], the Recommended Bit Rate MAC CE is a fixed-size control element consisting of two octets (16 bits). The subheader of the recommended bitrate MAC CE is 8 bits. The Recommended Bit Rate MAC CE fields are defined in Table 3.1 and shown in Figure 3.8 [28]:

Table 3.1: Recommended Bit Rate MAC CE Format

Field	Description
LCID (6 bits)	Logical channel identifier for bitrate recommendation
UL/DL (1 bit)	Direction indicator: 0 = Downlink, 1 = Uplink
Bit Rate (6 bits)	Index to bitrate table with bitrate values. This is specified in a predefined table (Table 6.1.3.20-1 in 3GPP TS 38.321). For example, Index 54 refers to the NR recommended bitrate 7000kbps.
X (1 bit)	Bit rate multiplier flag (used if bitRateMultiplier as specified in 3GPP TS 38 331 [46] is configured; when set to 1, the value of the bitrate index is multiplied by the bitRateMultiplier).
R (2 bits)	Reserved bit (set to 0)

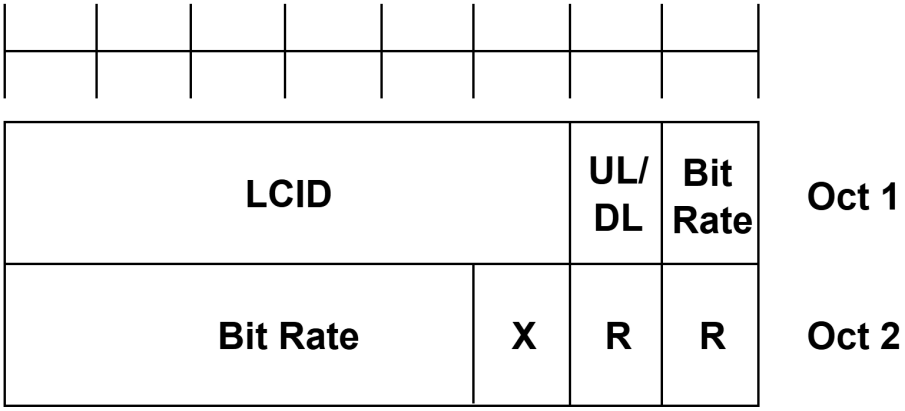


Figure 3.8: Recommended bitrate MAC CE as specified in 3GPP TS 38 321

Summary of ANBR Procedure

The ANBR procedure for downlink and uplink are summarized in Figure 3.9 and the procedure below.

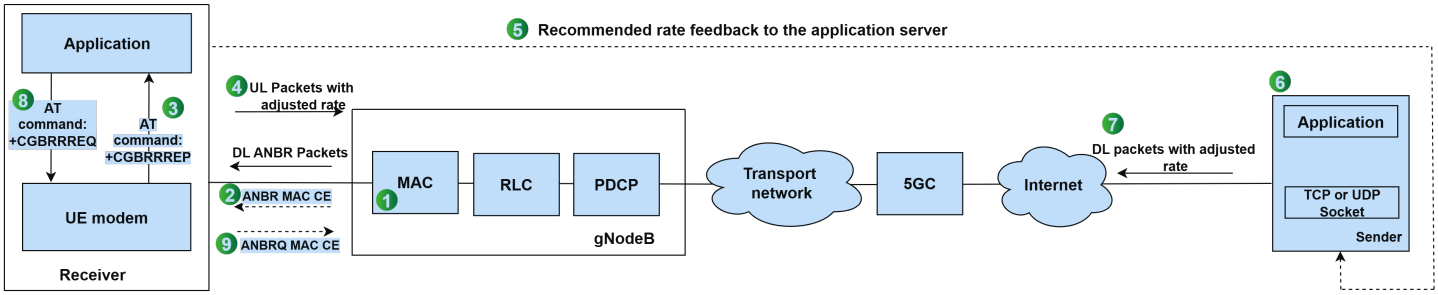


Figure 3.9: ANBR Uplink and Downlink Procedure

1. **Bitrate Estimation:** The gNodeB estimates the downlink bitrate.

2. **ANBR MAC CE Transmission:** The gNodeB communicates the recommended bitrate to the UE through the MAC CE.

3. **UE Modem to Application:** The UE modem delivers the recommended bitrate to the application (for example; using AT commands +CGBRRREP) as specified in 3GPP TS 26 510 [13] and 3GPP TS 27 007 [47]).

4. **UL application usage:** For uplink, the application adjusts the encoding rate based on the recommended bitrate.

5. **Feedback to the server for DL:** The UE sends the recommended bitrate to the server, by using a custom RTCP feedback (e.g., in VoLTE or VoNR) [12] or a custom application feedback.

6. **DL application usage:** The server adjusts the encoding rate based on the recommended bitrate.

7. **Adaptive Transmission:** DL Packets are transmitted according to the updated bitrate.

8. **Bitrate Query:** The application may query the modem for bitrate recommendations using +CGBRRREQ Attention commands [13], [47].

9. **ANBR Request to gNodeB:** The UE sends a MAC CE containing the ANBR Query to the gNodeB to request for an updated bitrate recommendation.

3.5 Comparative analysis of L4S and ANBR

This section presents a comparison of L4S and ANBR mechanisms. Additionally, a hybrid method is discussed, showing how L4S and ANBR can be combined to achieve better rate adaptation and media quality.

Comparison

Table 3.2 presents the key differences between L4S and ANBR.

Table 3.2: Comparison of L4S and ANBR

Criteria	L4S	ANBR
Rate Calculations	Specifically calculated using queueing delay and aimed at reducing queueing delay.	Although, there is no standardized calculations yet for the recommended bitrate, it is based on the optimal bitrate that the UE can achieve over a specific interval. It does not specifically target queueing delay, hence packets can get buffered up potentially resulting in higher latency.
Congestion Signalling Mechanism	Uses ECN bits in the IP header to signal congestion	Directly uses the bitrate to signal congestion levels. Higher bitrate indicates good network conditions and low congestion levels while lower bitrate indicates poor network conditions and high congestion levels.
Rate Increase After Congestion	The application relies on ECN CE marks, thus it gradually increases its rate based on the number of received/acknowledged packets without CE marks.	The gNodeB provides an explicit new bitrate recommendation, enabling the application to increase its rate faster.

Criteria	L4S	ANBR
ECN CE Thresholds and Window interval	Uses queue thresholds to control the level of queueing and reduce delays.	Uses a window interval to calculate and recommend the optimum bitrate.
Uplink Operation	Relies on a feedback loop from the server to the UE.	No feedback loop is required for the uplink. The UE adapts its rate directly based on the recommended bitrate.
Rate Calculation Location	The rate is calculated by edge devices, such as the UE (for uplink) and the remote application server (for downlink)	The rate is calculated by the gNodeB. Thus, packet overheads need to be accounted for.
Congestion Signaling Overhead	Consists of ECN bits in the IP header and feedback signaling overhead (only 2 ECN bits in the IP header + 2 bits in the RTCP feedback) for both downlink and uplink.	Involves a 24-bit MAC CE for bitrate recommendation from the gnodeB to the UE plus feedback overhead in the downlink. For the uplink, it uses a 24-bit MAC CE bitrate recommendation from the gnodeB to the UE which the UE directly uses to adapt its rate.
Implementation Approach	Based on standardized ECN marking behavior and scalable congestion control mechanisms.	Lacks standardized procedures for how the gnodeB should calculate the bitrate and how the application server should use the recommendation; implementation is application-specific.

Advantages and Disadvantages

Table 3.3 presents the strengths and limitations of L4S and ANBR.

Table 3.3: Advantages and Disadvantages of L4S and ANBR

Mechanism	Advantages	Disadvantages
L4S	• Specifically targets low queueing delay, thereby reducing end-to-end latency. • Convenient threshold tuning for QoS by assigning different queueing thresholds to different 5QIs. • Low control signaling overhead. • Uses the IP header at the IP layer, making it applicable end-to-end across both mobile and fixed networks, including the core. • Mature ecosystem; standardized and is gradually getting adopted.	• Requires an ECN-capable path; intermediate nodes must preserve ECN bits. • Slower bitrate increase post congestion potentially resulting in lower throughput. • For uplink traffic, congestion signals must reach the server before feedback is returned to the UE.
ANBR	• Higher bitrate increase resulting in higher throughput. • Integration is simplified as only RAN-level support is required in the mobile network, with no risk of other network nodes modifying the MAC CE congestion signaling. • UE can apply recommended bitrate directly for uplink traffic.	• Does not directly control queueing delay; can cause higher latency. • Requires additional MAC layer control signaling, increasing bandwidth usage. • Limited standardization on bitrate calculation and how the application server should apply the rates. • It is more complex to determine a level of queueing delay for the different QoS flows. • Mainly designed for use within the Access Network which limits its applicability and may hinder broader adoption across the telecoms ecosystem.

3.6 Combined use of L4S and ANBR

Combining the strengths of L4S and ANBR has the potential to improve throughput while maintaining low latency. L4S is specifically designed to reduce queueing delay and is conservative in its rate increase after congestion, gradually increasing its sending since it relies on ECN-based congestion feedback. ANBR, on the other hand, does not explicitly target queueing delay but can increase the bitrate more aggressively given that it relies on the RAN feedback.

Given these differences, a hybrid approach could be beneficial: using L4S for congestion detection and rate reduction, while using ANBR to signal bitrate increases after congestion subsides. This combination would allow the application to preserve L4S's low latency benefits during congestion while also benefiting from ANBR's faster rate recovery to improve overall throughput.

4

Simulation and Test Lab Setup

This chapter presents the simulation framework and the test lab evaluations conducted as part of this project. It describes the modeling approach adopted for simulating the 5G network, the experimental scenarios, and the specific evaluations performed in both the simulated and test lab environments.

4.1 Simulation modelling

The simulation environment was developed using ns-3, which is a widely adopted open-source network simulator designed for research and educational purposes. Written in C++, ns-3 offers a modular architecture that enables detailed modeling of network protocols across various network layers. However, its core modules do not natively support advanced features specific to 5G NR protocols.

To enable 5G NR network simulations, the 5G-LENA [48] module was integrated into the ns-3 framework. 5G-LENA is an open-source, pluggable extension specifically developed to support 3GPP-compliant simulations for 5G NR. It enables comprehensive end-to-end network modeling, from the application layer to the physical layer. This makes it suitable for evaluating bitrate adaptation techniques within a mobile network environment.

4.1.1 Network topology and simulation parameters

The simulated network is based on the 5G Urban Macrocell (UMa) deployment scenario, as defined in 3GPP TR 38.901 [49]. The topology, illustrated in Figure 4.1, consists of a single gNB, one UPF, multiple UEs, and two remote application servers: one dedicated to L4S/ANBR traffic, representing AR/VR use cases, and another generating background traffic to congest the network. The aim of the study is to understand the core rate adaptation mechanisms of ANBR and L4S. Hence, UE mobility was excluded from the simulation study. While mobility is also an important factor influencing rate adaptation, its impact is left for future research where it can be studied in greater depth within the same framework. The simulation focuses on downlink traffic, as most users primarily download data rather than upload.

With respect to the links between the UPF and the remote application servers, as well as between the UPF and the gNodeB, high-capacity links of 100 Gbps were configured. This

ensures that the RAN is the only bottleneck in the network where congestion can occur.

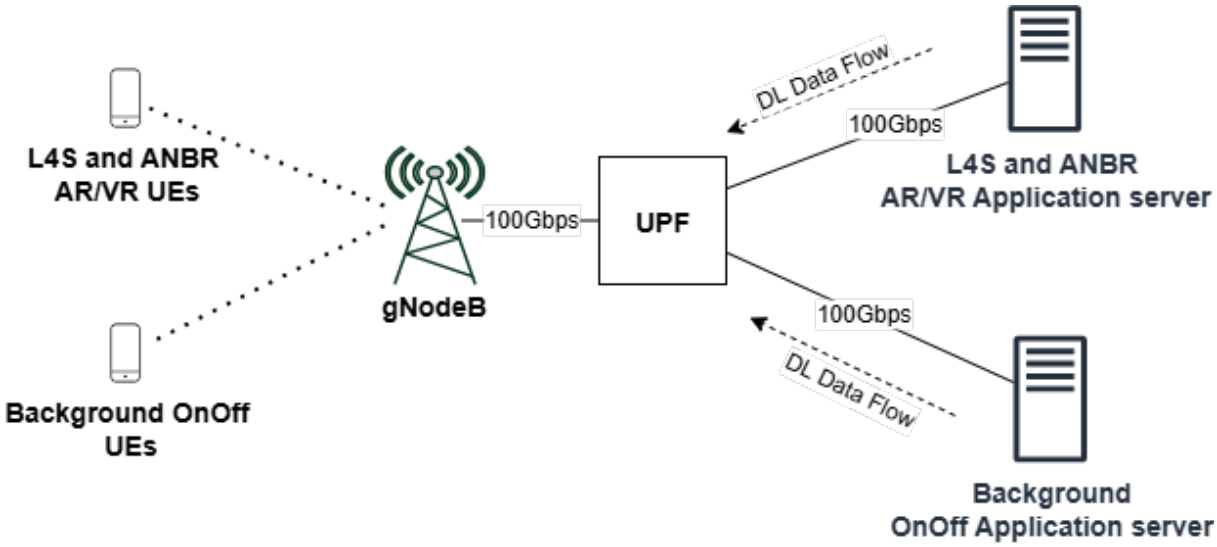


Figure 4.1: Simulation network topology

Regarding the gNodeB configuration, a carrier frequency of 700 MHz with a bandwidth of 15 MHz was used for the gNodeB. Propagation and channel modeling were implemented in accordance with 3GPP TR 38.901 [49]. The buffer size was configured to 1GigaByte, with an additional 15MegaBytes buffer also tested to evaluate the impact of packet loss. The Proportional Fair scheduler was selected for resource allocation, as it is widely used in mobile networks and balances throughput and fairness effectively. The gNB transmission power was set to 46 dBm, and UEs were positioned randomly within the cell coverage area.

Each simulation was run for a duration of 30 seconds. This time frame was chosen because the primary focus of the simulation is to assess and compare the performance of L4S and ANBR. A stable transmission rate was reached within this period in the simulations, which allowed the behavior of each mechanism to be clearly observed. Furthermore, five independent simulations were run, with the random parameters (including the background traffic) changed for each run. The averages of these values were then derived and recorded. The small number of independent simulations was chosen because the main goal was to understand the behavior of the systems rather than obtain a highly precise estimate.

A summary of the key simulation parameters is provided in Table 4.1.

Table 4.1: Simulation Parameters

Parameter	Value
Bandwidth	15 MHz
Frequency Band	700 MHz
gNodeB Antenna	1T1R
UE Antenna	1T1R
Buffer Size	1 GigaByte, 15MegaBytes
gNodeB Antenna Height	25 m
UE Antenna Height	1.5 m
Window Interval	40 ms, 100 ms, 200 ms
L4S Threshold Range	5-7 ms, 7-15 ms, 10-20 ms, 15-20 ms, 15-30 ms, 30-50 ms, 60-100ms
gNB Transmit Power	46 dBm
UE Transmit Power	23 dBm
Propagation Model	3GPP Urban Macro 3GPP TR 38.901
Scheduler	Proportional Fair (no priority)
Traffic Models	L4S/ANBR - 3GPP Generic Video (AR/VR) 3GPP TR 38.838 Background traffic - ns3 OnOffApplication
RLC Mode	Unacknowledged Mode
Numerology	1
Simulation runtime	30 seconds
Multiple Access	OFDMA

4.1.2 Traffic model and testing scenarios

As stated in Section 4.1.1, two types of traffic were generated in the simulation environment:

1. **L4S and ANBR traffic**: This was modelled using the 3GPP Generic Video Model for AR and VR flows, which was inspired by [50] and is defined in 3GPP TR 38.838 [51]. For both the L4S and ANBR traffic models, a single downlink AR/VR video stream was simulated, with a baseline and maximum data rate of 30Mbps and a frame rate of 60fps. In this work, it is referred to as AR/VR traffic because the single downlink stream for both AR and VR follows the same model as defined in 3GPP TR 38.838. The reason for using AR and VR traffic is to represent a use case with low latency and high throughput requirements.

2. **Background traffic**: Background traffic was modeled using ns-3's OnOffApplication to simulate varying levels of network capacity and congestion. The OnTime parameter represents periods of user activity that generate network load, while the OffTime parameter models idle periods, such as user reading or thinking time between the periods of the user's active times.

Two background traffic scenarios were configured:

The first scenario models a low-congestion scenario which simulates the background user with relatively light and bursty activity, resulting in lower overall congestion. The average background throughput was 6.71 Mbps in this case. In this scenario, the On-Time was drawn from a Pareto distribution with scale parameter $x_m = 0.133$ and shape parameter $\alpha = 1.5$, while the OffTime followed a log-normal distribution with parameters mu, $\mu = 0$ and sigma, $\sigma = 0.8$. This reflects heavy-tailed web-browsing behavior, where the user has short bursts of activity followed by longer idle periods [52].

The mean of a Pareto-distributed variable $X \sim \text{Pareto}(\alpha, x_m)$ is:

$$\mathbb{E}[X] = \begin{cases} \dfrac{\alpha\, x_m}{\alpha - 1}, & \alpha > 1, \\ \text{undefined}, & \alpha \leq 1 \end{cases} \tag{4.1}$$

and its variance is:

$$\text{Var}(X) = \begin{cases} \dfrac{\alpha\, x_m^2}{(\alpha-1)^2(\alpha-2)}, & \alpha > 2, \\ \infty, & 1 < \alpha \leq 2 \end{cases} \tag{4.2}$$

For $\alpha = 1.5$ and $x_m = 0.133$, the mean OnTime is:

$$\mathbb{E}[X] = \frac{1.5 \times 0.133}{1.5 - 1} = 0.399 \text{ seconds.} \tag{4.3}$$

Since $1 < \alpha \leq 2$, the variance is infinite, thus occasional long bursts of activity can occur.

For the OffTime, a log-normal distribution $X \sim \text{LogNormal}(\mu, \sigma^2)$ is used. It is important to note that the mean parameter μ of a log-normal distribution is not the mean of the original data, but rather the mean of the natural logarithm of the data. The mean and variance of the original data are:

$$\mathbb{E}[X] = e^{\mu + \frac{\sigma^2}{2}}, \tag{4.4}$$

$$\text{Var}(X) = \left(e^{\sigma^2} - 1\right) e^{2\mu + \sigma^2}. \tag{4.5}$$

With $\mu = 0$ and $\sigma = 0.8$, the mean OffTime is:

$$\mathbb{E}[X] = e^{0.32} \approx 1.377 \text{ seconds,} \tag{4.6}$$

$$\text{Var}(X) = (e^{0.64} - 1)\, e^{0.64} \approx 1.699. \tag{4.7}$$

This yields an average idle time of 1.38 seconds.

In the second scenario, a high-congestion profile was used to simulate background traffic from the user but with more continuous and intense activity, resulting in a higher congested scenario. Thus, in this case, the average background throughput increased to 11.29 Mbps. Here, both OnTime and OffTime were modeled using exponential distributions, with means of 0.06 s and 0.03 s, respectively. This results in shorter idle periods and more continuous traffic, creating a more congested environment.

4.1.3 L4S simulation

The L4S simulation was implemented using the UDP Prague congestion control algorithm, which is based on the official open-source reference code developed by the L4S Team [10]. As part of this study, the code was then adapted for use within the ns-3 environment.

UDP Prague was selected as the L4S algorithm for this work because most real-time low-latency applications, such as real-time AR/VR and cloud gaming, are built on top of UDP transport protocols rather than TCP. Additionally, the algorithm adheres to the core principles outlined in the IETF Prague requirements for L4S [9].

The key components of the L4S simulation in ns-3 are described below.

Remote Application server: A custom application was used to generate AR/VR traffic based on the 3GPP Generic Video Model as described in earlier sections. The application integrates the UDP Prague congestion control mechanism for L4S adjustments.

gNodeB: The RLC layer was configured to operate in Unacknowledged Mode (UM) and perform ECN marking based on the head-of-line (HoL) queueing delay. As discussed in Chapter 2 and 3, according to literature, the RLC sends the marking information to the PDCP layer, which sets the ECN bits in the IP header before the packet gets encrypted. In this simulation, however, marking was applied at the RLC layer as encryption of data in the PDCP layer is not currently supported in ns-3. Furthermore, ns-3 packet tags, rather than IP header bits, were used to mark the packets. Packet tags are metadata attached to packets and do not contribute to the size of the packet. Since a major focus of the simulation was to analyze the rate adaptation, this approach was considered suitable and sufficient.

When the HoL delay exceeded a predefined threshold, packets were probabilistically tagged with the Congestion Experienced (CE) codepoint, as defined in Section 3.3.3 (Equation 3.1). Multiple delay thresholds were evaluated, including 5–7 ms, 7–15 ms, 10–20 ms, 15–20 ms, 15–30 ms, 30–50 ms, and 60–100 ms. These values were chosen to span a range of low and high thresholds in order to evaluate the case where a low latency is desired (by the low threshold 5-7ms) and also to analyze the impact of the thresholds on latency and throughput when they are increased .

Receiver / UE: The UE receives the incoming packets, extracts any ECN markings, and gets the original timestamps which was inserted at the server (the timestamp is the time the packet was sent by the server). It then forwards both the ECN marking information and the corresponding timestamps to the remote application server and the UDP Prague congestion control module. The server uses the ECN markings to compute the congestion level parameter, α, and use the timestamps to estimate the RTT which is calculated as the difference between the original timestamp and the current time.

UDP Prague Rate Adaptation Module: As mentioned previously, the L4S algorithm was adapted from the reference UDP Prague implementation and integrated into the ns-3 framework. The UDP Prague module emulates a continuous streaming application with a single pacing rate that dynamically adjusts its encoding rate based on network feedback.

The main components of the UDP Prague algorithm align with Section 3.3.4 and the L4S requirements defined in [9]. Although UDP Prague operates over UDP, it adopts the congestion window increase and decrease mechanism (traditionally used in TCP congestion control) to

guide the rate adaptation for L4S traffic. Unlike TCP, which uses the congestion window to limit the number of packets it sends into the network until acknowledgements are received, UDP continues sending packets regardless of what has been received by the client. Therefore, in UDP Prague, the congestion window is conceptual and serves as a guide for calculating the target bitrate rather than imposing a strict in-flight packet limit.

The key components of the algorithm are as follows:

- The sender maintains a smoothed EWMA estimate of the congestion level, denoted as α, as defined in Equation 3.3.
- When packets marked with the Congestion Experienced (CE) codepoint are detected, the sender reduces the congestion window (cwnd) in proportion to the estimated congestion level α . The reduction follows the formula defined in Equation 3.6.
- Following a congestion event and the corresponding reduction in the congestion window, the window is allowed to grow again after one RTT has passed. However, the growth is proportional to the number of newly received packets that are not marked with ECN-CE. This is adapted from Equation 3.7.
- Finally, the target bitrate is updated based on the calculated congestion window and RTT as given in Equation 3.8.

4.1.4 ANBR simulation

The ANBR mechanism is simulated by monitoring the downlink data delivered to each UE over a defined window and providing the bitrate feedback to the UE, which then informs the server of the rate.

Key Components

- **Bitrate Estimation:** The gNodeB tracks the transmission capability of the gNodeB for each UE by calculating the amount of bits transmitted to each UE over a fixed time window. In the experiments, packet headers were accounted for: the sizes of the UDP, IP, PDCP, RLC, and MAC headers in ns-3 were subtracted from each packet when calculating the recommended bitrate. ROHC is not present in ns3 and was not included since it is not the focus of this work. The window values chosen for the experiment were 40ms, 100ms and 200ms. To avoid very small windows, a 40ms window was chosen as the smallest value in this simulation. This was chosen because 40ms provides enough samples for a stable bitrate estimate and to prevent oscillations between the rate, while still enabling reasonably fast adaptation by the server.
- **Rate Adaptation:** This recommended bitrate is sent to the UE which is sent to the server, which then adjusts its sending rate accordingly for the next packets it sends.

4.1.5 Hybrid simulation (L4S and ANBR)

A hybrid approach was also evaluated to combine the strengths of both L4S and ANBR mechanisms. As described in Chapter 3, the goal of this method is to improve throughput while maintaining low latency by switching between L4S and ANBR based on network conditions.

In this approach, L4S is used during periods of congestion. When ECN marks are detected, the rate calculated by L4S is used to ensure prompt reaction to congestion signals. When no ECN marks are detected, the algorithm selects the higher of the rates calculated by L4S and ANBR. This ensures that once congestion has been addressed through L4S, the sending rate can subsequently increase faster, while still remaining responsive to L4S ECN signals when congestion is detected. For this evaluation, an ANBR windowing value of 40 ms was used, alongside the L4S queueing delay thresholds described in the L4S section (Section 4.1.3), excluding the 60-100 ms threshold. The 60-100 ms threshold was excluded because it rarely produces ECN marks, and the rate reduction is mainly determined by the RTT, as will be shown in Chapter 5. In this case, the hybrid method would rely mostly on ANBR's rate, which would override L4S's ECN-driven rate control.

Figure 4.2 presents a flowchart summarizing the simulation processes for L4S, ANBR, and the hybrid adaptation approach.

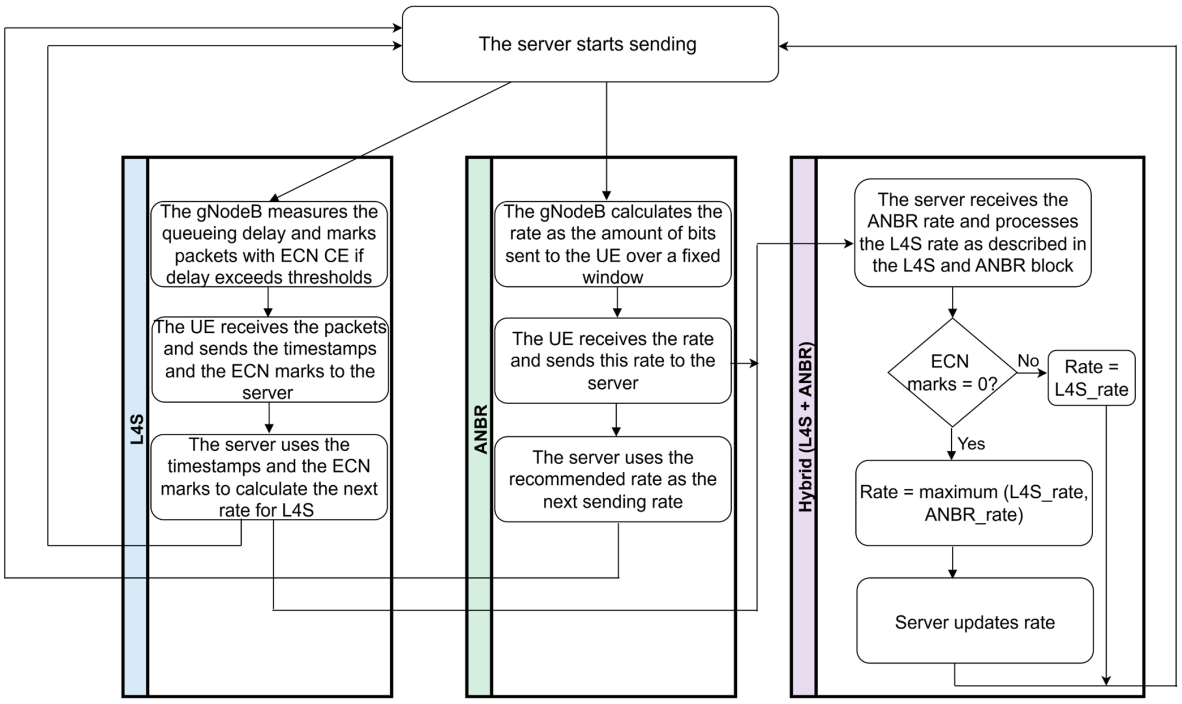


Figure 4.2: Flowchart showing the simulation processes for L4S, ANBR, and the hybrid method

4.2 Testbed

This section describes the testbed setup and the scenarios used to evaluate L4S in a real-world environment. Due to hardware limitations, ANBR could not be included in this evaluation. The assessment of L4S was carried out using the KPN 5G SA test network.

4.2.1 L4S testbed setup

The test network infrastructure is shown in Figure 4.3. While the actual intermediate network nodes are more complex than what is shown, their details have been abstracted, as they are not the primary focus of this study. The level of detail illustrated in the figure is enough to convey the main components of the test network. The session under test was a downlink session.

The gNodeB, operating at a carrier frequency of 3.5 GHz, consists of an Ericsson's Baseband Unit 6630 and Radio Unit 4408. The node was configured with a 20 MHz bandwidth, a transmit power of 1 Watt, and a one transmit, one receive (1T1R) antenna configuration. This setup was chosen to reduce capacity, as only two UEs were used during testing. The reduced capacity was aimed at causing congestion.

On the client side, two laptops were used. The first laptop, used for L4S testing, ran Ubuntu 24.04.2 LTS operating system (OS) with Linux kernel version 6.11.0-29-generic. The second laptop, used for generating background traffic, operated within the Windows Subsystem for Linux (WSL) environment,specifically WSL2,running Ubuntu 22.04.5 LTS OS with Linux kernel version 5.15.167.4-microsoft-standard-WSL2.

Network connectivity for both laptops was established using USB tethering to two Google Pixel 9 smartphones, each running Android 15 with support for 5G SA. To remove external interference and ensure consistent radio conditions throughout the experiments, the smartphones were placed inside a shielded box together with the antenna, as shown in Figure 4.4.

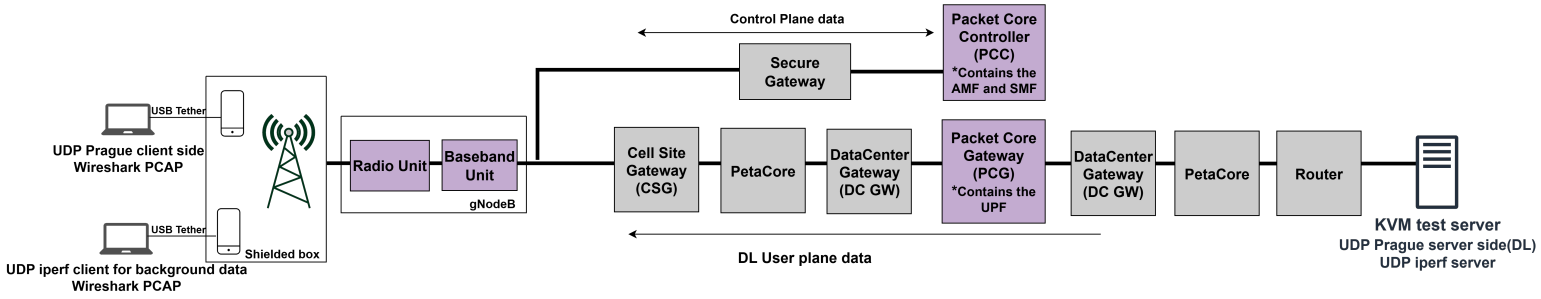


Figure 4.3: Test Lab Setup

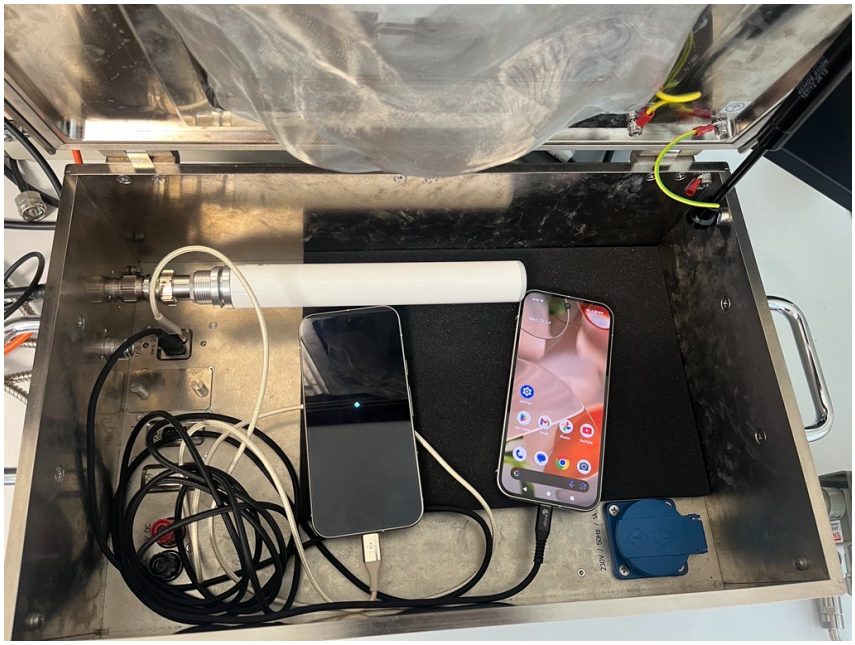


Figure 4.4: The shielded box showing the antenna (white object) and the two Google Pixel 9 smartphones

On the server side, a Kernel-based Virtual Machine (KVM) server was running a virtual machine with Debian GNU/Linux 12 operating system with kernel version Linux 6.1.0-23-amd64 and was directly connected to the KPN network. As shown in Figure 4.3, this server is connected to a router which is then connected to the KPN PetaCore router, which forwards the packets to the Data Center Gateway (DC-GW). The DC-GW interfaces with the virtualized core network functions, including the Packet Core Gateway (PCG), which is a cloud native user plane function responsible for handling user plane traffic. Control-plane data is routed through the Secure Gateway to the Packet Core Controller (PCC), which is a cloud native control plane signaling processing function that hosts the AMF and the SMF. These network functions (PCG and PCC) are part of Ericsson's core equipment, which combines multiple core network functions into compact units. KPN uses Ericsson's core equipment to implement its virtualized core network.

The Cell Site Gateway (CSG), which includes devices from the Huawei ATN series, serves as a transport node for traffic between the core and access networks. It interfaces with the Baseband Unit (BBU), which is connected to the radio unit using fiber optic cables. The radio unit is connected to the shielded box using a coaxial cable. Inside the box (Figure 4.4), an antenna delivers the downlink signal to the smartphones, which in turn delivers the signal to the laptops through USB cables.

An Access Point Name (APN), which is referred to as the Data Network Name (DNN) in 5G terminology, was configured in the PCC to enable the UEs to access the KPN test network and to reach the test VM server.

The objective of the test was to observe and compare the latency, throughput, and packet loss of L4S-enabled and non-L4S traffic under background traffic load and different queue delay thresholds. The L4S feature was enabled on the gNodeB and activated on the default internet bearer, configured with 5QI 8. To ensure that the L4S QoS flow and L4S queue carried only L4S traffic, two UEs were used, each handling a different traffic type. One UE received the L4S test traffic, while the other handled the background traffic. This separation ensured that the flows were processed in different queues at the RAN.

Since only two UEs were involved, this setup removed the need for the PCF or the SMF to steer L4S traffic into a separate QoS flow, as would typically be required in multi-flow scenarios as described in Chapter 3.3.

4.2.2 L4S testing scenarios

The congestion control mechanism evaluated was UDP Prague, an open-source implementation of the Prague congestion control developed by the L4S Team [10], as described in the simulation section (Section 4.1.3). In parallel, a second laptop generated background traffic using ‘iperf3‘, which was configured to transmit UDP flows in constant bitrate (CBR) mode at a fixed rate of 2 Mbps per flow. The use of UDP and CBR ensured consistent data transmission across all test scenarios, to allow for a more controlled background traffic condition.

Each experiment involved running a continuous L4S-enabled UDP Prague flow for a total duration of 200 seconds. The following three configurations were tested:

1. L4S disabled at the gNodeB
2. L4S enabled with queue delay thresholds of 5–7 ms
3. L4S enabled with queue delay thresholds of 7–15 ms

To simulate varying congestion levels, the background traffic was introduced at predefined intervals throughout the 200 second test period. Specifically:

- **0–40 s:** No background traffic (L4S flow only)
- **40–80 s:** 13 concurrent UDP ‘iperf3‘ background flows added (L4S + iperf3 flows)
- **80–120 s:** Background traffic removed (L4S flow only)
- **120–160 s:** 18 concurrent UDP ‘iperf3‘ background flows added (L4S + iperf3 flows)
- **160–200 s:** Background traffic removed (L4S flow only)

5

Results

This chapter presents and discusses the results from both the simulation and test lab experiments described in the previous chapter. The results for various key performance indicators (KPIs) are presented and analyzed to draw conclusions regarding latency, throughput, packet loss, and rate adaptation behavior for L4S, ANBR, and their combination.

5.1 Simulation results

This section presents the results from the simulation study. The KPIs analyzed include the end-to-end latency, server transmission rate, received throughput, and packet loss, all of which were derived from the simulation logs.

5.1.1 Summary of simulation results

This subsection provides a high-level summary of the latency, throughput, and packet loss KPIs observed for L4S, ANBR, and the hybrid approach. A more detailed analysis is presented in subsequent subsections. As discussed in Chapter 4, two types of background traffic were simulated: one with relatively lower background traffic load and another with higher background traffic load. The aim is to evaluate the performance of each mechanism under both conditions, assess how each mechanism adapts its transmission rate, and determine the extent to which they improve QoS under different capacity levels.

To begin, a reference simulation scenario was run using a constant transmission rate of 30 Mbps, without any rate adaptation. This was followed by three configurations: L4S only, ANBR only, and the hybrid method which is a combination of both L4S and ANBR, as discussed in Chapter 4 (Section 4.1.5). These tests were conducted using a large buffer size of 1 GB; results for a smaller buffer are presented later in Section 5.1.6.

The throughput was calculated at the receiver end using the FlowMonitor module in ns-3, which provides detailed performance metrics per flow. The throughput was calculated as the total number of bits successfully received by the UE over a given duration. Since two L4S/ANBR UEs were involved, the throughput of each UE was summed to obtain the total throughput achieved by the gNodeB.

The end-to-end latency, also referred to as the one-way delay, was evaluated using the FlowMonitor module in ns-3. The one-way delay represents the time taken for a packet to travel from the server to the receiver. For the simulation, the one-way delay is calculated as the average delay over a time interval by summing the delays of all successfully received packets within an interval and dividing by the number of received packets over that interval.

The packet loss was also evaluated using the FlowMonitor module in ns-3. In this test, it is defined as the ratio of packets sent by the server but not received by the UE due to being dropped at the gNodeB's buffer when it becomes full as a result of packets being queued up.

As described in Chapter 4, five independent simulations were performed, and the KPIs were averaged. The results for the L4S/ANBR traffic are summarized in Table 5.1 and Table 5.2. Table 5.1 reports the KPIs for the lower background traffic scenario, while Table 5.2 presents the results for the higher background traffic scenario.

In the lower background traffic scenario (Table 5.1), both L4S and ANBR significantly reduced latency compared to the constant-rate configuration. The constant-rate configuration had an end-to-end delay of 1.73 seconds. L4S with a 5-7 ms ECN threshold reduced the end-to-end delay from 1.73 seconds to 13.31 ms, while a more relaxed L4S setting of 30–50 ms resulted in a delay of 20.16 ms. The hybrid method with an L4S threshold of 5-7 ms and an ANBR window of 40 ms, achieved a delay of 14.57 ms. ANBR alone, with a 40 ms window, had a delay of 47.21 ms.

In the higher background load scenario (Table 5.2), the constant-rate configuration led to a high queue buildup, resulting in latency of over 3 seconds. L4S with a 5-7 ms threshold reduced this to 13.87 ms, while the 30–50 ms threshold yielded a latency of 25.15 ms. The hybrid method, combining an L4S threshold of 5–7 ms with an ANBR window of 40 ms, achieved a delay of 14.18 ms. ANBR alone, with a 40 ms window, had a latency of 49.52 ms.

Throughput results show that all the rate adaptation mechanisms reduced data rates to some extent as compared to the configuration with a constant bitrate, which is expected for the case with a large buffer size. In the lower background load scenario, the constant bitrate had a throughput of 57.15 Mbps. L4S (5–7 ms) reduced throughput from 57.15 Mbps to 45.39 Mbps. The hybrid method with an L4S threshold of 5-7ms and an ANBR window of 40ms had a throughput of 47.49 Mbps. L4S (30-50 ms) achieved 53.22 Mbps. ANBR with a 40 ms window reached 56.63 Mbps.

In the high background load scenario, the constant bitrate had a throughput of 50.29Mbps. L4S (5–7 ms) reduced throughput from 50.29 Mbps to 41.05 Mbps. The hybrid method with an L4S threshold of 5-7 ms and an ANBR window of 40 ms had a throughput of 41.36 Mbps. L4S (30–50 ms) resulted in a throughput of 47.18 Mbps, while ANBR with a 40 ms window reached 49.58 Mbps.

No packet loss was observed in the tests due to the use of a large 1 GB buffer. Section 5.1.6 discusses a configuration with a reduced buffer size of 15 MB, where packet loss becomes a factor.

Rate Adaptation	Packet Loss Ratio	End-to-End Latency (ms)	Throughput (Mbps)
L4S:5-7ms	0	13.3058	45.3942
L4S:7-15ms	0	13.315	45.3865
L4S:10-20ms	0	14.2567	47.3174
ANBRL4S:5-7ms	0	14.5692	47.4894
ANBRL4S:7-15ms	0	14.6642	47.6356
ANBRL4S:10-20ms	0	15.4175	48.7866
L4S:15-20ms	0	16.14	50.3197
ANBRL4S:15-20ms	0	17.4942	51.1112
L4S:15-30ms	0	17.5792	51.7716
ANBRL4S:15-30ms	0	18.9275	52.2493
L4S:30-50ms	0	20.1592	53.2228
L4S:60-100ms	0	20.9258	53.5521
ANBRL4S:30-50ms	0	25.8458	54.4932
ANBR:40ms	0	47.2117	56.6267
ANBR:100ms	0	52.5983	56.6751
ANBR:200ms	0	67.8483	56.8589
Without	0	1734.78	57.1496

Table 5.1: Summary of simulation results under the lower background traffic load

Rate Adaptation	Packet Loss Ratio	End-to-End Latency (ms)	Throughput (Mbps)
L4S:5-7ms	0	13.8683	41.0474
L4S:7-15ms	0	13.91	41.0684
ANBRL4S:5-7ms	0	14.1808	41.364
ANBRL4S:7-15ms	0	14.2542	41.5475
L4S:10-20ms	0	14.4617	41.6686
ANBRL4S:10-20ms	0	14.8742	42.3304
L4S:15-20ms	0	16.425	43.4287
ANBRL4S:15-20ms	0	17.0875	44.4244
L4S:15-30ms	0	18.7508	44.8084
ANBRL4S:15-30ms	0	19.53	45.7337
L4S:30-50ms	0	25.1492	46.5883
ANBRL4S:30-50ms	0	27.6842	48.2587
L4S:60-100ms	0	29.995	47.1761
ANBR:40ms	0	49.5242	49.5788
ANBR:100ms	0	74.8183	49.7104
ANBR:200ms	0	104.2233	49.7621
Without	0	3380.179	50.2912

Table 5.2: Summary of simulation results under the higher background traffic load

5.1.2 Effect of the thresholds

This subsection analyzes how the different ECN marking thresholds in L4S and the window sizes in ANBR influence performance. The results, as presented in Table 5.1, Table 5.2 and in the scatterplot in Figure 5.1 and Figure 5.2, show how each mechanism behaves in terms of latency and throughput under the different threshold and window configurations. Table 5.1 and Figure 5.1 presents the case for the lower background load while Table 5.2 and Figure 5.2 presents the case for the higher background load.

Although the absolute KPI values differ between the low and high background load scenarios, the trends are similar across both. As shown in the Tables 5.1 and 5.2, and in Figure 5.1 and 5.2, reducing the ECN marking thresholds results in a reduction in latency, but this comes with a slight decrease in throughput. The observed throughput reduction is relatively minor especially in the case of ANBR (which has similar throughput values), showing that stricter delay control can be achieved without significantly compromising throughput performance. The throughput and latency trade-off in L4S (shown with circle markers in Figure 5.1 and 5.2) occurs because a higher queue threshold, delays and relaxes the marking of the packets, thus reducing the frequency of the congestion signals (i.e., the ECN CE marks). As a result, the sender maintains a higher transmission rate for longer periods, increasing overall throughput at the RAN. However, this also allows longer queues to build up before the sender reacts, resulting in increased queueing delay and higher end-to-end latency.

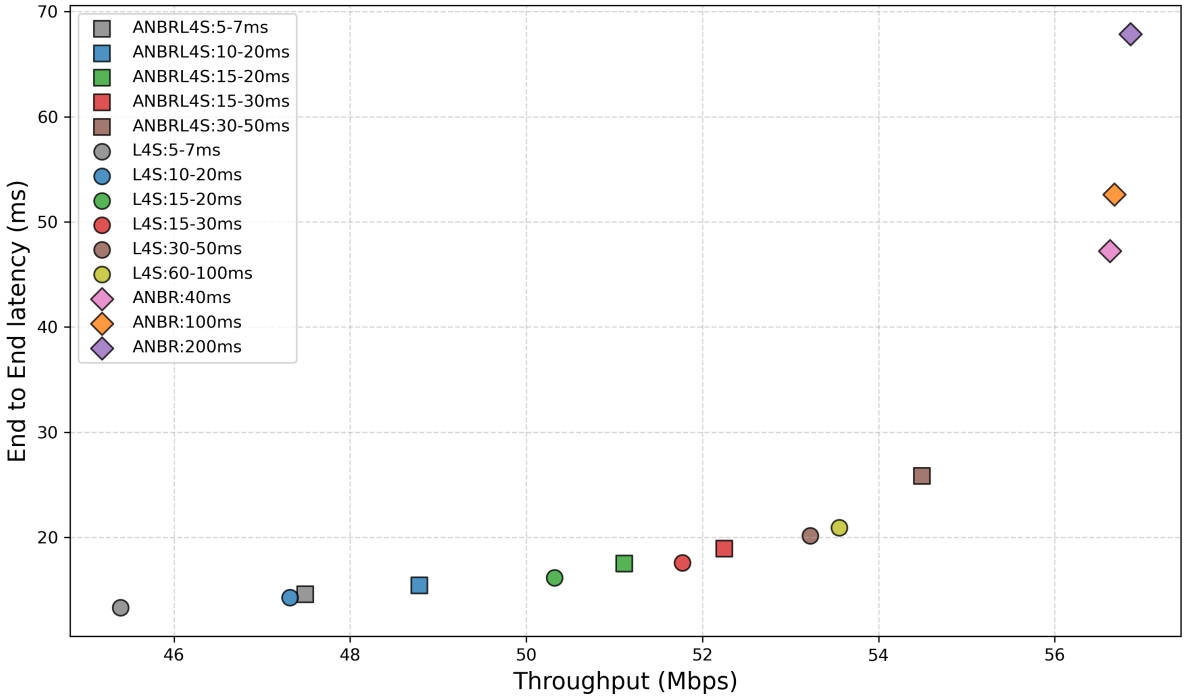


Figure 5.1: Scatter plot showing throughput vs. latency for different L4S thresholds and ANBR window sizes under the lower background load

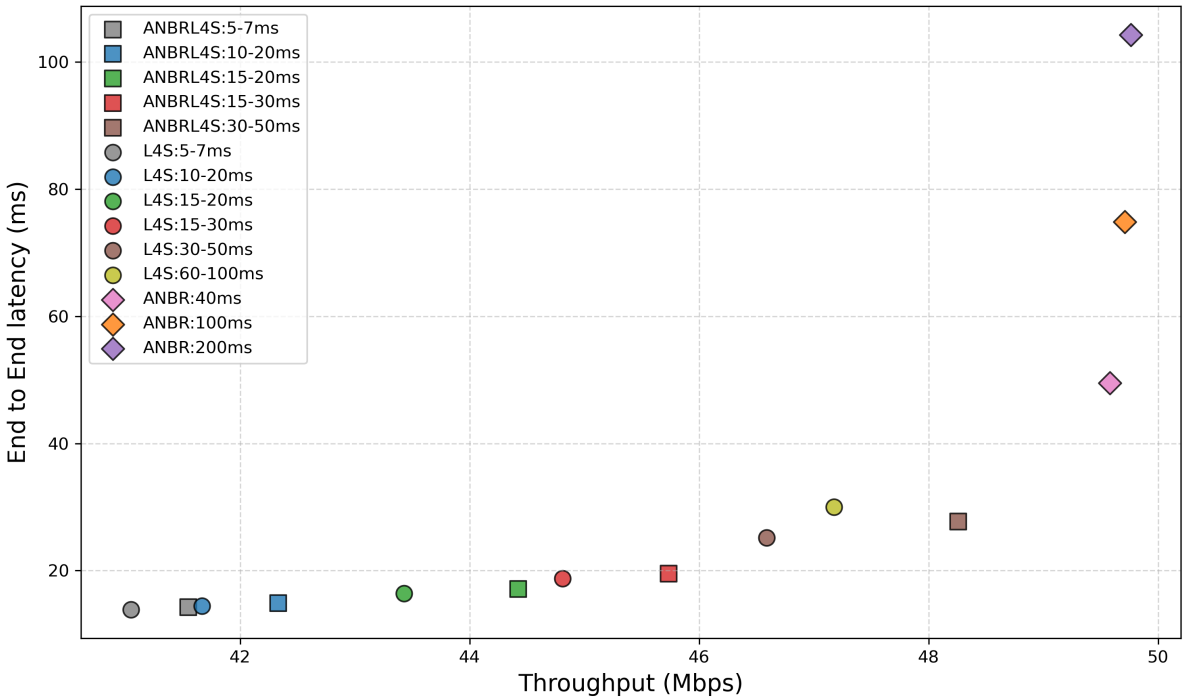


Figure 5.2: Scatter plot showing throughput vs. latency for different L4S thresholds and ANBR window sizes under the higher background load

For ANBR, the results in Table 5.1, Table 5.2, Figure 5.1 and Figure 5.2(shown with diamond markers) show that smaller window sizes lead to a decrease in latency and negligible lower throughput. Smaller window sizes allow the bitrate estimation to respond more quickly to short-term fluctuations in available network capacity. This enables the server to adapt the sending rate more accurately and maintain lower latency. However, windows that are too small can introduce instability in the estimated rate, as insufficient data within a short time frame can result in noisy or unreliable estimates. This can also cause rapid fluctuations in the sending rate and unnecessary signaling overhead. As stated in Section 4.1.4, to avoid significantly small windows, a 40 ms window was chosen as the smallest value in this simulation.

One thing to note is that the reduction in throughput when using lower window sizes in ANBR is negligible. Unlike L4S, which reacts to queueing delay and tends to keep the buffer very small, ANBR does not directly monitor queueing delay. As a result, more data remains in the buffer for scheduling, allowing throughput to remain similar in all the window values used for the experiment. This behavior is also bounded by the maximum achievable throughput that the gNodeB can provide, regardless of the rate at which the server transmits. In the case without rate adaptation, when the sender transmits at full speed in the scenario with the low background traffic load, the throughput is 57.15 Mbps, while in the scenario with the higher background traffic load, it is 50.29 Mbps. Once throughput begins to approach these limits, it saturates, and further increases in the server sending rate has little effect on the achievable throughput, even as latency continues to increase.

As also described in Chapter 4 (Section 4.1.5), a hybrid method combining L4S with ANBR was tested. The hybrid method was evaluated by varying the L4S threshold but keeping the ANBR window at 40 ms. This approach reduces the transmission rate based on L4S when ECN marks are detected, and increases the transmission rate using the maximum of L4S and ANBR's rate when no ECN marks are present. In the scatter plots (Figure 5.1 and 5.2), this hybrid method is shown with square markers.

The results indicate that the hybrid approach achieves a slightly higher throughput compared to standalone L4S with the same threshold, although still at the cost of some latency, for both background load scenarios. However, it is observed that in the higher background load scenario (Figure 5.2), the hybrid scheme with a 30–50 ms threshold (brown square marker) achieves lower latency than the 60–100 ms L4S case (gold circle marker) while also delivering higher throughput. This behavior is linked to how the 60–100 ms threshold adapts its sending rate, as will be explained in more detail in Section 5.1.4. Since the 60-100 ms configuration rarely encounters ECN marks, it relies primarily on RTT measurements. Under the higher congested scenario, the high RTT causes significant rate oscillations, leading to latency spikes but, on average, a lower transmitted rate than the hybrid 30-50 ms threshold. In contrast, the hybrid 30–50 ms case does not have such oscillations, resulting in both a higher average throughput and lower average latency than 60–100 ms standalone L4S. This is illustrated in Appendix B.

5.1.3 Received throughput

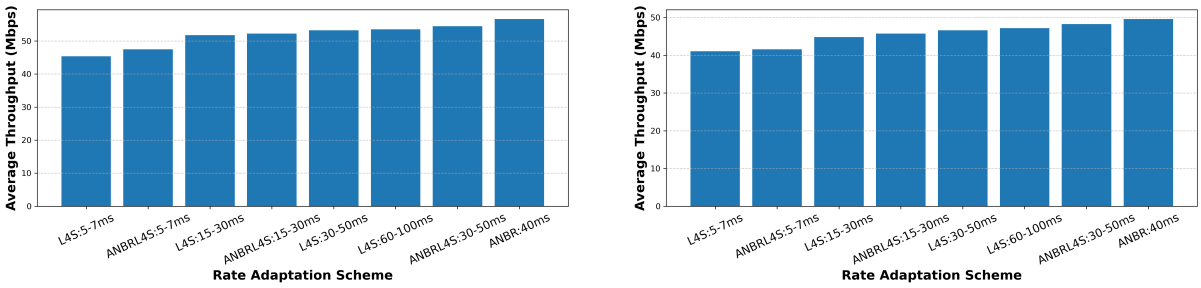
This subsection provides a more detailed analysis of the throughput performance of L4S and ANBR. The results are visualized in the bar chart in Figure 5.3a(lower background load sce-

nario) and Figure 5.3b(higher background load scenario), which clearly illustrates the differences between the various configurations. Only a subset of the configurations is shown here, as they have already been presented in Table 5.1, Table 5.2, Figure 5.1, and Figure 5.2.

The L4S mechanism generally yields lower throughput than ANBR because it actively reduces the sending rate in response to early congestion signals (ECN marks) in order to maintain low queueing delay. As stated earlier, the configurations with higher ECN thresholds result in higher throughput than lower thresholds, since packets are marked later, allowing the sender to maintain a higher transmission rate. In contrast, with lower thresholds, packets are marked earlier, causing the sender to reduce its transmission rate more often, leading to more aggressive rate reductions and thus lower throughput.

ANBR achieves higher throughput than L4S. This is because ANBR recommends a sending rate based on the estimated available capacity at the gNodeB, without directly considering current queueing delay. Since packets may already be buffered during the estimation, the recommended rate can exceed the rate needed to avoid queueing delay, leading to higher throughput but potentially increased latency.

The hybrid configuration, which combines L4S and ANBR, shows higher throughput values than L4S alone. This aligns with expectations, as the hybrid approach aims to increase the transmission rate of L4S by taking advantage of the higher rate increases of ANBR.



(a) Received throughput for the lower background load scenario

(b) Received throughput for the higher background load scenario

Figure 5.3: Received throughput under different ANBR and L4S configurations across the low and high background traffic loads

To illustrate the impact of the two background traffic scenarios, Figure 5.4 and Figure 5.5 show the received throughput over time for the L4S (5–7 ms) case under both the low and high background load, respectively, as discussed in Section 4.1.2). In the lower background traffic load scenario (Figure 5.4), the background traffic has longer idle periods, allowing the L4S and ANBR flow to have more access to radio resources and scheduled more frequently and thus resulting in higher throughput. In contrast, in the higher background traffic load scenario (Figure 5.5), the background traffic has fewer idle times, resulting in less scheduling opportunities for the L4S and ANBR traffic, and thus a corresponding reduction in throughput.

Despite these differences, the overall behavior of L4S and ANBR are consistent across both traffic conditions. As shown in Figure 5.3, L4S consistently achieves lower throughput than ANBR even for the L4S case with 30-50ms threshold.

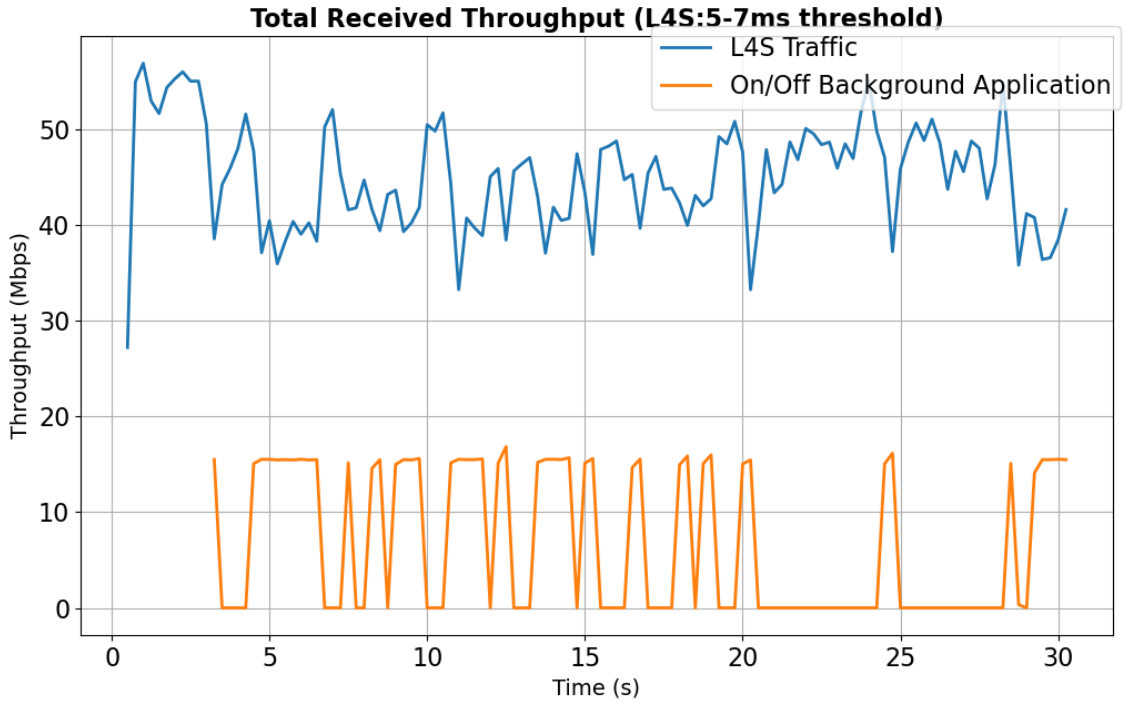


Figure 5.4: Received throughput under the lower background traffic

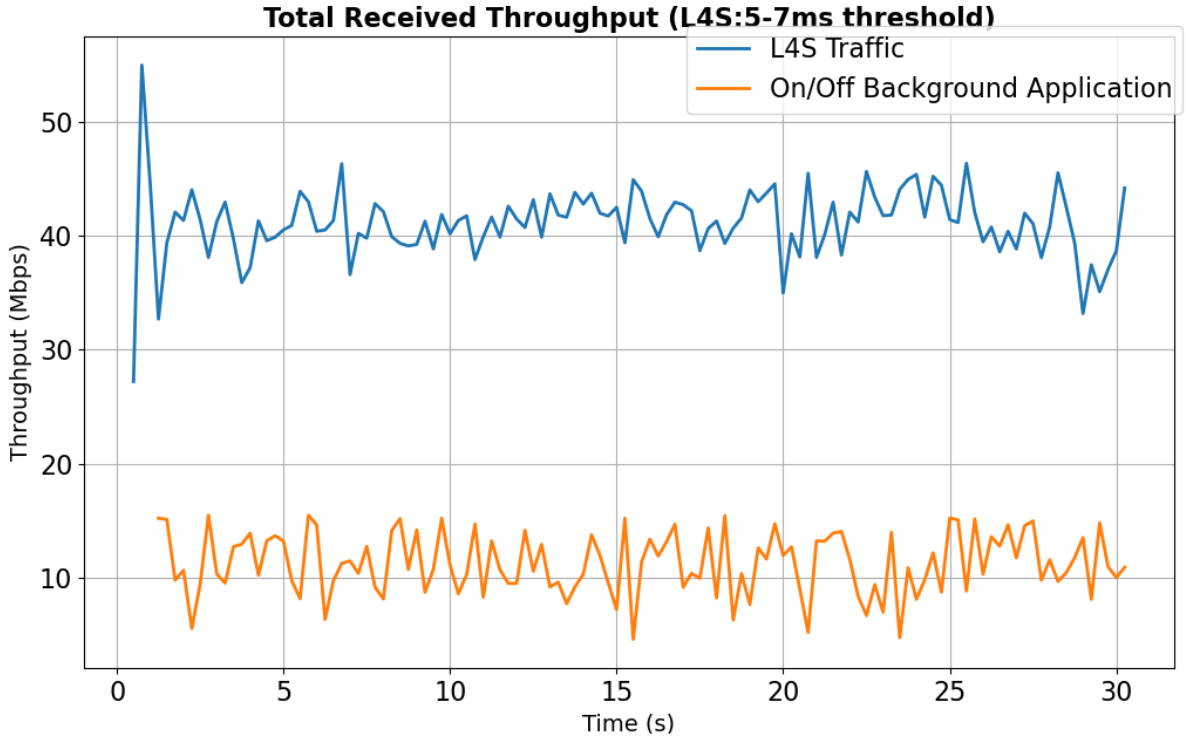


Figure 5.5: Received throughput under the higher background traffic

5.1.4 Server transmission rate

This subsection examines how the server transmission rate changes over time. The server transmission rate refers to the rate at which the application server sends packets into the network. For applications with a constant pacing rate and for this simulation, this corresponds to the application encoding rate. A key objective of the simulation is to observe how this rate adapts in response to congestion and background traffic.

As shown in Figure 5.6 and Figure 5.7, for L4S, the transmission rate decreases greatly when ECN marking increases, indicating congestion and queue build-up. Once network conditions improve and ECN marking reduces, the rate increases again.

Figure 5.6 shows this behavior in the scenario with lower background traffic load. The congestion window (purple line), grows steadily based on the number of received packets without ECN CE marks as described in Section 3.3.4 and Section 4.1.3, until it reaches a peak. At that point, the head-of-line (HoL) queueing delay (green line), also increases. When this delay exceeds the ECN marking threshold, packets are marked (red line), prompting a reduction in the congestion window. The server then lowers its transmission rate, which is calculated based on both the congestion window (purple line) and the smoothed round-trip time (orange line), as discussed in Chapter 3 (Equation 3.8). This cycle of rate increase and reduction repeats throughout the simulation.

A similar pattern is observed under the case with the higher background traffic load, as shown in Figure 5.7. In this case, ECN marks occur more frequently due to the higher load. Nevertheless, the server continues to adjust its transmission rate and congestion window based on the level of congestion, following the same overall behavior.

It is important to note that the transmission rate also depends on the smoothed round-trip time (srtt), irrespective of the presence of ECN marks. The congestion window is continuously updated whether or not ECN marks are received. It increases based on the number of received packets that were not ECN-marked, and the transmission rate is then obtained by dividing the congestion window by the srtt. This explains why slight decreases in the transmission rate can be observed even when no ECN marks are present; such reduction is a result of variations in srtt rather than the ECN congestion signals. The influence of srtt becomes particularly noticeable in the 60–100 ms case, which is discussed below.

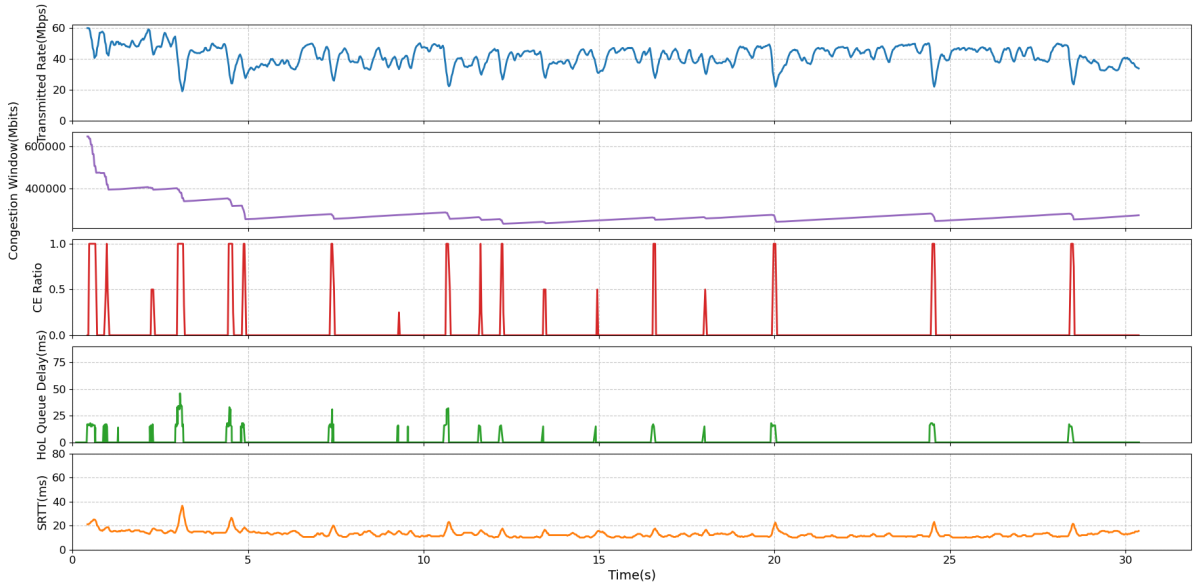


Figure 5.6: Analysis of the L4S server transmission behavior under the 5-7 ms ECN threshold for the case with the lower background traffic

Figure 5.7: Analysis of the L4S server transmission behavior under the 5-7 ms ECN threshold for the higher background traffic

For the 60–100 ms case, as shown in Figure 5.8, it is observed that there are very few ECN marks due to the high marking threshold. In this situation, the transmission rate continues to increase as though the path were uncongested. However, because the srtt is high due to high queueing delay, the actual transmission rate (congestion window divided by srtt) begins to strongly follow the srtt pattern. A higher queueing delay results in a higher srtt, and therefore, a lower transmission rate following such periods, even without ECN marks being received.

It is also observed that the congestion window (purple line) for the 60-100ms threshold reduces even when no ECN marks are received. This behavior results from the maximum data rate cap (30 Mbps). The window grows continuously with each received packet (Equation 3.7), and in the absence of congestion signaling, it keeps increasing until the calculated transmission rate exceeds 30 Mbps. The final transmission rate is set to 30 Mbps once the calculated transmission rate exceeds 30 Mbps. At that point, the congestion window is recalculated as the *transmission rate* × *srtt* (Equation 3.8), and if the srtt decreases (for example, when the queueing delay temporarily clears), the congestion window can drop even without the presence of ECN marks.

This behavior shows that L4S was specifically designed for low latency scenarios, where early congestion signals are required. When thresholds are kept low (5–7 ms), ECN marks arrive early and the flow quickly adapts its sending rate based on congestion feedback. The window reduction in that case is driven by ECN, as intended. In contrast, when thresholds are set high (60–100 ms), ECN feedback comes too late, and the srtt rather than ECN becomes the dominant factor in deriving the transmission rate.

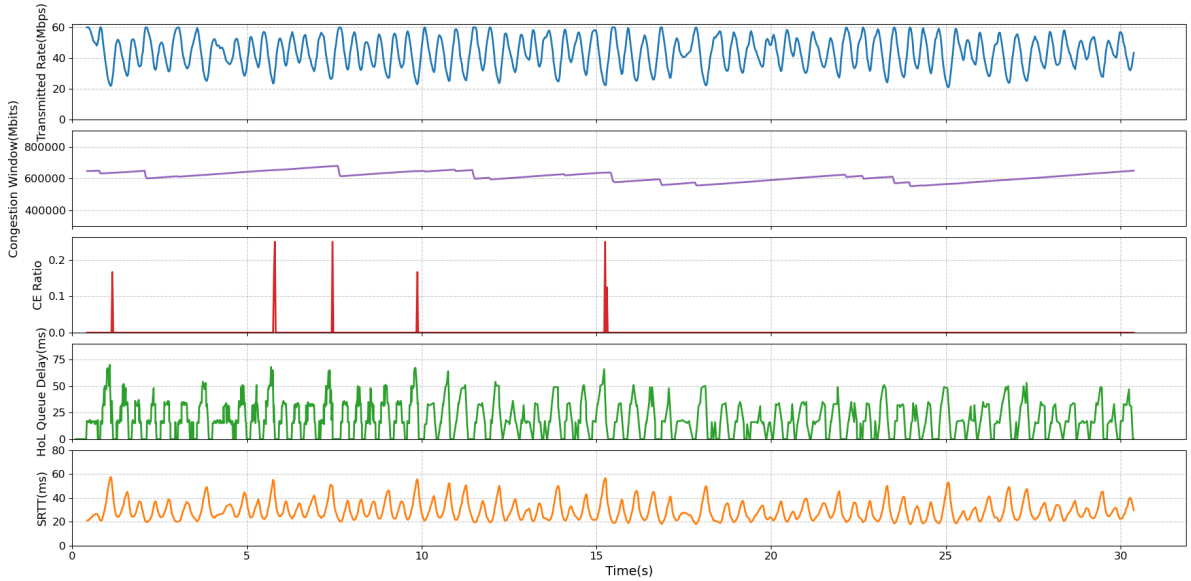


Figure 5.8: Analysis of the L4S server transmission behavior under the 60-100 ms ECN threshold for the higher background traffic

The transmission rate of ANBR was also evaluated. As shown in Figure 5.9 and Figure 5.10, ANBR generally maintains a higher transmission rate compared to L4S. This is because ANBR does not focus on the queueing delay; instead, it aims to keep the delay low while also trying to maximize bandwidth utilization based on the feedback from the gNodeB. In contrast, L4S is designed to prioritize low latency by reacting to early congestion signals, which leads it to reduce its rate much more to keep the queueing delay low.

Finally, the combined use of L4S and ANBR was analyzed. As illustrated in Figure 5.9 and Figure 5.10, the hybrid approach achieves a higher transmission rate than L4S alone, though still lower than standalone ANBR. This is especially noticeable in the case with lower background traffic (Figure 5.9). Here, the lower background traffic case allows the hybrid flow

to increase its sending rate more frequently, thus giving the ANBR component in the hybrid method more opportunities to take advantage of these increases. In contrast, with the higher background traffic (Figure 5.10), congestion occurs more often, limiting the sender's ability to raise its rate and thus restricting the throughput gains from ANBR. However, on average, the hybrid method still achieves a higher transmission rate than L4S alone. This is shown in the bar chart in Figure 5.11, which presents the average transmitted rate under higher background traffic for all thresholds. In the figure, the transmitted rate for the hybrid method (shown in red), is higher than that of L4S (shown in blue), for the thresholds.

Another observation is the increased rate reductions at certain timestamps in the hybrid method compared to L4S for both the low and high background traffic loads. During these periods, the ECN CE mark ratio of the hybrid method is higher than that of L4S, as seen in the ECN CE Marks Ratio subplots in Figure 5.9 (lower background load) and Figure 5.10 (higher background load). This higher marking is as a result of the hybrid method exceeding the ECN marking threshold more than the L4S method at those periods because it is sending at a higher rate at those times. This higher ECN marking causes the L4S component in the hybrid method to reduce its rate more aggressively. While this leads to increased throughput in the hybrid method, it also increases latency, thus highlighting the latency–throughput trade-off inherent in the approach. The corresponding throughput and latency plots are shown in Appendix B.

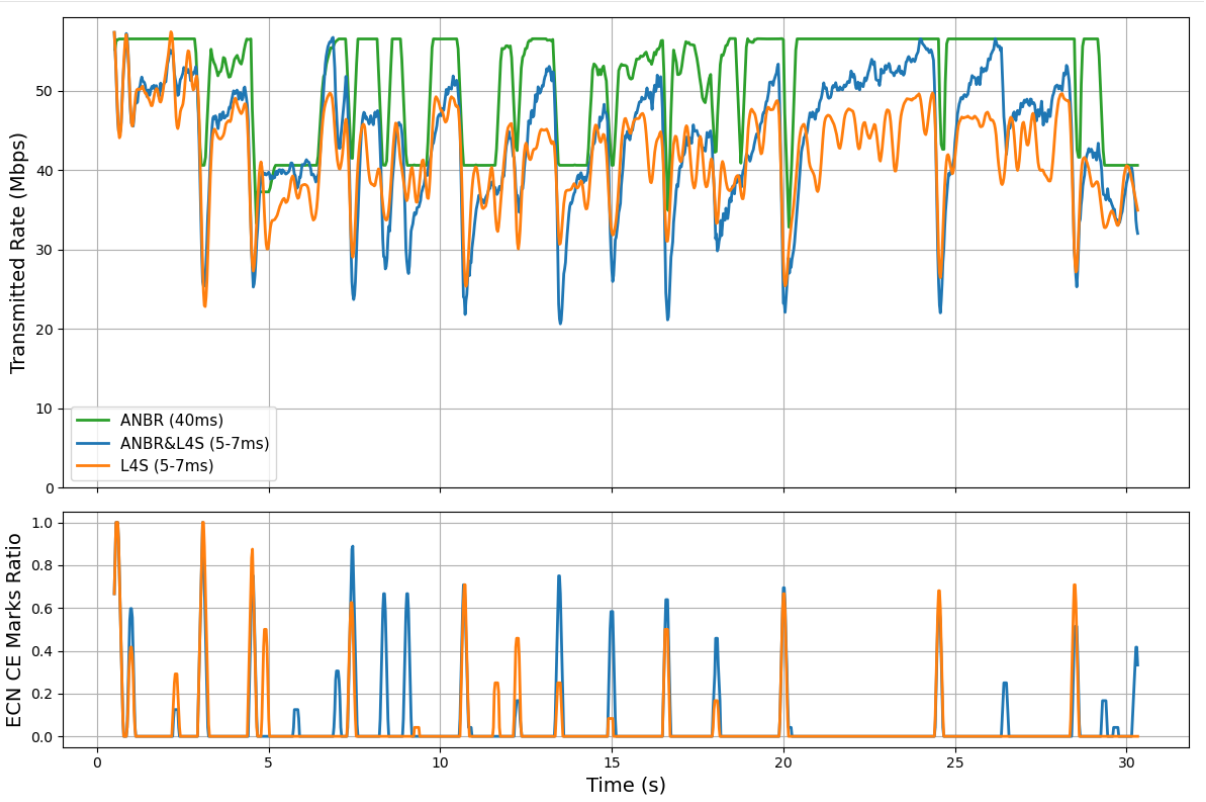


Figure 5.9: Server transmission rate for the case with the lower background traffic of L4S:5-7ms threshold, ANBR:40ms window, ANBR&L4S:40ms window and 5-7ms threshold

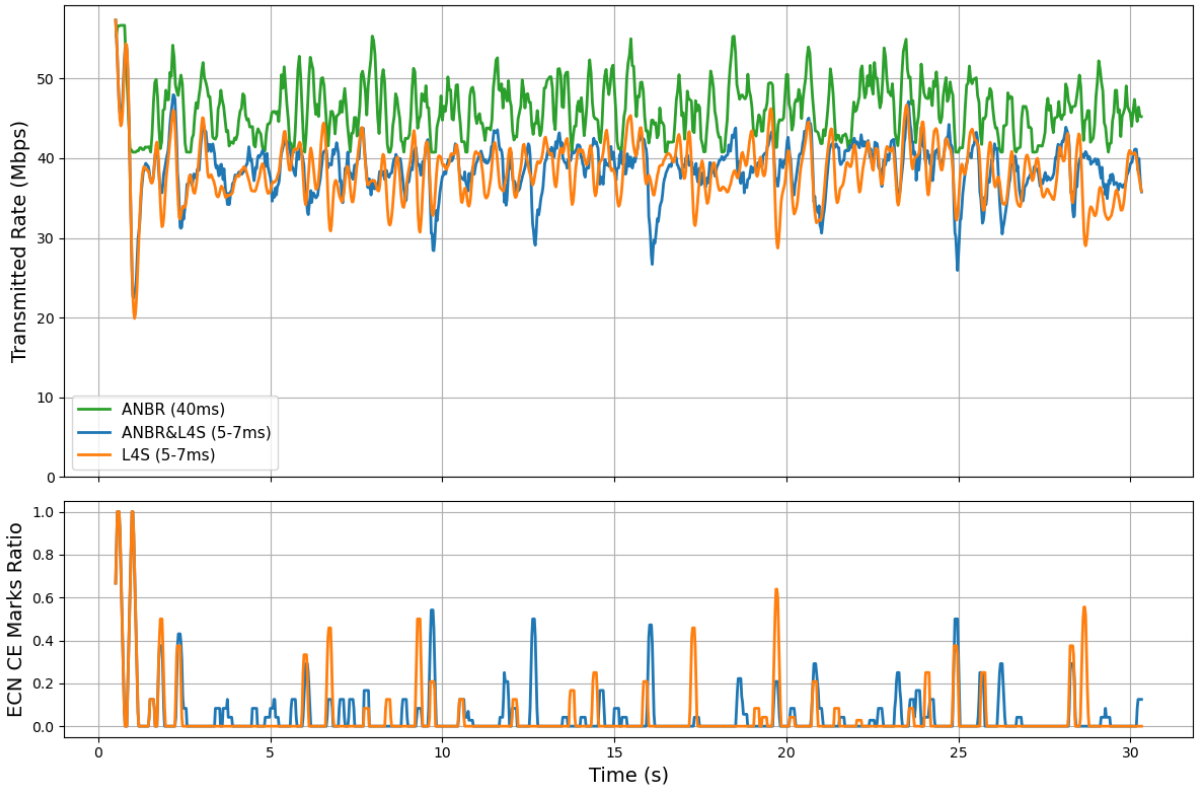


Figure 5.10: Server transmission rate for the case with the higher background traffic of L4S:5-7ms threshold, ANBR:40ms window, ANBR&L4S:40ms window and 5-7ms threshold

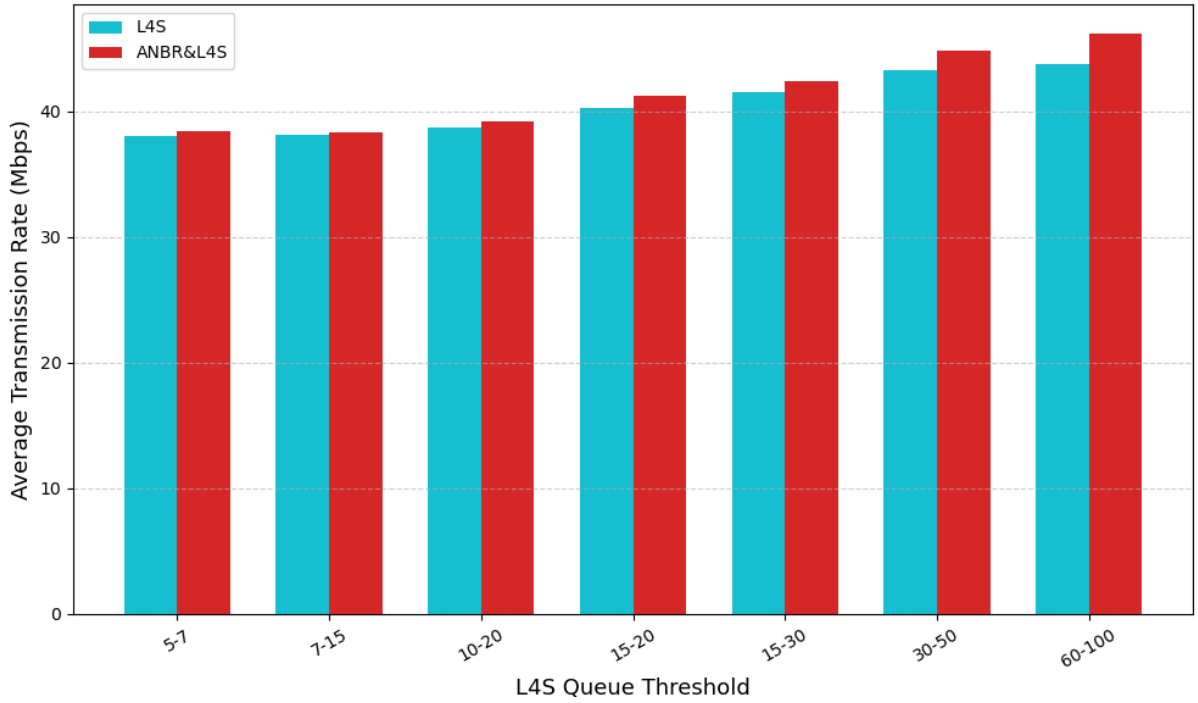


Figure 5.11: Transmission rate for L4S and the hybrid method under the high background traffic with the different threshold values

5.1.5 End-to-End latency

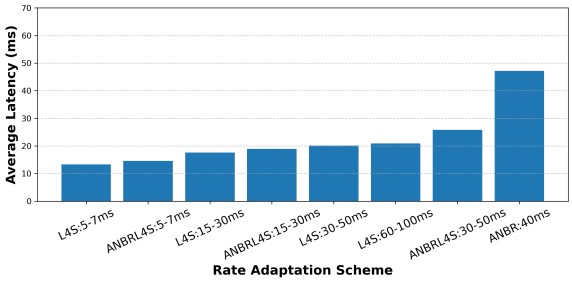
This subsection analyzes the latency behavior in more detail. Figure 5.12 presents a bar chart of the average one-way delay across different configurations. As shown, latency is lowest when L4S is active, particularly with the 5-7 ms ECN threshold, which increases as the ECN threshold values increase.

To provide additional insight, a cumulative distribution function (CDF) plot is shown in Figure 5.14, to highlight the distribution of the delay values under the different L4S thresholds. With 5-7 ms and 7-15 ms thresholds, approximately 88% of delay samples fall below 15 ms. In contrast, only around 21% of packets fall under 15 ms with the 15-30 ms threshold. This percentage drops further with larger thresholds: roughly 1% for 30-50 ms and nearly 0% for 60–100 ms. These values further strengthen the fact that higher thresholds lead to delayed congestion signaling and larger queueing delays, resulting in higher latency for a greater number of packets.

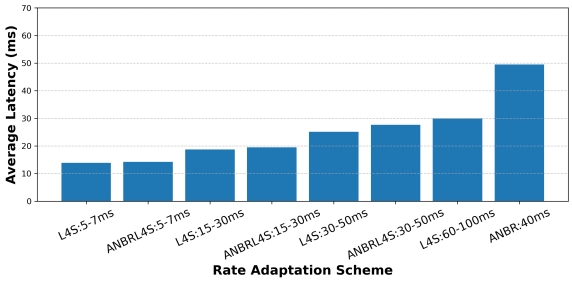
The bar chart in Figure 5.12 reiterates and visualizes the fact that ANBR with a window of 40 ms has the highest average delay among the L4S configurations.

The hybrid approach, which combines L4S and ANBR, shows latency characteristics that fall between those of the standalone mechanisms. As expected, in the bar chart (Figure 5.12), when comparing the hybrid mechanism with threshold 5-7ms and windowing of 40ms, its one-way delay is higher than L4S alone (with the same threshold configuration of 5-7ms), but lower than ANBR. This is because the hybrid method attempts to strike a balance between minimizing delay (through L4S) and achieving higher throughput (through ANBR). Finally,

Figure 5.13 illustrates the head-of-line (HoL) queueing delay for each scheme. The HoL delay increases with higher thresholds, which aligns with the increase in average latency seen in Figure 5.12, showing how the different schemes affect queueing and overall delay.

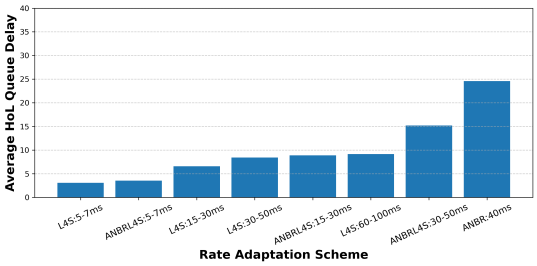


(a) Latency under the lower background load

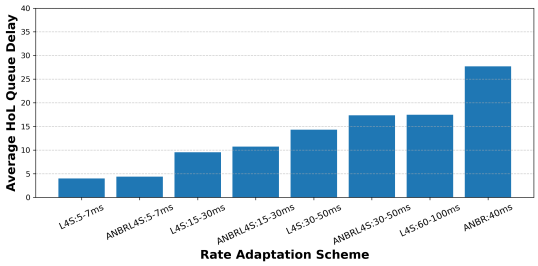


(b) Latency under the higher background load

Figure 5.12: Latency for different ANBR and L4S configurations under the two background load scenarios



(a) Average HoL queueing delay under the lower background traffic



(b) Average HoL queueing delay under the higher background traffic

Figure 5.13: Average Head of Line (HoL) queueing delay under the two background load scenarios

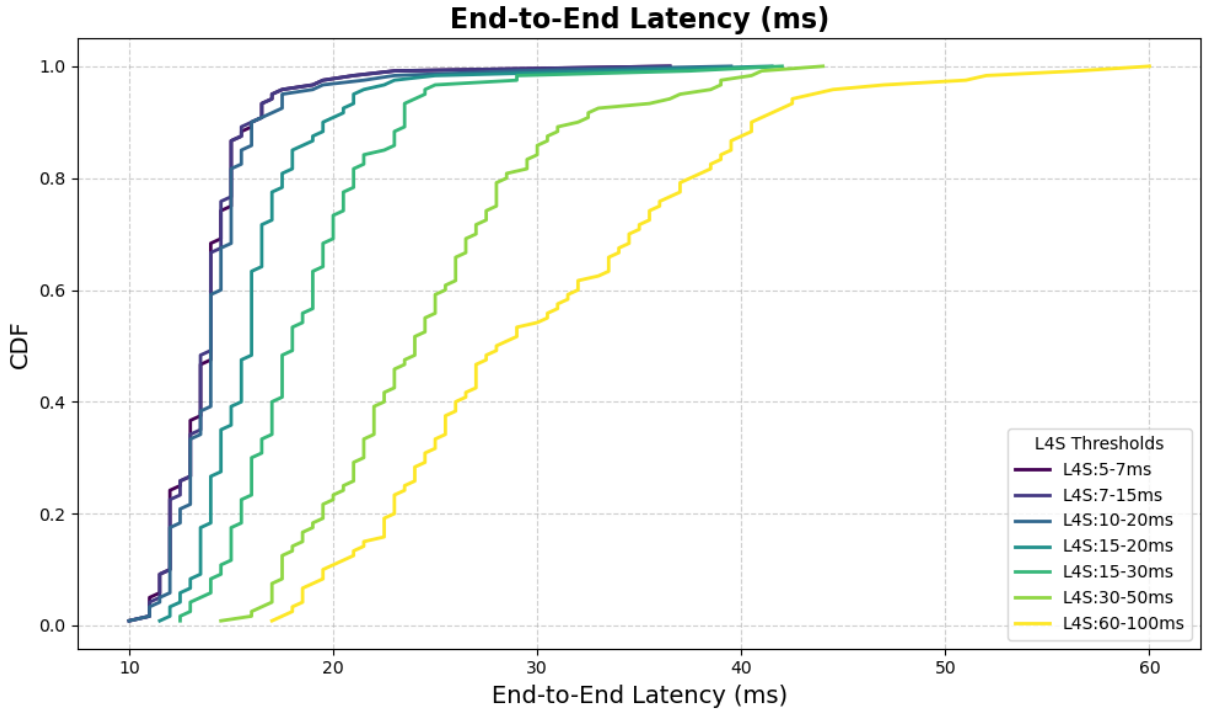


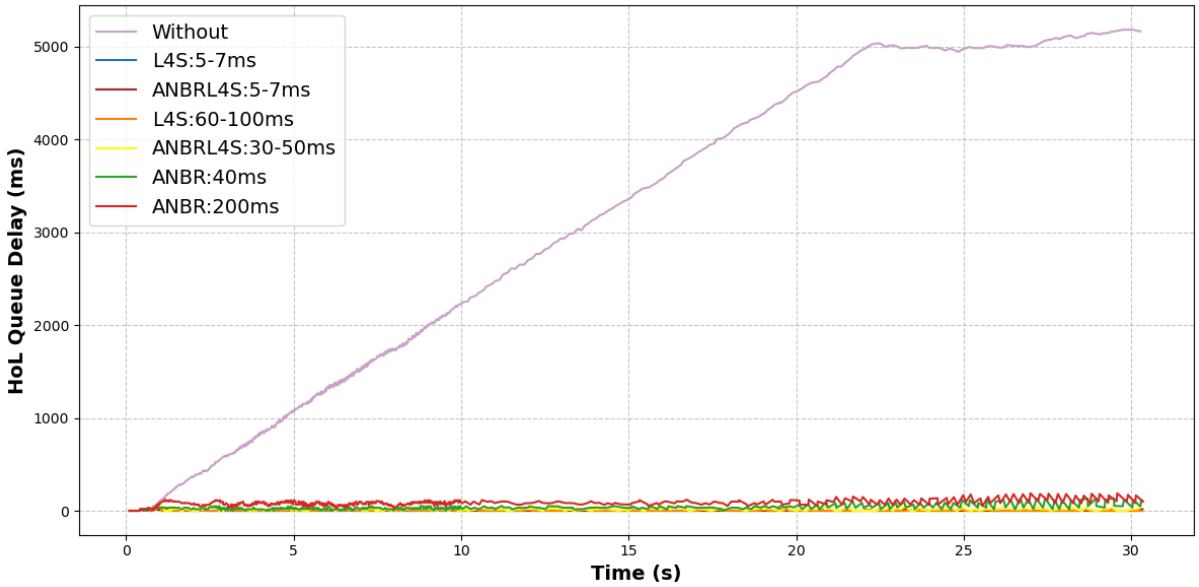
Figure 5.14: CDF for the end to end latency for L4S thresholds

5.1.6 Effect of reducing the buffer size on packet loss ratio

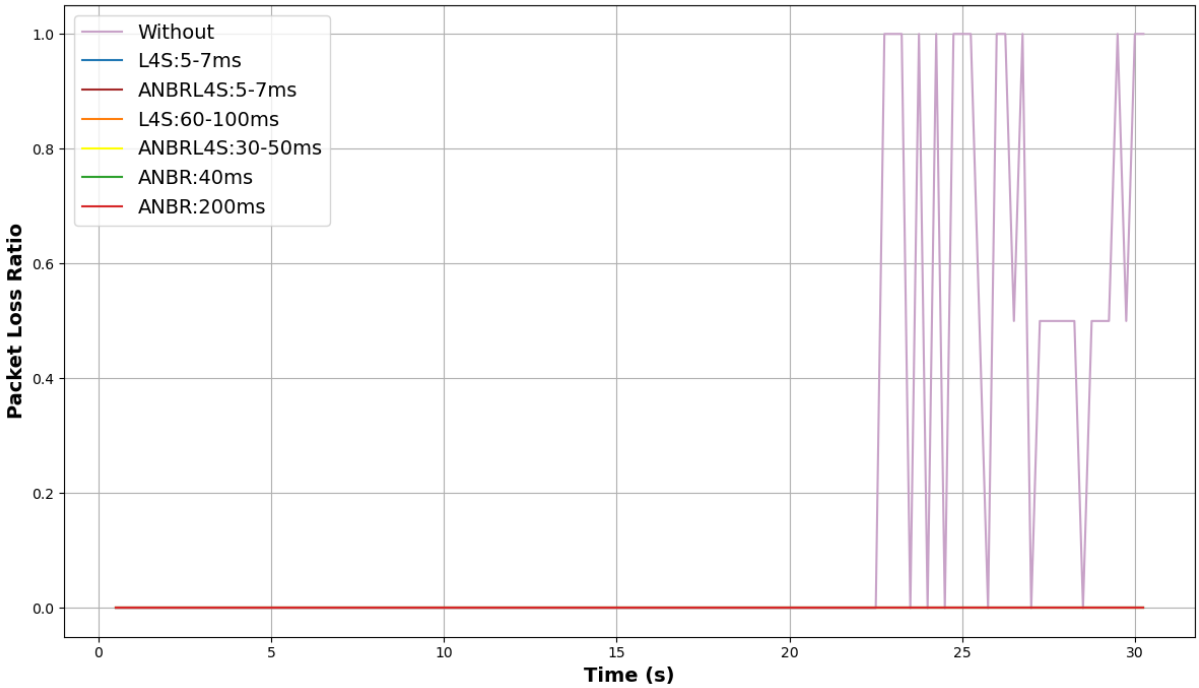
To evaluate the impact of different rate adaptation mechanisms on packet loss, the buffer size at the gNodeB was reduced to 15 MB. This allows us to observe how each mechanism behaves under minimized buffer conditions. This subsection presents the packet loss and throughput results observed when the buffer size was set to 15 MB.

The packet loss is defined as the ratio of packets sent by the server but not received by the UE due to being dropped at the gNodeB's RLC buffer when it becomes full. In earlier evaluations, a large buffer size of 1 GB was used to avoid buffer overflow and thus no packet drops occurred. However, with the reduced buffer size, the case without any form of rate adaptation experiences buffer buildup, eventually leading to queue overflow and packet loss. As shown in the bar chart in Figure 5.16a, the packet loss ratio in the no-adaptation scenario increases to 1.16%. In contrast, L4S and ANBR maintain a packet loss ratio of 0%, even under the same buffer constraints. L4S threshold of 60-100 ms and ANBR threshold of 200 ms were also visualized to show the extreme values of the simulation and in both scenarios, packet loss was zero.

As seen in Figure 5.15, the queue delay and packet loss plots further confirm this behavior. In the no-adaptation case, queueing delay (Figure 5.15a) continuously increases until about the 22nd second in which the buffer limit is reached, at which point, packets begin to drop (Illustrated in Figure 5.15b). For L4S and ANBR, the delay is being controlled through rate adaptation, thus the delay does not get so high, thereby, preventing the buffer from becoming full and consequently avoiding any packet loss.

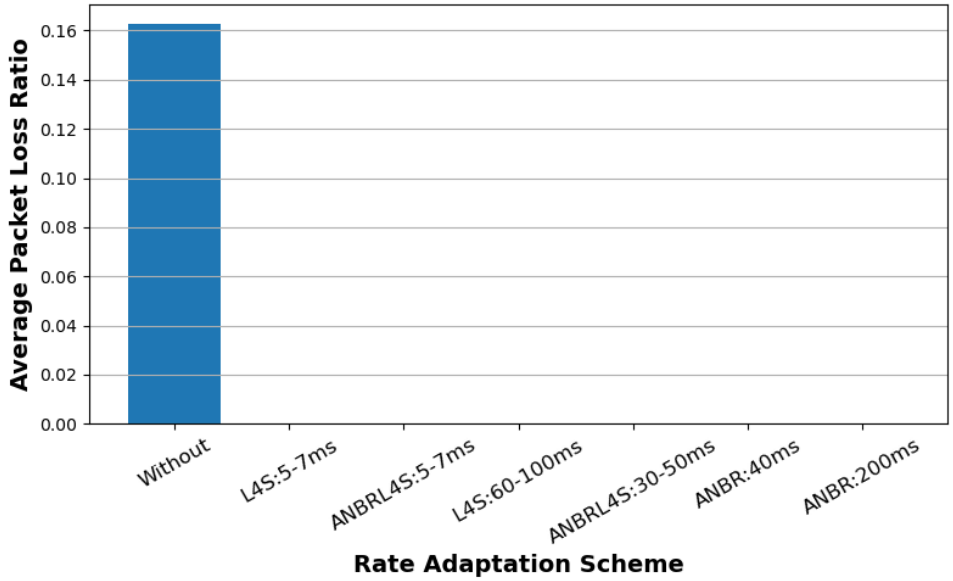


(a) Head-of-Line (HoL) Queue Delay

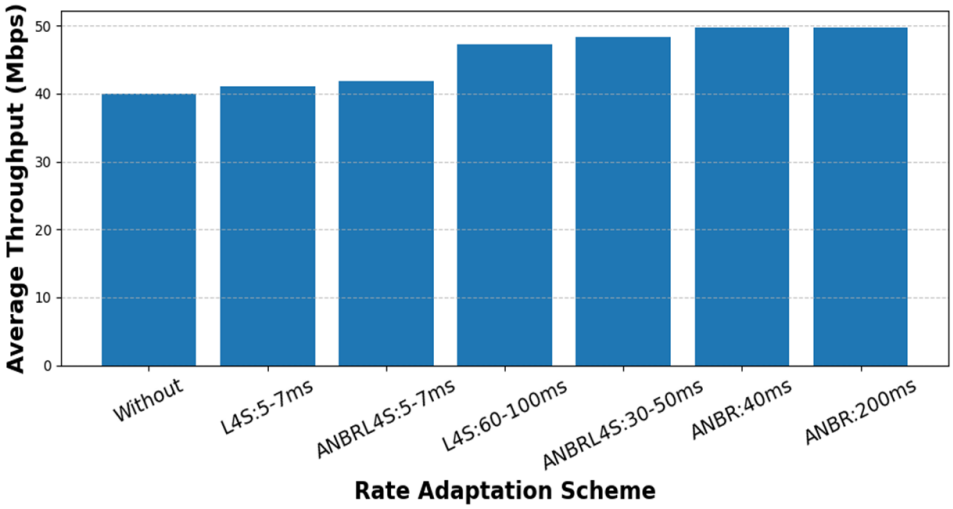


(b) Packet Loss Ratio

Figure 5.15: Queueing behavior and packet loss when the buffer size is reduced



(a) Packet Loss Ratio



(b) Throughput

Figure 5.16: Packet loss and throughput when the buffer size is reduced

5.2 L4S testlab results

This section presents the results obtained from the KPN 5G SA testbed based on the configurations outlined in Chapter 4 (Section 4.2).

5.2.1 Verification of ECN Bit preservation across the KPN test network

The first step in the test lab was to verify that ECN bits were not altered or overwritten by intermediate nodes in the network. This check was necessary to ensure that when the server sets the ECN bits to "01" (ECT(1)), as explained in Chapter 3, that they are preserved end-to-end and correctly interpreted at the receiver.

This validation confirms two things:

1. The server is marking packets as L4S-capable (setting ECN bits to 01).
2. The gNodeB is performing ECN marking when congestion occurs (setting ECN bits to 11).

To verify this, Wireshark was used at the receiver to check the ECN field in the IP headers of incoming packets. As shown in Figure 5.17, the part highlighted in orange arrived with the ECN field set to "ECN-Capable Transport Codepoint 01 (ECT(1))". This indicated that either the queueing delay was below the ECN marking threshold, or the packet was not marked due to probabilistic marking, where packets are marked based on the amount of queueing delay. In any case, this confirmed that the L4S capability was correctly signaled by the sender and that intermediate nodes had not changed the ECN bit.

As shown in Figure 5.18, the packets had the ECN field changed to "Congestion Experienced (CE)", with a codepoint value of "11". This indicates that the queueing delay at the gNodeB had exceeded the marking threshold and the packet was captured by the probabilistic marking. This verified that the gNodeB was performing ECN marking in response to congestion, as expected.

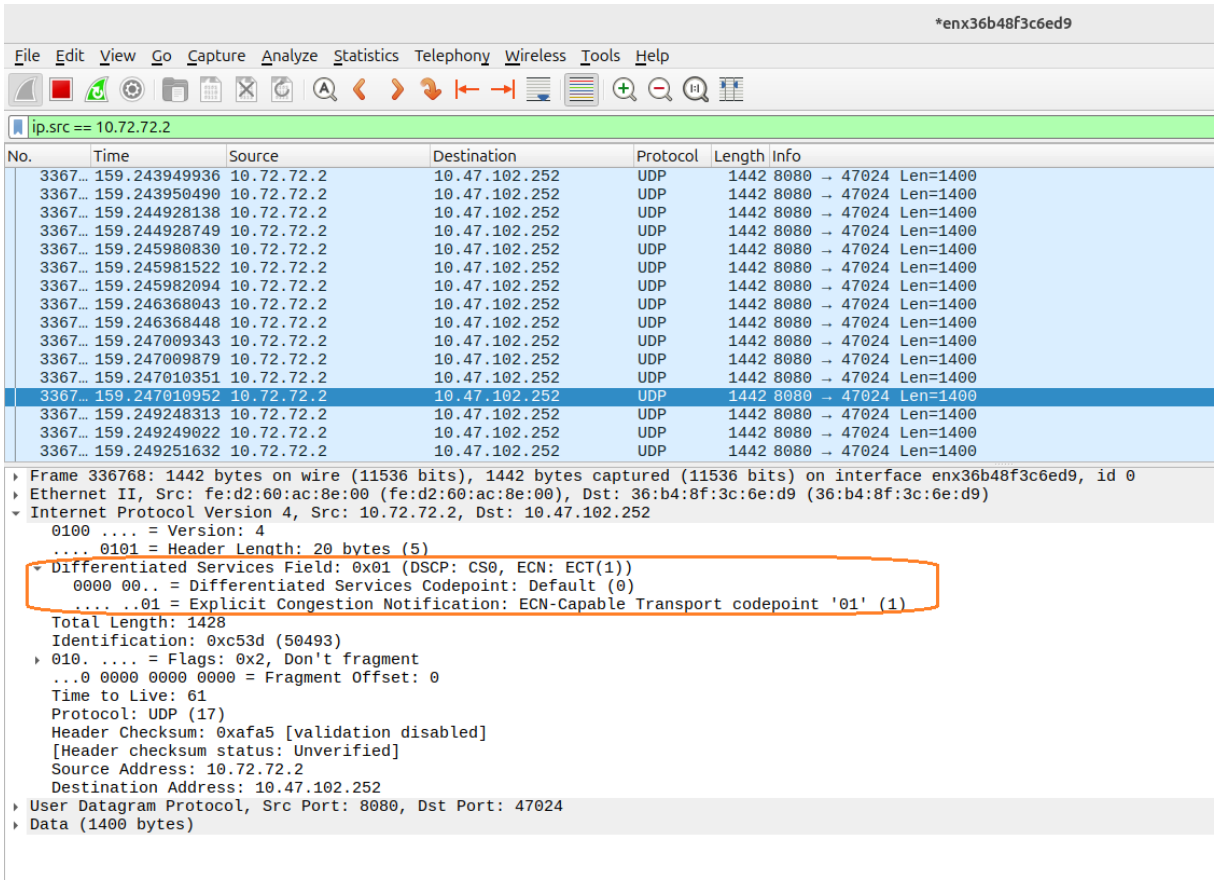


Figure 5.17: Wireshark capture at the receiver showing ECN-Capable Transport Codepoint 01 (ECT(1)) packets detected

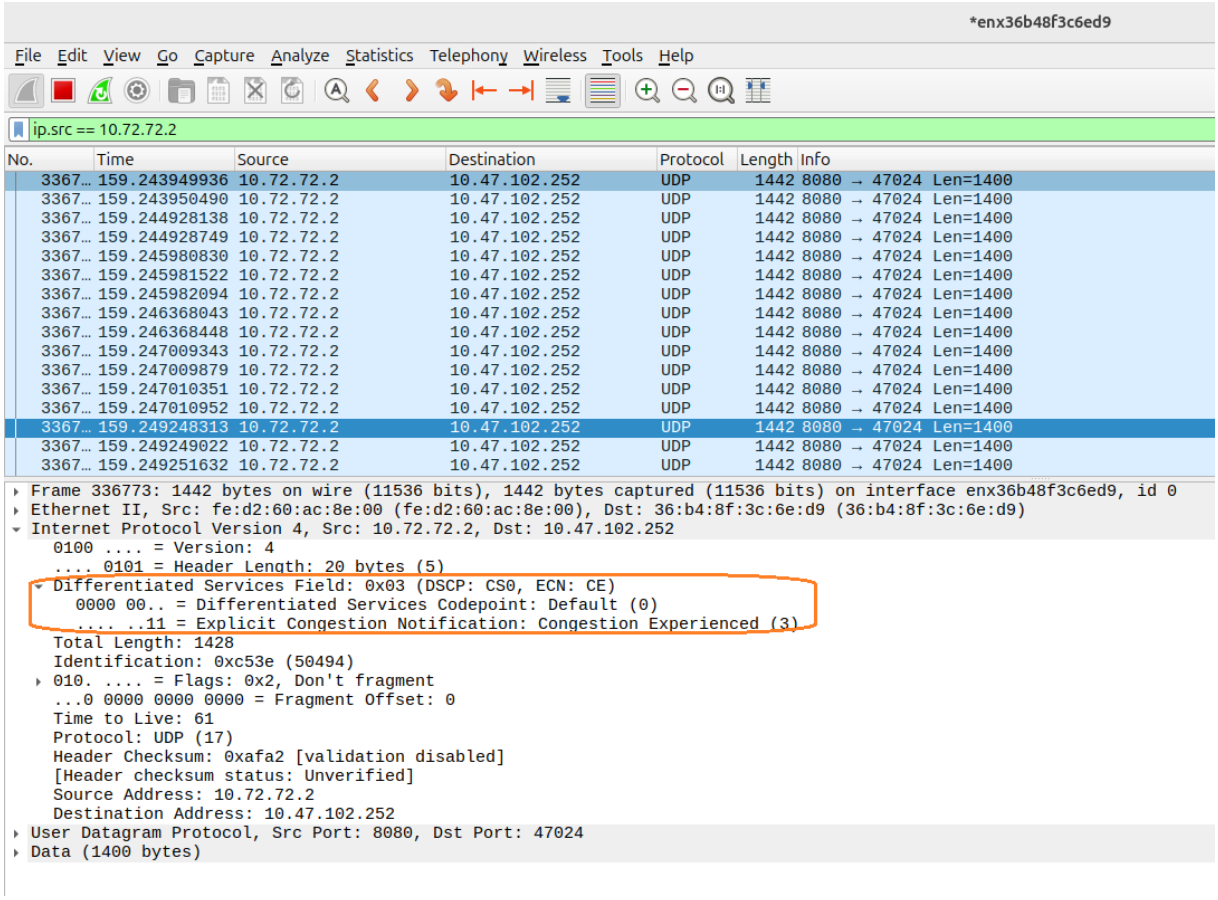


Figure 5.18: Wireshark capture at the receiver showing Congestion Experienced, with codepoint 11 packets detected

To further confirm that these bits were not being tampered with, the UDP Prague congestion control algorithm was used to monitor the ECN field and ensure that no unexpected values were observed. Specifically, it was verified that no packets had their ECN bits changed to any values other than "01" (ECT(1)) or "11" (Congestion Experienced). As shown in Figure 5.19, this metric reported 0 for all packets, indicating that no errors were detected.

These observations confirm that KPN's network infrastructure under the test supports L4S traffic correctly. ECN bits are preserved throughout the transmission path in the test infrastructure, and L4S is properly implemented by the gNodeB, fulfilling the requirements for L4S.

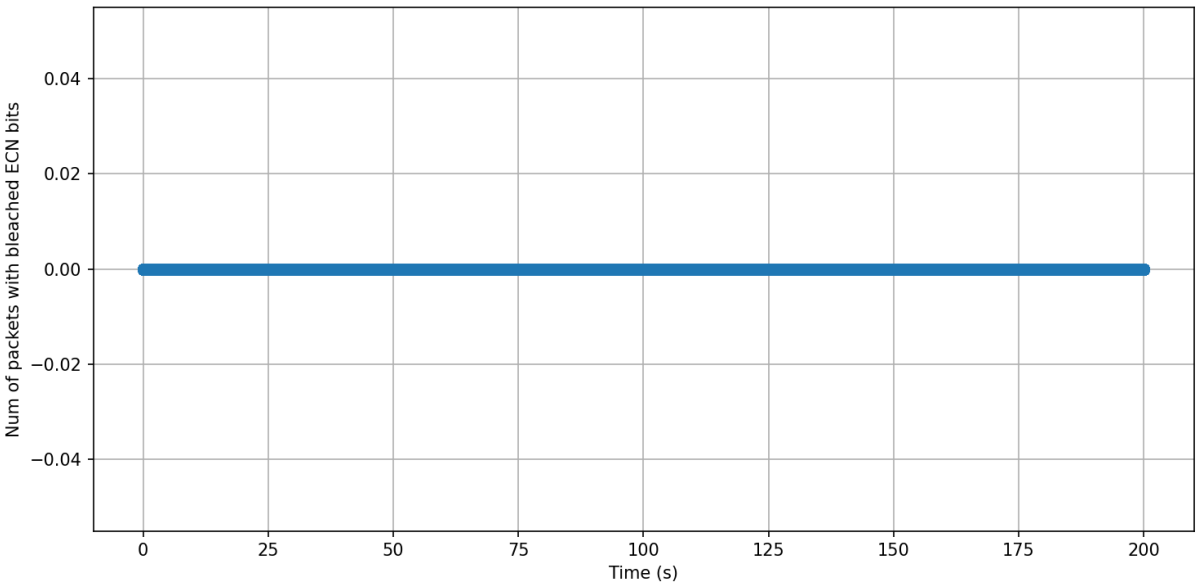


Figure 5.19: Number of packets with ECN bits that have been changed to values other than 01 or 11

5.2.2 Key performance indicators

The results presented in this section include the received throughput, RTT, ECN marking rate, packet loss, and transmitted rate from the testlab evaluations. These metrics were obtained from both Wireshark and the UDP Prague test tool.

The received throughput was measured at the laptop using Wireshark's I/O statistics, computed as the number of bits received over 1-second intervals. The RTT was derived from the UDP Prague test logs and defined as the time taken for a packet to reach the receiver and for its acknowledgment to return to the sender. The ECN marking rate and packet loss were also extracted from the UDP Prague logs. Packet loss is defined as the number of packets lost divided by the number of packets sent. Similarly, the ECN marking rate is calculated as the number of ECN-marked packets divided by the number of packets received, multiplied by 100 to get a percentage.

The transmitted rate, which in this case is interpreted as the application's encoding rate, was recorded as the rate at which the server sends packets, again derived from the UDP Prague logs.

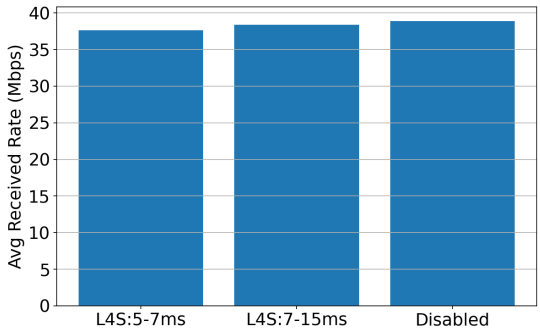
Figure 5.20 shows the average values of the received throughput, RTT, ECN mark rate and packet loss under three conditions: when L4S is disabled at the gNodeB, and when L4S is enabled with queue marking thresholds of 5–7 ms and 7–15 ms. The results are consistent with expectations and align with the simulation findings. When L4S is enabled, the received throughput slightly decreases, which corresponds with the increased ECN marking at the gNodeB.

The average RTTs, shown in Figure 5.20b, are as follows: L4S 5–7 ms = 21.96 ms, L4S 7–15 ms = 25.51 ms, and L4S disabled = 37.15 ms. These results indicate a latency improvement of approximately 40.9% with the 5–7 ms threshold and 31.3% with the 7–15 ms threshold, com-

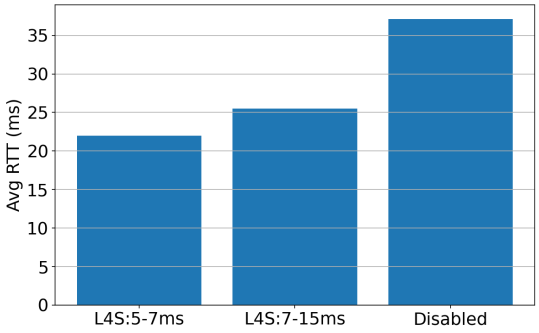
pared to when L4S is disabled. However, this improvement comes at the cost of a slight reduction in throughput.

The corresponding average throughputs, shown in Figure 5.20a, are: L4S 5–7 ms = 37.66 Mbps, L4S 7–15 ms = 38.40 Mbps, and L4S disabled = 38.91 Mbps. The ECN marking rate, as illustrated in Figure 5.20c, is higher for the 5–7 ms threshold (2.71%) compared to the 7–15 ms threshold (1.09%), since the lower delay threshold causes the gNodeB to mark packets earlier and more frequently.

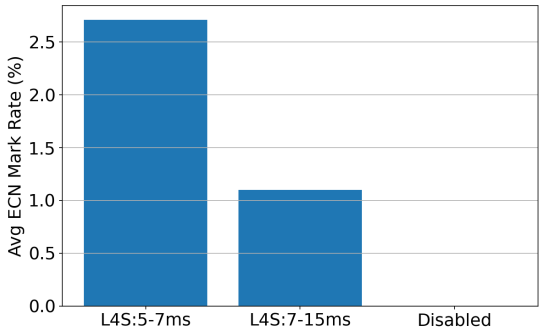
As shown in Figure 5.20d, the packet loss rate for both L4S configurations is 0%. A minimal loss rate of 0.00014% was observed when L4S was disabled. This is negligible and not directly caused by buffer overflow. Overall, L4S shows no packet loss in either configuration, even under congestion.



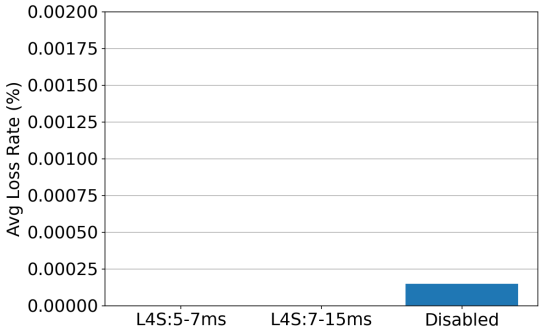
(a) Received Rate



(b) RTT



(c) ECN Congestion Experienced (ECN CE) Marking Rate



(d) Packet Loss Rate

Figure 5.20: Bar Chart Comparison of the Transmission Metrics across the Test Configurations

Figure 5.21 shows the time-series behavior of the transmitted rate, ECN mark rate and RTT under L4S and when L4S was disabled at the RAN. The UDP Background Traffic shown in figure 5.22 was introduced at the 40th second, which consisted of 15 flows at 2 Mbps each (totaling 30 Mbps). This traffic is removed after 40 seconds and reintroduced at the 120th second with 18 flows (36 Mbps) for another 40 seconds.

Figures 5.21 show that the RTT is lowest when L4S is configured with a 5–7 ms marking threshold, followed by the 7–15 ms threshold. The highest RTT is observed when L4S is dis-

abled in the RAN. As discussed earlier, this reduction in latency with L4S comes at the cost of a reduced transmission rate and, consequently, slightly lower throughput.

When L4S is disabled in the RAN, the UDP Prague algorithm relies only on the RTT to estimate the transmission rate. Since it does not receive ECN feedback from the network, it cannot respond to the queueing delay as efficiently as L4S. As a result, the sender transmits more data than the network can handle, leading to queue build-up and increased RTT.

The values in the figure are averaged over 1-second intervals to improve readability. Although the congestion control algorithm operates in milliseconds, raw values at that timescale are too detailed and would be difficult to visualize.

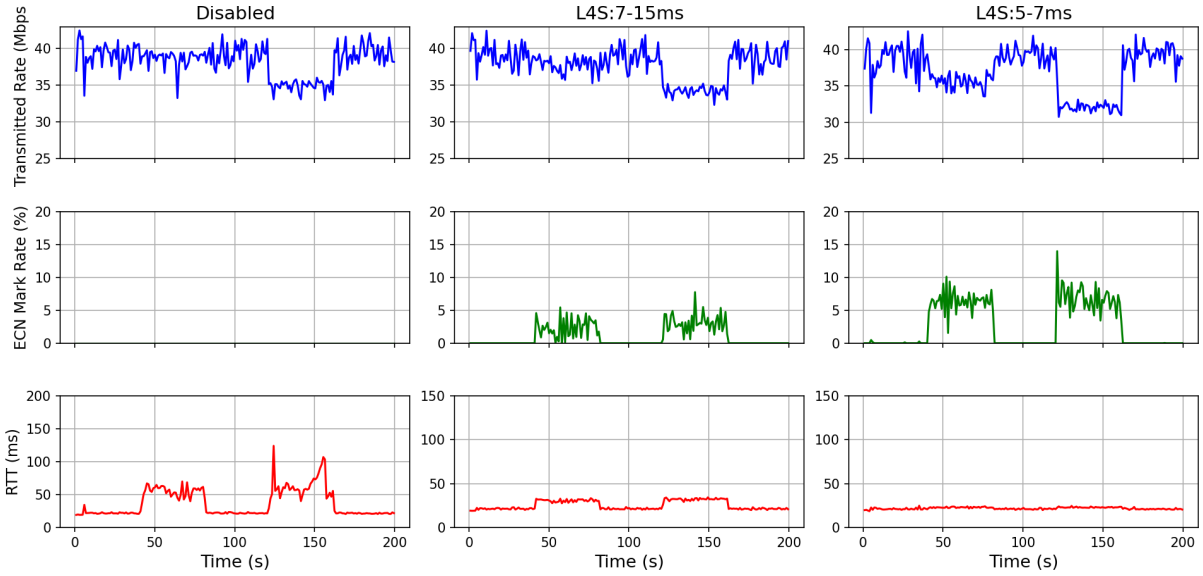


Figure 5.21: Testlab Results:Time series

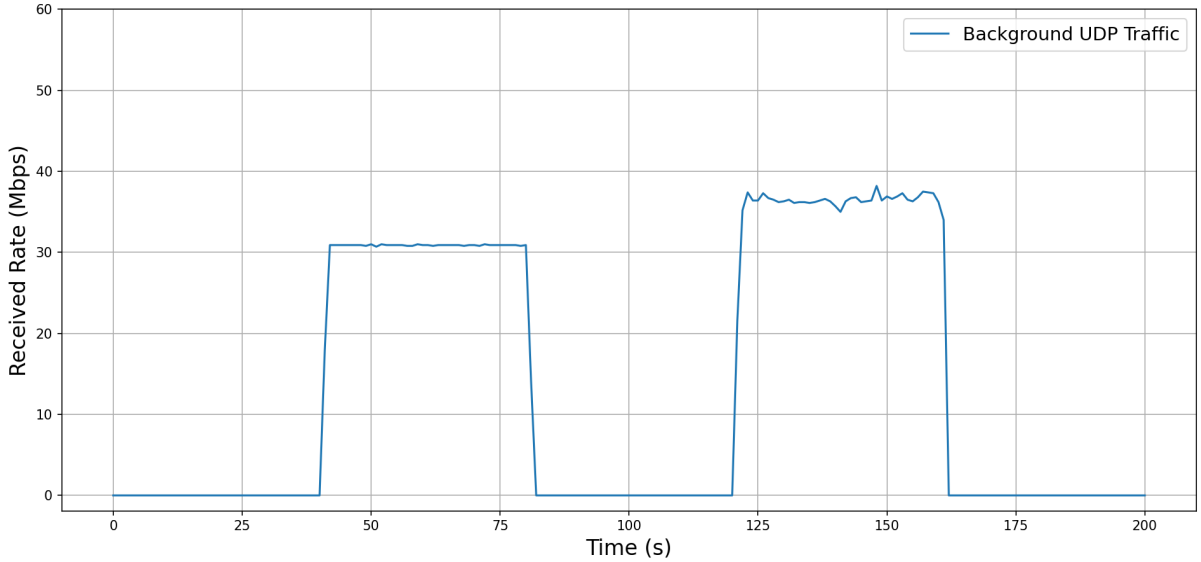


Figure 5.22: Rate of UDP Background Traffic

6

Conclusions

This chapter presents the conclusions of this thesis, summarizing key findings and providing recommendations for future work. Specifically, it revisits the research (sub)questions which were mentioned in Chapter 1, with answers based on the research done.

6.1 Research findings

As mentioned in Chapter 1, the main research question guiding this thesis was: *How can a mobile network operator integrate Low Latency, Low Loss, Scalable Throughput (L4S) and/or Access Network Bitrate Recommendation (ANBR) into its network?*

To answer this, several sub-questions were defined. Below, each sub-question is addressed based on the results of this research.

SQ1: How do L4S and ANBR affect key network performance indicators such as latency, packet loss, and throughput?

The literature review in Chapter 2 presents studies showing that L4S adapts its rate to improve latency without significantly compromising throughput. L4S was shown to reduce latency compared to Google Congestion Control, resulting in improved playback stability. Although there were slight reductions in video quality and temporary rate reductions, the latency gains were more worthwhile. Furthermore, in literature, L4S, using TCP Prague, was also compared with TCP Cubic and BBR. It achieved the best latency performance while also avoiding the significant throughput reduction that BBR undergoes when it needs to reduce its rate to calculate the minimum round-trip time.

Regarding ANBR, its evaluation in literature is limited and primarily focused on VoLTE and VoNR. The focus mainly arises from the ability of ANBR codec rate adaptation to improve packet losses and voice quality due to poor radio conditions by reducing the codec rate. One study assessed its effectiveness in maintaining QoE at the cell edge for VoNR, finding that handover strategies were more effective. Other work mentioned ANBR in relation to codec rate adaptation, but its application to services beyond voice has received minimal attention in literature.

This thesis compared the performance of L4S and ANBR for high-rate applications through 5G ns3 simulations and real-world experiments in KPN's 5G SA test network. In the simu-

lation environment, both L4S and ANBR were evaluated, while in the test lab, only L4S was done. The UDP Prague congestion control algorithm was used in both environments. Results showed that both L4S and ANBR significantly improved latency and improved packet loss compared to non-adaptive scenarios. However, L4S achieved better latency reduction than ANBR, even at higher thresholds, while ANBR delivered higher throughput. The latency and throughput values were also impacted by the L4S thresholds and ANBR window sizes. Lower L4S thresholds resulted in reduced latency but slightly lower throughput. Similarly, reducing the ANBR window size decreased latency, while its impact on throughput was negligible. It was further observed that with a 60–100 ms ECN threshold, L4S rarely reacted to ECN marks. Instead, the congestion control relied primarily on RTT behaviour: as queuing delay increased, the smoothed RTT increased sharply, which in turn reduced the sending rate without frequently triggering the ECN threshold. To further investigate packet loss, the buffer size was reduced in the simulation. Both L4S and ANBR maintained zero packet loss, in contrast to the non-adaptive case, which experienced packet loss.

SQ2: What are the comparative advantages and disadvantages of L4S and ANBR?

A comparative evaluation of L4S and ANBR is detailed in Chapter 3 and summarized in Table 3.2 and Table 3.3.

One of the primary advantages of L4S is its ability to maintain ultra-low latency by dynamically adjusting the transmission rate in response to queueing delay. This is achieved through configuring the ECN marking thresholds, which allow the gNodeB to manage congestion by specifying the delay level at which the rate reduction should start. However, because L4S relies on congestion feedback through ECN CE marks, it cautiously increases its transmission rate following congestion events, which can result in reduced throughput. Additionally, the rate reductions triggered by the queueing delay thresholds further contribute to lower overall throughput.

In contrast, ANBR does not directly respond to queueing delay. Instead, it estimates the transmission and scheduling capabilities of the flow over a fixed window to get an optimal bitrate that reduces latency, packet loss and maximizes throughput. As a result of not making use of the queueing delays as compared to L4S, there is higher latency. However, the presence of more buffered data, leads to higher throughput.

Another key difference is in their deployment models. L4S congestion signaling operates end-to-end and relies on ECN markings in IP headers to indicate congestion, which requires that intermediate network nodes do not change the ECN bits along the transmission path. ANBR, on the other hand, mainly requires support between the RAN and the UE, which simplifies its deployment complexity. It was discovered during testing in the KPN test environment (as discussed in Chapter 5), that the KPN test infrastructure did not change these ECN bits, which shows that L4S can be deployed without modification to the existing transport network in KPN.

Furthermore, with the ECN threshold mechanism in L4S, the queueing delay at which rate reduction should occur can be specified by the network operator, and thus the end-to-end delay can be controlled more efficiently for different types of applications. ANBR, on the other hand, uses a windowing mechanism as a time interval for bitrate estimation, without direct control over the delay thresholds. This limits ANBR's ability to control latency as effec-

tively as L4S.

SQ3: Which application types or network scenarios stand to benefit most from the use of L4S and/or ANBR?

Applications that require ultra-low latency benefit most from L4S. These include use cases such as remote control, telemedicine and time-sensitive industrial automation. Enhanced Mobile Broadband (eMBB) applications with low-latency requirements, such as AR/VR, and cloud gaming can also see significant improvements with L4S. While L4S may lead to a slight reduction in throughput, this trade-off is negligible and well worth it for latency-sensitive applications, where timely delivery is as important as high throughput. It was also noted that when the ECN marking threshold was set to 60–100ms, ECN marking was rarely done, thus effectively disabling the L4S capability. This further highlights that L4S is particularly designed to deliver low-latency performance and performs best under lower threshold settings.

On the other hand, applications like high-definition video streaming, where consistent throughput is more critical than low latency, are better suited for ANBR. In such scenarios, buffering is acceptable, and although ANBR can reduce latency by indirectly lowering the queueing delay, it does not explicitly target specific queueing delay requirements and therefore has less control over latency compared to L4S. This makes ANBR less suitable for ultra-low latency applications but a good fit for throughput-sensitive use cases where occasional latency spikes can be tolerated.

SQ4: What recommendations and deployment strategies should be considered by mobile network operators, equipment vendors, and application developers to use L4S and ANBR more effectively?

As shown in the results and findings, both L4S and ANBR offer clear benefits. With the growing demand for latency-sensitive applications, application developers are increasingly focused on reducing latency, while network operators are seeking ways to deliver the best possible user experience. Quality of service is increasingly becoming a key differentiator for customers when choosing between network providers. For KPN, which aims to maintain its position as the leading network, it is therefore essential to explore strategies that ensure consistent and reliable QoS.

One might assume that with network slicing, QoS differentiation for various applications is already addressed, potentially reducing the need for additional mechanisms like L4S or ANBR. However, even with network slicing in place, L4S can provide latency and packet loss improvements, particularly during congestion. This makes it a valuable complement to network slicing.

In addition, L4S is seeing increasing adoption and support across the telecommunications ecosystem. Since both L4S and ANBR require end-to-end integration, from the application layer to the UE, it is more practical for KPN to focus on deploying L4S in the short term to realize immediate performance gains. In contrast, ANBR remains relatively immature and may take several years to become mature. Deploying ANBR at this stage would offer limited benefits, as most application developers have yet to incorporate or consider the support for it into their software.

In the case of deployment for L4S, the mobile network operator needs to carefully classify traffic and assign appropriate queue management thresholds. This involves configuring the ECN marking thresholds tailored for different application types. These thresholds can be aligned with the packet delay budgets (PDB) defined by 3GPP in the 5QI table in 3GPP TS 23.501 [8], thereby ensuring that queueing delays remain within acceptable limits and that the overall network delay stays below the PDBs. While lowering the thresholds may slightly impact throughput, the resulting latency improvements are worth it, particularly for low-latency eMBB applications as stated in SQ3. For example, AR applications mapped to 5QI 3 have a PDB of 10 ms, as defined in 3GPP TS 23.501. Assuming 2 ms is allocated for UPF-to-access delay, the remaining 8 ms must be maintained for the radio interface. Thus, the ECN thresholds can be set to enable the radio delay to stay way below this budget. By aligning L4S thresholds with these values, KPN can better control queueing in the RAN and improve latency for time-critical applications.

Another thing to note is that Ericsson already supports SMF features in the core network that enable the detection of L4S traffic and allow it to be steered into dedicated L4S QoS flows. It is also possible to assign a specific ECN threshold to a QoS flow and 5QI. When a threshold is assigned to a 5QI, all applications mapped to that 5QI will be treated using the same delay threshold settings. With this capability, KPN can steer applications into L4S QoS flows based on their latency requirements. Applications with similar QoS needs can be grouped under the same 5QI and assigned the same ECN threshold. The traffic steering becomes especially important when a single UE, using the same Data Network Name (DNN), has both L4S and non-L4S traffic. If these traffic types are not separated into different flows, they will be handled in the same queue. In such case, the non-L4S traffic could cause queueing delays that the L4S traffic is trying to avoid. Separating them into different flows ensures that L4S traffic maintains low-latency.

For equipment vendors and application developers, the full potential of ANBR can be realized by integrating RLC buffer information and queue delay metrics to determine appropriate bitrates so that latency can be further reduced. This approach needs further investigation in future studies. Additionally, the rate adaptive performance can further be enhanced by using hybrid strategies that combine L4S and ANBR. L4S increases its rate more cautiously. In such scenarios, ANBR can enable applications to increase their sending rate without significantly exceeding latency constraints.

6.2 Recommendations for future work

Evaluation of ANBR in the testlab: In this thesis, only L4S was evaluated in the test lab. Although ANBR was evaluated in the simulation environment and provided a representation of its performance based on the method applied in this work, it would be beneficial to perform evaluations in a real test network using KPN's equipment. This would allow a more reliable comparison between L4S and ANBR in practice, particularly given that ANBR's rate recommendation is vendor-specific.

Standardization of ANBR Algorithms: Future research should also explore the development of standardized algorithms that incorporate queueing delay as a key parameter in calculating the recommended sending rate for ANBR. One of the strengths of L4S is its use of queueing

delay for rate adjustment. Currently, ANBR lacks a standardized method for bitrate recommendation, and integrating queueing delay as a standard method for its input metric could significantly improve its rate adaptation capabilities. By monitoring queueing delay in real time, ANBR could adjust its recommendations to maintain low latency, comparable to L4S, while potentially achieving higher link utilization.

More efficient hybrid methods: In this research, the hybrid method selected the maximum of the L4S and ANBR rates whenever no ECN marking was observed, enabling a faster increase in the transmission rate. This approach led to periods of increased transmission rates, which in turn caused higher latency. Future work could explore alternative hybrid strategies that increase the rate more gradually based on both the L4S and ANBR rates, thus having a rate higher than L4S but lower than ANBR in such cases, to determine whether throughput can be improved beyond what standalone L4S provides while still maintaining the same latency.

ANBR and L4S for VoLTE and VoNR Services: Another relevant direction for future work is to investigate the impact of ANBR and L4S in VoLTE and VoNR. One study from the literature (Section 2.6) suggests that adaptive codec rate switching improves user experience, while another indicates that handover mechanisms provide better results at cell-edge scenarios. It is especially important to investigate this under heavy load conditions, where handover strategies may lose effectiveness if neighbouring cells are congested. In addition, the potential adaptation of L4S for VoLTE and VoNR remains unexplored. A focused comparison of L4S and ANBR in terms of packet loss improvement, latency reduction, voice quality and overall user experience for these IMS services would provide valuable insights into how these mechanisms could complement or substitute each other in real-world deployments.

Coexistence of L4S with TCP Optimization: TCP Optimization, discussed in Section 3.1, shares similar goals with L4S, as both use network-aware congestion information rather than relying solely on packet loss to adjust their sending rate. Furthermore, TCP optimization can accelerate the slow-start phase, improving the ramp-up of the transmission rate and overall bandwidth utilization. Traditional TCP congestion control like TCP Reno and TCP Cubic reduce their rates by a fixed percentage after congestion, whereas TCP Prague (the congestion control algorithm for L4S) dynamically adjusts its rate in proportion to the actual congestion, thus enhancing bandwidth utilization. Future work could investigate whether L4S alone is sufficient for efficient rate adaptation, potentially eliminating the need for TCP optimization and the network buffering. However, an advantage of TCP optimization is that it handles retransmissions locally, allowing faster recovery from packet losses. It also enables centralized congestion control, irrespective of the specific congestion control algorithms used by different application servers. Combining TCP Prague with TCP Optimization could therefore provide additional benefits by taking advantage of the strengths of both approaches. Further research could explore this combination in more detail.

Use of Artificial Intelligence and Machine Learning: In addition, the application of artificial intelligence (AI) and machine learning (ML) techniques in the RAN could be used to predict network capacity changes and proactively adjust bitrate recommendations. The predictive models could improve ANBR's responsiveness and the sender can adjust its rate more quickly before congestion occurs.

Impact of Packet Loss, queueing Delay, and Buffer Size: Another important area for inves-

tigation is the relationship between packet loss, queueing delay, and buffer size. Packet loss often results from increased queueing delay in the presence of small buffers. Therefore, future testbed evaluations should include experiments with reduced buffer sizes to assess the impact of L4S and ANBR on packet loss and overall performance based on these buffer sizes.

References

[1] "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond," International Telecommunication Union - Radiocommunication Sector (ITU-R), Tech. Rep. Recommendation ITU-R M.2083-0, 2015. [Online]. Available: https://www.itu.int/rec/R-REC-M.2083-0-201509-I/en.

[2] M. R. Leitner. "On link modeling, network emulation and its impacts on applications." Accessed: 2025-05-25. (Aug. 2017), [Online]. Available: https://developers.redhat.com/blog/2017/08/31/on-link-modeling-network-emulation-and-its-impacts-on-applications.

[3] P. Willars, E. Wittenmark, H. Ronkainen, et al., "Enabling Time-Critical Applications over 5G with Rate Adaptation," Ericsson (with Deutsche Telekom), White Paper, May 2021, Available online: https://www.ericsson.com/en/reports-and-papers/white-papers/enabling-time-critical-applications-over-5g-with-rate-adaptation.

[4] D. Brunello, I. Johansson, M. Ozger, and C. Cavdar, "Low Latency Low Loss Scalable Throughput in 5G Networks," in 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 2021, pp. 1–7. DOI: 10.1109/VTC2021-Spring51267.2021.9448764.

[5] B. Briscoe, K. D. Schepper, M. Bagnulo, and G. White, Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture, IETF RFC 9330, Jan. 2023. DOI: 10.17487/RFC9330. [Online]. Available: https://www.rfc-editor.org/info/rfc9330.

[6] K. D. Schepper and B. Briscoe, The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S), RFC 9331, Jan. 2023. DOI: 10.17487/RFC9331. [Online]. Available: https://www.rfc-editor.org/info/rfc9331.

[7] K. D. Schepper, B. Briscoe, and G. White, Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S), RFC 9332, Jan. 2023. DOI: 10.17487/RFC9332. [Online]. Available: https://www.rfc-editor.org/info/rfc9332.

[8] "3GPP TS 23.501 V18.5.0: System Architecture for the 5G System (Release 18)," 3rd Generation Partnership Project (3GPP), Technical Specification TS 23.501, Nov. 2024.

[9] K. D. Schepper, O. Tilmans, B. Briscoe, and V. Goel, "Prague Congestion Control," Internet Engineering Task Force, Internet-Draft draft-briscoe-iccrg-prague-congestion-control-04, Jul. 2024, Work in Progress, 34 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-briscoe-iccrg-prague-congestion-control/04/.

[10] L4S Team, UDP Prague: UDP Congestion Control Algorithm for L4S, https://github.com/L4STeam/udp_prague, Accessed: 1 June 2025, 2024.

[11] I. Johansson, M. Westerlund, and M. Kühlewind, "Self-Clocked Rate Adaptation for Multimedia," Internet Engineering Task Force, Internet-Draft draft-johansson-ccwg-rfc8298bis-screamv2-03, Mar. 2025, Work in Progress, 34 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-johansson-ccwg-rfc8298bis-screamv2/03/.

[12] "TS 26.114: IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction," Technical Specification 3GPP TS 26.114 version 18.6.0 Release 18, 2024.

[13] 3GPP, "TS 26.510 V18.1.0: 5G; Media delivery; Interactions and APIs for provisioning and media session handling," 3rd Generation Partnership Project (3GPP), Tech. Rep., Oct. 2024.

[14] Apple Developer, Testing and Debugging L4S in Your App, https://developer.apple.com/documentation/Network/testing-and-debugging-l4s-in-your-app, 2024.

[15] Comcast Kicks Off Industry's First Low Latency DOCSIS Field Trials, 2023. [Online]. Available: https://corporate.comcast.com/stories/comcast-kicks-off-industrys-first-low-latency-docsis-field-trials.

[16] Apple Inc. "Reduce network delays with L4S." WWDC23 Session 10004. (2023), [Online]. Available: https://developer.apple.com/videos/play/wwdc2023/10004/ (visited on 05/30/2025).

[17] What is the L4S Setting in the GeForce NOW Streaming Quality Menu? 2023. [Online]. Available: https://nvidia.custhelp.com/app/answers/detail/a_id/5522/~/what-is-the-l4s-setting-in-the-geforce-now-streaming-quality-menu%3F.

[18] Nokia, L4S – Low Latency, Low Loss, and Scalable Throughput : Enabling large-scale deployments of low-latency services, 2023. [Online]. Available: https://www.nokia.com/asset/213410.

[19] Nokia and Vodafone Conduct World's First Trial of L4S Technology over an End-to-End PON Network, 2024. [Online]. Available: https://www.nokia.com/about-us/news/releases/2024/04/03/nokia-and-vodafone-conduct-worlds-first-trial-of-l4s-technology-over-an-end-to-end-pon-network/.

[20] Elisa Corporation and Nokia Corporation. "Elisa and Nokia first in the Nordics to showcase 5G-Advanced L4S technology for real-time applications in congested network environments." Press release. (Feb. 2024), [Online]. Available: https://elisa.com/corporate/news-room/press-releases/elisa-and-nokia-first-in-the-nordics-to-showcase-5g-advanced-l4s-technology-for-real-time-applications-in-congested-network-environments/31899461721279/.

[21] Ericsson and e& UAE First to Implement 5G Advanced Time-Critical Communication Solution in Middle East and Africa, 2024. [Online]. Available: https:

//www.ericsson.com/en/press-releases/5/2024/ericsson-and-e-uae-first-to-implement-5g-advanced-time-critical-communication-solution-in-middle-east-and-africa.

[22] G. Pan, S. Xu, and P. Jiang, "Optimizing 5G-Advanced Networks for Time-Critical Applications: The Role of L4S," IEEE Wireless Communications, pp. 1–8, 2024.

[23] J. Son, Y. Sanchez, C. Hampe, D. Schnieders, T. Schierl, and C. Hellge, "L4S Congestion Control Algorithm for Interactive Low Latency Applications over 5G," in 2023 IEEE International Conference on Multimedia and Expo (ICME), 2023, pp. 1002–1007.

[24] A. Srivastava, F. Fund, and S. S. Panwar, "An Experimental Evaluation of Low Latency Congestion Control for mmWave Links," in IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 352–357.

[25] L. V. Monteiro, V. S. Simão, R. B. Lira, L. C. Almeida, R. D. Gomes, and P. D. Maciel Jr., "L4S in Private 5G Industrial Networks: A Case Study for Real-Time Video Transmission in Programmable Networks," in IEEE NFV-SDN, 2024.

[26] 3GPP, "TS 36.300 V18.1.0: LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), Tech. Rep., May 2024.

[27] 3GPP, "TR 26.919 V16.2.0: 5G; Study on media handling aspects of conversational services in 5G systems," 3rd Generation Partnership Project (3GPP), Tech. Rep., Nov. 2020.

[28] 3GPP, "TS 36.321 V18.1.0: LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), Tech. Rep., May 2024.

[29] 3GPP, "TS 38.321 V18.2.0: 5G; NR; Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), Tech. Rep., Aug. 2024.

[30] 3GPP, "TS 26.501: 5G; 5G Media Streaming (5GMS); General description and architecture," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2024.

[31] MediaTek, 5G NR Voice Solutions Overview and Deployment Guidelines, 2021. [Online]. Available: https://newsletter.mediatek.com/hubfs/MediaTek-5G-Voice-Solutions-Whitepaper-PDF5GNRSWP-0821.pdf.

[32] 5G Media Action Group, 5G Media Streaming Architecture, https://5g-mag.github.io/Getting-Started/pages/5g-media-streaming/, Accessed: 2024-05-31, 2023.

[33] A. Prasad, S. Bhatia, L. Duan, et al., "Enhanced Voice Services Based VoLTE Rate Adaptation Mechanism to Improve Quality of Experience," in 2019 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2019, pp. 1–6.

[34] S. Xu, Y. Fu, F. Li, H. Zhang, and J. Xin, "Impact of Jitter and Packet Loss on Enhanced Voice Services for Voice over NR," in Proceedings of the 7th International Conference on Control Engineering and Artificial Intelligence (CCEAI), ACM, 2023, pp. 149–156. DOI: 10.1145/3580219.3580246.

[35] J. Karjee, S. Khatter, D. Sarkar, H. L. C. Tammineedi, and A. K. R. Chavva, "5G-NR Cross Layer Rate Adaptation for VoIP and Foreground/Background Applications in UE," in 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), IEEE, 2020, pp. 1–7.

[36] L. L. Peterson and B. S. Davie, Computer Networks: A Systems Approach, 6th ed. San Francisco, CA, USA: Morgan Kaufmann, 2021, ISBN: 9780128182000.

[37] B. Bojovic, S. Lagen, K. Koutlia, X. Zhang, P. Wang, and Q. Qu, "Enhancing 5G QoS Management for XR Traffic Through XR Loopback Mechanism," IEEE Journal on Selected Areas in Communications, vol. 41, no. 6, pp. 1769–1782, 2023.

[38] Sandvine, "TCP Optimization: Opportunities, Key Performance Indicators, and Considerations," Sandvine, Whitepaper, Jul. 2019. [Online]. Available: https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/Whitepapers/Sandvine_WP_TCP%20Optimization%2020190701.pdf.

[39] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, and C. Mulligan, 5G Core Networks: Powering Digitalization. Academic Press, 2018, pp. 74–96, ISBN: 978-0-12-814323-0.

[40] 3GPP, 5G System Overview, urlhttps://www.3gpp.org/technologies/5g-system-overview, Accessed: 2025-05-29, 2024.

[41] E. Dahlman, S. Parkvall, and J. Skold, 5G NR: The Next Generation Wireless Access Technology, 2nd. Academic Press, 2020, ISBN: 9780128223253.

[42] S. Floyd, D. K. K. Ramakrishnan, and D. L. Black, The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168, Sep. 2001. DOI: 10.17487/RFC3168. [Online]. Available: https://www.rfc-editor.org/info/rfc3168.

[43] B. Briscoe, M. Kühlewind, and R. Scheffenegger, "More Accurate Explicit Congestion Notification (AccECN) Feedback in TCP," Internet Engineering Task Force, Internet-Draft draft-ietf-tcpm-accurate-ecn-34, Mar. 2025, Work in Progress, 73 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tcpm-accurate-ecn/34/.

[44] Z. Sarker, C. Perkins, V. Singh, and M. A. Ramalho, RTP Control Protocol (RTCP) Feedback for Congestion Control, RFC 8888, Jan. 2021. DOI: 10.17487/RFC8888. [Online]. Available: https://www.rfc-editor.org/info/rfc8888.

[45] 3GPP, "Study on XR (Extended Reality) and Media Services (Release 18)," 3rd Generation Partnership Project, Tech. Rep. TR 23.700-60 V18.0.0, 2022.

[46] 3GPP, "TS 38.331 V18.1.0: NR; Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification 38.331, version 18.1.0, May 2024.

[47] "TS 27.007: AT Command Set for User Equipment (UE)," Technical Specification 3GPP TS 27.007 Release 18, 2024.

[48] Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), 5G-LENA: The 5G NR module for the ns-3 simulator, Accessed: 2025-07-28, 2025. [Online]. Available: https://5g-lena.cttc.es/.

[49] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, 3GPP Technical Report TR 38.901, version 17.0.0, 2022, Accessed: 2025-07-28. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173.

[50] B. Bojović, S. Lagén, K. Koutlia, X. Zhang, P. Wang, and L. Yu, "Enhancing 5G QoS Management for XR Traffic Through XR Loopback Mechanism," IEEE Journal on Selected Areas in Communications, vol. 41, no. 6, pp. 1772–1786, 2023. DOI: 10.1109/JSAC.2023.3273701.

[51] 3GPP, "Study on XR (Extended Reality) evaluations for NR," 3GPP, Technical Report TR 38.838, version 17.0.0, 2022, Accessed: 2025-07-28. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3736.

[52] J. D. Chimeh, M. Hakkak, and P. Azmi, "Internet Traffic Modeling and Capacity Evaluation in UMTS," International Journal of Hybrid Information Technology, vol. 1, no. 2, pp. 109–117, 2008.

[53] L. Peterson, L. Brakmo, and B. Davie, TCP Congestion Control: A Systems Approach. Systems Approach LLC, 2022, Creative Commons CC BY-NC-ND 4.0 license; source available on GitHub.

[54] 3GPP, "TS 38.300 V18.1.0: 5G; NR; NR and NG-RAN Overall description; Stage-2," 3rd Generation Partnership Project (3GPP), Tech. Rep., May 2024.

A

Definitions

Active Queue Management (AQM): A mechanism used in routers and switches to proactively manage congestion by detecting early signs of buffer build-up and signaling congestion before queues overflow. Unlike tail drop, which reacts only when buffers are full, AQM operates by measuring queue length or delay and generating early congestion signals through packet drops or ECN marking, resulting in lower latency and reduced packet loss [53].

Access Network Bitrate Recommendation (ANBR): A method where the RAN monitors network conditions and recommends optimal bitrates to the UE using Media Access Control Control Element (MAC CE). The UE relays these recommendations to the application server, enabling rate adaptation [12],[54].

Congestion Window (cwnd): The maximum amount of unacknowledged data that can be sent into the network at any time. It regulates the sender's transmission rate and adapts dynamically based on detected network congestion [53].

Encoding: The process of converting raw video data into a compressed digital format suitable for transmission, storage, and playback. At the receiver side, decoding reconstructs the video for playback [36].

Explicit Congestion Notification (ECN): A congestion signaling mechanism that marks packets instead of dropping them. ECN uses two bits in the IP header to indicate congestion: Not-ECT (non-ECN capable), ECT(0) (classic ECN), ECT(1) (L4S traffic), and CE (Congestion Experienced), which signals congestion to the sender without packet loss [53].

Low Latency, Low Loss, Scalable Throughput (L4S): A network mechanism standardized by the IETF (RFC 9330-9332) and 3GPP Release 18. It uses ECN with AQM mechanisms to provide early congestion signals, enabling low queueing delays and high throughput. Senders respond by using scalable congestion control algorithms like TCP Prague, UDP Prague, or SCReAM [5].

MAC Control Element (MAC CE): A signaling message in mobile networks that communicates recommended bitrates from the gNodeB or eNodeB to the UE to support ANBR[26],[54].

Prague Congestion Control: A scalable congestion control algorithm designed to be L4S-compliant. It reacts proportionally to the amount of ECN feedback it receives to maintain low latency and high throughput. Variants include TCP Prague (for TCP), UDP Prague (for UDP-based real-time applications), and L4S-compatible SCReAM [9].

Smoothed Round Trip Time (sRTT): An exponentially weighted moving average of measured RTT values. It is used by congestion control algorithms to track network latency trends while filtering out short-term fluctuations [9].

Transmission Control Protocol (TCP): A reliable, connection-oriented transport protocol that ensures in-order delivery of data with error checking and retransmissions. TCP is suitable for applications like file transfers or video-on-demand where reliability is prioritized over latency [36].

User Datagram Protocol (UDP): A connectionless protocol that transmits data without guarantees for reliability or ordering. It is ideal for delay-sensitive real-time applications and the application handles it's congestion control [36].

Virtual RTT (vRTT): A way to mitigate RTT bias in Prague congestion control. It sets a lower bound (e.g., 25 ms) for RTT in certain prague calculations, ensuring fairness between low-RTT Prague flows and classic congestion control flows [9].

B

Throughput and latency of the hybrid and L4S methods

Hybrid and L4S methods (5-7ms threshold)

This section presents the throughput and latency charts for the hybrid and L4S schemes, focusing on the 5–7 ms threshold. As discussed in the report, the goal of the hybrid method is to leverage ANBR to increase the transmitted rate. However, this increase in throughput often comes at the cost of higher latency.

As illustrated in Figures B.1 and B.2, latency spikes correspond to periods where the Hybrid method achieves higher throughput. These plots highlight the trade-off between throughput improvement and latency increase when utilizing the hybrid approach.

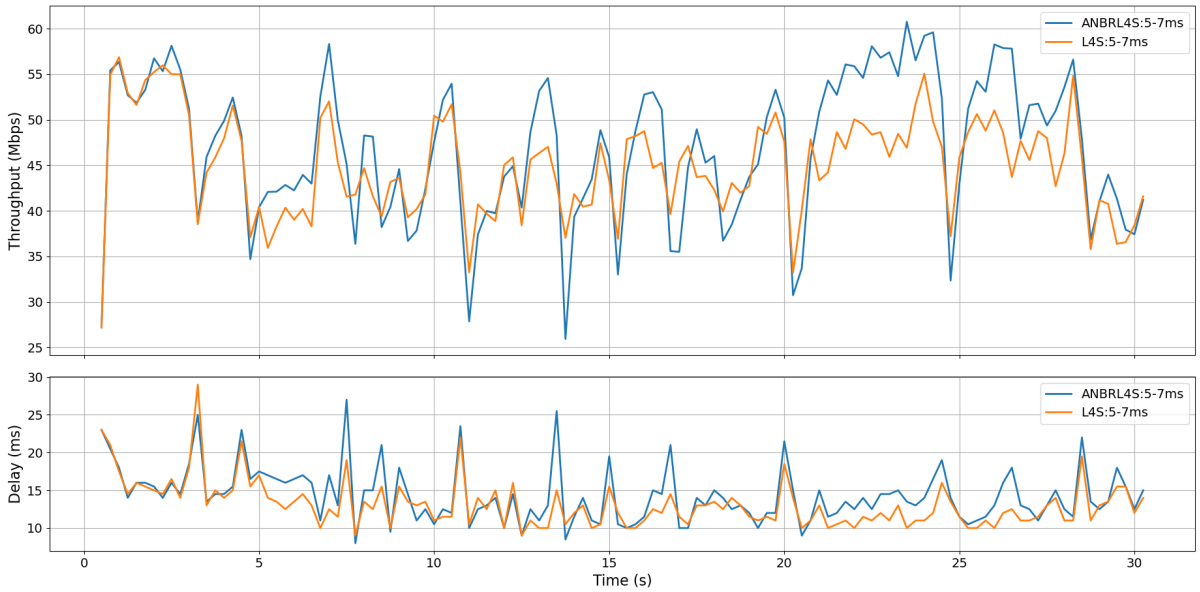


Figure B.1: Throughput and latency of the Hybrid and L4S methods for the 5–7 ms threshold under the lower background traffic.

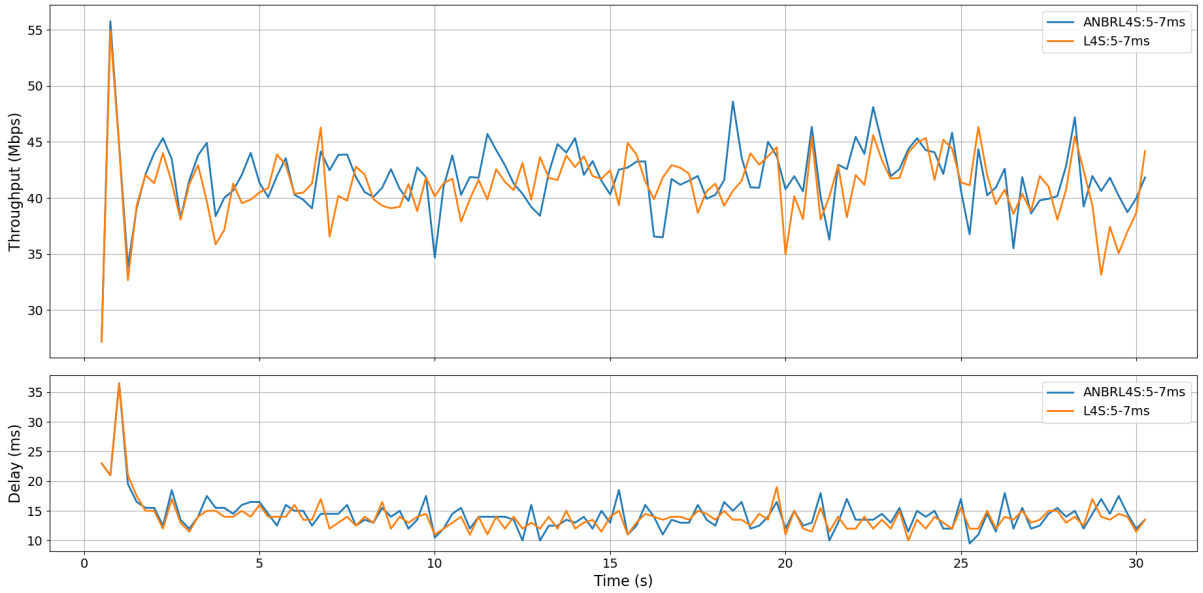
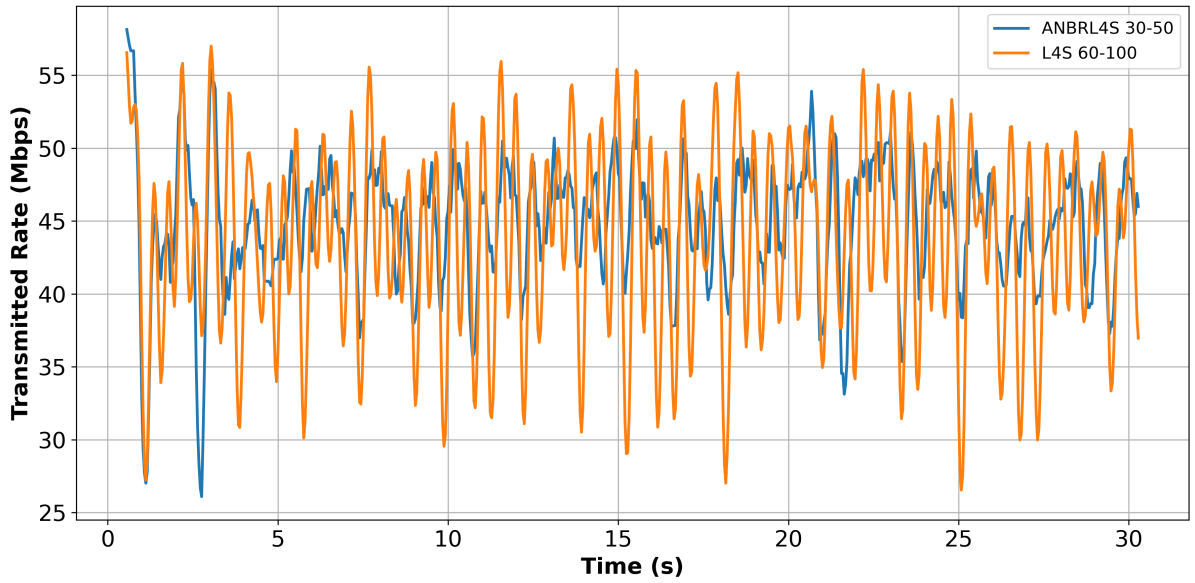


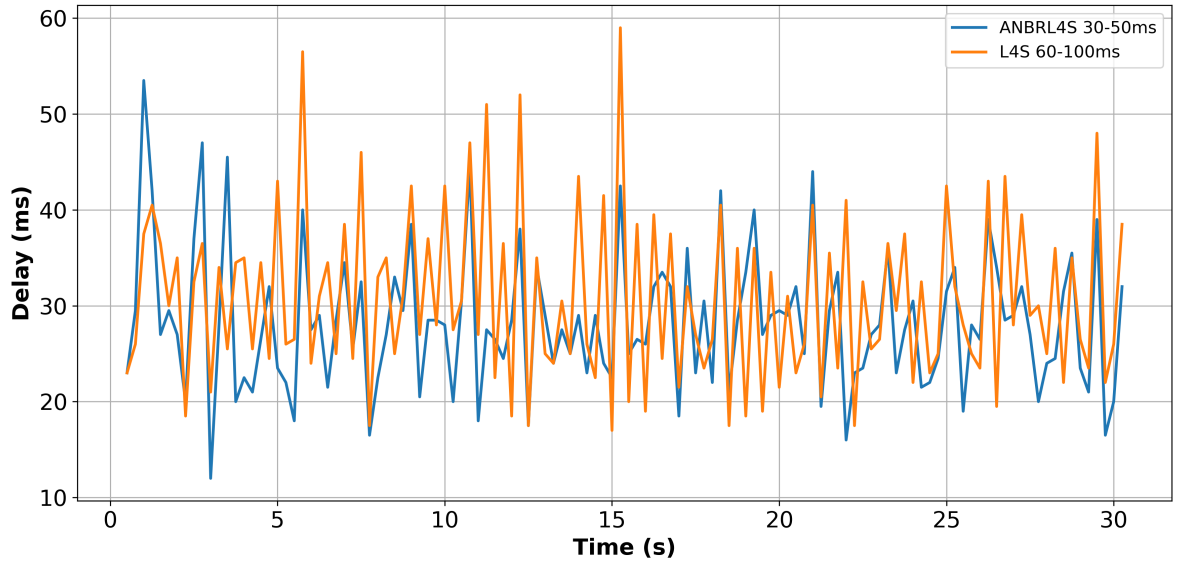
Figure B.2: Throughput and latency of the Hybrid and L4S methods for the 5–7 ms threshold under the higher background traffic

Hybrid method (30–50 ms) and L4S method (60–100 ms) thresholds

This section presents the transmitted rate and one-way delay for the L4S method with a 60–100 ms threshold and the hybrid method with a 30–50 ms threshold for the higher background load scenario. As discussed in Section 5.1.2, the hybrid method (30-50ms) achieves lower latency and higher throughput compared to L4S (30-50ms) for the high background load scenario. In the 60–100 ms case, the L4S method shows a highly variable transmitted rate (Figure B.3a), driven by large RTT fluctuations (see Section 5.1.4). Since L4S at this threshold bases its rate calculations on RTT rather than ECN marks, it often transmits at excessively high rates. This then produces latency spikes (Figure B.3b). As a result of the large variations of the transmitted rate, the average transmitted rate of L4S at 60–100 ms is lower and its latency higher than that of the hybrid method at 30–50 ms, leading to reduced throughput and increased delay in the L4S case.



(a) Transmission rate



(b) One-way delay

Figure B.3: Transmission rate and one-way delay for the L4S method (60–100 ms) and the hybrid method (30–50 ms) under the high background load