Active Queue Management as Quality of Service Enabler for 5G Networks

Mikel Irazabal, Elena Lopez-Aguilera and Ilker Demirkol
Dept. of Network Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain
Email: mikel.irazabal@upc.edu, elopez@entel.upc.edu, ilker.demirkol@entel.upc.edu

***Abstract*—5G is envisioned as the key technology for guaranteeing low-latency wireless services. Packets will be marked with QoS Flow Indicators (QFI) for different forwarding treatment. 3GPP defines the end-to-end delay limits, but leaves the QoS provisioning methods as implementation dependent. Different services with different constraints will inevitably share queues at some network entity. On the one hand, maintaining the shared queues uncongested will guarantee a rapid packet delivery to the subsequent entity. A brief sojourn time is indispensable for an on time low-latency priority traffic delivery. On the other hand, if shared queues are maintained undersized, throughput will be squandered. In this paper, we propose the use of AQM techniques in 5G networks to guarantee delay limits of QoS flows. Through the evaluation of realistic delay-sensitive and background traffic, we compare different possible solutions. We show that AQM mechanisms together with limited queues, maintain the system uncongested, which reduces drastically the delay, while effectively achieving the maximum possible throughput.**

I. Introduction

A crucial challenge for achieving a deterministic delay in 5G networks is the latency that is incurred by the large queues of the network entities. This problem, known as bufferbloat [1], happens in current cellular systems since the Radio Access Network (RAN) employs large buffers to compensate the capacity variance of the radio physical channel. This conservative but usual approach, creates unnecessary delays for traffic flows that share the same buffer. However, since in the 5G there will be services mapped to the same QoS class sharing the same queues, it is critical to have a method that ensures the required delay, while achieving the maximum possible throughput.

Although there are Active Queue Management (AQM) algorithms such as CAKE, FQ-CoDel or CoDel that target to reduce the delay on bottleneck links, their applicability in 5G networks has not been deeply studied before. In this paper, we study the use of CoDel within the 5G domain at different entities and layers in order to fulfill the Quality of Service (QoS) requirements of delay-sensitive services. With this aim, we assess the benefits of using CoDel at the newly defined Service Data Adaptation Protocol (SDAP) layer at the 5G Access Network (5G-AN), which does the mapping of QoS classes to Data Radio Bearers (DRB), and the benefits of the use of CoDel at lower layers in combination with restricted buffer sizes.

We evaluate different implementation scenarios with delay-sensitive traffic generated with the parameter values of real network traces [2]. In the evaluations, such traffic competes with a bulky TCP traffic for the network resources. Experimental results expose the benefits that AQM brings into delay-sensitive traffic, and corroborate that AQM mechanisms will be key enablers to guarantee the QoS criteria defined by 3GPP.

II. Related work

The softwarization process [3] of telecommunication networks is leading to new and heterogeneous business models with different constraints and challenges. Thus, the QoS scenario for the 5G standalone network proposed by the 3GPP is challenging. Even though slicing has emerged as the correct tool for virtualizing the 5G stack, the QoS problems associated with each slice will remain if slices are required to provide more than one service. Different services with different QoS constraints that share resources will have to be segregated in order to guarantee delivery rate and latency. In 5G, even though this aspect will be crucial as business enabler, especially for ultra-reliable low-latency communications, no substantial effort has been invested to mitigate this problem.

Due to the low memory prices, routers are deployed with large buffers that can hold several megabytes of data, which can introduce delays in the order of tens of seconds [1]. This completely distorts TCP's congestion control algorithm feedback and, thus, nullifies its ability to quickly adapt the transmission rate to the data link capacity. Therefore, TCP creates large buffers that cause important packet sojourn times. In order to tackle this problem in the Internet routers, AQM has been employed. A natural deployment for AQM mechanisms in 5G is the Radio Link Control (RLC) layer where data is buffered, segmented, reordered and transmitted to the following layers [4]. At [5], a modified version of the RED algorithm [6] at RAN's Layer 2 RLC entity is proposed. RED considers the growing rate of the queue as a congestion symptom and increases the probability of discarding a packet accordingly. While persistent queues indicate congestion, the growing rate of a queue does not. The bursty traffic nature of concurrent TCP sources can grow and shrink the queues before RED can effectively react accordingly [7]. Thus, the RED algorithm needs some tuning and can conceivably cause problems if it is implemented without a tedious study of the traffic patterns. Therefore, the RED algorithm was never widely implemented [8].

Segregating the traffic correctly before the scheduler is also crucial. Priority traffic can be firstly scheduled avoiding the large sojourn time that may occur if priority traffic has to share the queue with bulky traffic. One of the first network algorithms that addressed such a problem is the Stochastic Fair Queuing (SFQ) [9]. Flows are hashed and assigned to different queues. Every active queue is assigned an equal egress rate in a Round Robin manner. However, due to the hashing nature, two flows can end sharing a queue, splitting each flow's theoretical corresponding share of bandwidth. This situation is partly alleviated by periodically adding a perturbing value to the hash function that rehashes the flows, thus reducing the possibility of different flows sharing the same queues for large periods.

This method has been explored by [10] with a SFQ mechanism implemented at the Packet Data Convergence Protocol (PDCP) entity. The PDCP entity is responsible for header compression, ciphering and in-sequence delivery among other tasks. This approach segregates the traffic that has already been aggregated into a QoS Flow Indicator (QFI) in order to fairly distribute the egress rate between different 5-tuple flows. QFI is a scalar that is used as the finest granularity reference to a specific QoS forwarding behaviour (e.g., scheduling prioritization, queue management, packet loss rate, packet delay budget). All the traffic mapped into a given QFI must experience the same forwarding treatment according to [11]. Therefore, segregating the traffic from a QFI is a non-3GPP compliant technique. At [10], the possibility of implementing a communication mechanism between the RLC and the PDCP is also explored, in order to maintain the buffers at RLC in an optimal size.

Some industrial brute force approaches for 5G Non-Standalone scenarios have been deployed by reserving enough resources to guarantee certain bit rates for high priority data [12]. This solution implies that high priority traffic will not yield resources to other flows potentially underutilizing them. Such a trivial solution can only last as a transitory solution due to its lack of scalability and efficiency.

Some other more interesting approaches [13] explore the possibility of implementing AQM algorithms that rely on packet sojourn time, specifically the CoDel algorithm. They use it in combination with a modified Round Robin scheduler that segregates the traffic in different queues known as Deficit Round Robin [14]. The combination of both is known as the FlowQueue-CoDel Packet Scheduler (FQ-CoDel) [15] and has become the "de facto" standard in different embedded routing open source projects [16]. At [13], the Round Trip Time (RTT) of the packet is measured and the egress rate of the UPF entity is adapted accordingly. The egress rate is constraint to the maximum bandwidth of the link. This ensures that the packet accumulation will happen at the UPF queues rather than at the 5G-AN entities. This approach presents several problems. In the first place, the 5G networks do dynamically and abruptly change their bandwidth due to its dependence with the radio channel conditions. If more bandwidth is available, bandwidth will be squandered as the egress rate control mechanism depends on the feedback from the RTT and needs some time to adapt correctly. Moreover, the UPF can reside relatively far from the 5G-AN, which will increase the response time due to bandwidth variability. Secondly, this approach relies on protocols that send some feedback to the sender. While most of the 5G traffic will certainly be implemented in such a manner, low-latency time constraint traffic may not rely on a feedback from the transport layer (e.g., QUIC, SCTP).

III. 5G QoS Scenario and the Proposed QoS Provisioning Solution

In order to tackle the QoS problem in 5G networks, we consider a full 5G QoS scenario as the one shown in Fig. 1.

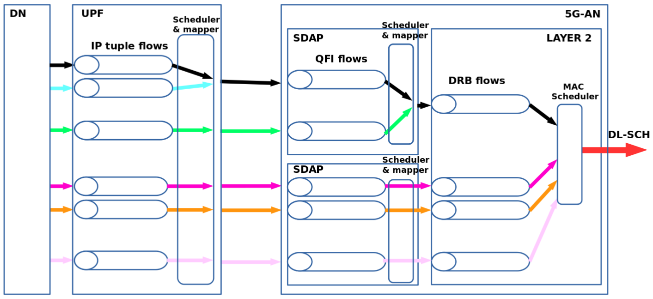

Fig. 1. 5G QoS scenario

Although the presented QoS scenario describes a downlink scenario, similar SDAP and DRB mappings are also present in the uplink scenario.

The data packets arrive from the data network (DN) to the UPF. These packets are firstly enqueued and then mapped to QFI flows according to Packet Detection Rules (PDR) [11]. Once they arrive to the 5G-AN, these data packets are handled by the SDAP [17], which is responsible for mapping the QFI flows into DRB flows. Different SDAP entities can coexist for a UE, since an SDAP entity is instantiated per Protocol Data Unit (PDU) session. Finally, the MAC scheduler is responsible to deliver every Transmission Time Interval (TTI) the data quantity requested by the physical layer (PHY), through the Downlink Shared Channel (DL-SCH) transport channel.

Maximizing the throughput while prioritizing the packets and reducing the latency is a complex task. On the one hand, if traffic with high priority arrives, it is desirable to forward it as soon as possible to the DL-SCH transport channel. Once packets are aggregated into a flow, they cannot be segregated again [11]. Therefore, if a high priority packet is forwarded to a congested queue, the packet will suffer a big sojourn time until the queue is emptied. Hence, it would be advisable to maintain the buffers as empty as possible. On the other hand, for each TTI, the MAC scheduler should send as many data through the DL-SCH as requested by the PHY entity in order not to squander any transmission possibility. Otherwise the throughput will be reduced. Hence, it would be advisable to maintain the buffers as full as possible. In addition to the problem described above, the number of packets required by the PHY entity changes dynamically due to diverse factors (e.g., radio channel conditions, HARQ retransmissions).

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

In order to tackle the aforementioned problems, we explore the following solutions. In the first place, we implement the CoDel AQM algorithm [18]. CoDel operates with an *interval time* parameter. If within this interval time all the packets' sojourn time is above a given *target time* parameter, this indicates a congestion state, and the following packet is dropped. This drop notifies the sender that excessive buffering is happening. In this case, the *interval time* is divided by $\sqrt{x}$, where x starts from 2 and is incremented by 1 for each consecutive drop, i.e., for each consecutive interval time with congestion state, the interval time is reduced. If, however, an interval time without congestion state occurs, the interval time is reset. In this way, CoDel adapts efficiently to abrupt changes in the egress rate, which makes it a good candidate for 5G networks.

In the second place, we propose to maintain DRB queues on 5G QoS scenario limited to values slightly above the order of magnitude of the maximum possible egress rate from the MAC scheduler. We do not study the values below that rate, as it would just sacrifice throughput. This principle is well known in other disciplines that have to deal with queues that are formed in the lower layers. Network Interface Controller (NIC) software developers vary the queue limits according to the egress rate in order to avoid large sojourn times at the network card without squandering transmission possibilities [19].

IV. Evaluation Framework

In order to evaluate our proposed AQM based solution and compare it with the baseline solutions, we implement a queue system that emulates different 5G entities and their queues presented in Figure 1. As per QoS traffic, we define a delay-sensitive traffic flow, taking gaming application as a reference [2]. For this, we configure the well-known *ping* tool with a realistic gaming traffic packet size of 100 bytes and an interval that varies from 10 to 70 ms in increments of 10 ms in line with [2]. As background traffic, we use a second flow of TCP, generated by the *iperf3* software. We run our experiments for 30 seconds for each *ping* interval.

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

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

We classify the flows according to their source IP address/port number, destination IP address/port number and the protocol in use, known as the 5-tuple. These values are hashed with the Jenkins hash function and classified into IP tuple flows. A mapper at UPF, multiplexes the IP tuple flows into QFI flows. We implement a SFQ [9] as the UPF scheduler where 10 IP packets are egressed every 1 ms. We enhance the SDAP [17] capabilities from mapping to scheduling and mapping. We implement the SDAP scheduler as a Round Robin scheduler where 10 packets are egressed fairly among active queues every 1 ms. The QFI flows are mapped into DRB flows by the SDAP entity. Finally, the MAC scheduler egresses 10 packets fairly among the active DRB flows every 10 ms for a theoretical maximum throughput of 11.68 Mbps considering a MTU of 1500 bytes and excluding the TCP/IP headers. Once a packet is egressed from the MAC scheduler, it is forwarded to the kernel space with a *forward* verdict. When an AQM mechanism decides to drop a packet, the *discard* verdict is passed to the NFQUEUE that informs the kernel space to drop the packet.

CoDel is well known as a knob-less QoS solution. It is governed by the two aforementioned variables, the *interval time* and the *target time*. At [18] the target time is recommended to be set at around 5% of the proposed interval time of 100 ms. Under our test conditions, CoDel would classify all the packets into the dropping state with direct consequences for the bandwidth [20], since the MAC scheduler forwards 10 packets every 10 ms in discrete time. With the default CoDel parameter values, all the packets would be dropped in our scenario. Hence, we increased the target time to 15 ms and the interval time to 300 ms, while meeting the requirements of setting the target time close to the RTT. This value has been heuristically proven to be correct for the current scenario.

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

We evaluate four different solutions within these scenarios. In the first solution, which is similar to the default one used in the current cellular systems, buffers are unlimited and no

AQM mechanism is implemented. In the second solution, the DRB buffer capacity is limited. The SDAP scheduler does not forward any packet that would surpass the DRB limited buffer capacity and no AQM mechanism is implemented. In the third solution, CoDel is implemented at the DRB queue without any buffer limitation. Finally, our proposal of using CoDel AQM for the QFI queue and limiting the DRB queue capacity is evaluated. In Fig. 2, we depict the key components of our proposed solution as a flowchart.

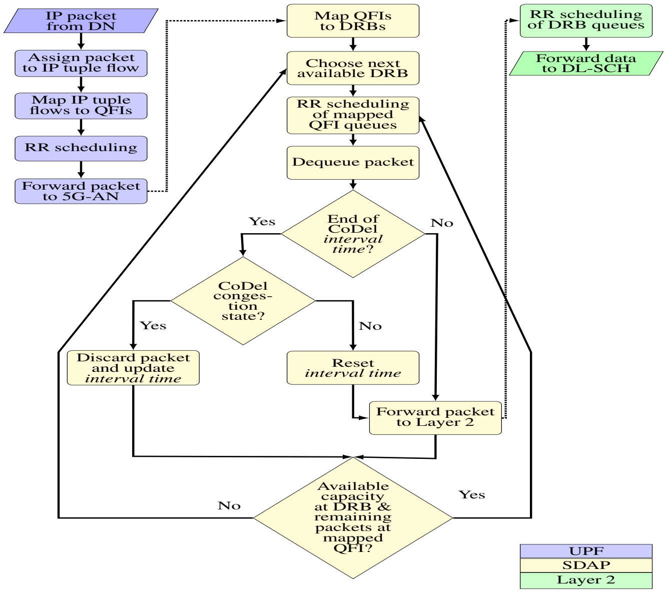
IP packet from DN
Assign packet to IP tuple flow
Map IP tuple flows to QFIs
RR scheduling
Forward packet to 5G-AN
Map QFIs to DRBs
Choose next available DRB
RR scheduling of mapped QFI queues
Dequeue packet
End of CoDel *interval time*?
Yes
No
CoDel congestion state?
Yes
No
Discard packet and update *interval time*
Reset *interval time*
Forward packet to Layer 2
Available capacity at DRB & remaining packets at mapped QFI?
No
Yes
RR scheduling of DRB queues
Forward data to DL-SCH
UPF
SDAP
Layer 2

Fig. 2. Flowchart of the proposed solution with CoDel at SDAP layer and DRB queue size limitation.

V. Experimental results

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

The experimental results corresponding to the first scenario can be seen in Figs. 3 – 5. The first case corresponds to the conventional solution of not limiting the buffers. Since the buffers are not limited, the packets are forwarded to the DRB buffer, where they accumulate. There are always enough packets at the DRB to fulfill the maximum egress rate and, therefore, no bandwidth is squandered. However, the delay-sensitive traffic suffers from important delays, since the DRB queue presents a large occupancy when the delay-sensitive traffic packet is enqueued.

The second, third and fourth cases correspond to the solution of only limiting the DRB buffer size. With this aim, the DRB buffer is limited to 10, 20 and 30 packets, respectively. As it can be seen from Fig. 3, the packets accumulate at the QFI queue since the SDAP entity does not forward more packets to the DRB queue once its buffer limit has been reached. However, the total number of packets in the system remains constant in the three cases. The throughput is maintained as well as the delay, as observed from Figs. 4 and 5. The system continues to be congested, and shrinking the DRB queue does not have any effect on the delay or the number of packets in the system.

As an alternative solution, in the fifth case, CoDel is implemented at the DRB queue. It shows a clear advantage on the way to reduce the congestion of the system. The total number of packets in the system is significantly reduced as can be observed from the queues' occupancy in Fig. 3. CoDel discards packets if the lowest sojourn time exceeds the target packet delay time in an interval, effectively dropping the TCP transmitting rate, and avoiding the creation of persistent queues. Since the occupancy level of the buffers is low, the delay-sensitive traffic can avoid large sojourn time in queues, and thus, it is delivered faster as observed in Fig. 4. Unfortunately, CoDel also introduces an important variation at the DRB queue occupancy as observed by the standard deviation in Fig. 3, which also translates to the throughput (Fig. 5) and to the delay (Fig. 4) performance. The variation at the DRB queue occupancy leads to TTIs where the DRB queue does not have enough packets to fulfill the maximum egress rate, and therefore, the total TCP throughput is reduced since not all the transmission opportunities are exploited (Fig. 5).

The sixth, seventh and eight cases correspond to our proposed solution of limiting DRB buffer size and using CoDel for the QFI queue. Again, the DRB buffer is limited to 10, 20 and 30 packets, respectively. In this solution, the DRB queue's standard deviation is reduced (Fig. 3) as CoDel acts in the QFI, and therefore, the TTIs where the DRB does not have enough packets to fulfill the maximum egress rate are reduced. Augmenting the size of the DRB buffer, reduces the possibilities of squandering transmit opportunities. However, there exists a limit where augmenting the buffer will not augment the throughput, as all the transmission opportunities are already exploited. From Fig. 5, it can be observed that augmenting the buffer from 20 to 30, does not lead to a throughput growth in the full interval range (Fig. 5). Moreover, as it can be seen from Fig. 4, incrementing the DRB queue capacity increases the delay. CoDel manages to maintain the buffer occupancy at the QFI queue low, but the RTT augments as the *ping* packet's sojourn time increases according to DRB's buffer capacity.

One of the effects observed is the TCP throughput rise as the *ping* interval increases. As there are less delay-sensitive traffic packets in the system, a larger amount of packets from the TCP flow can be forwarded and, therefore, the throughput increases. If CoDel is in the congested state after an interval time, it discards the next egress packet from the queue without distinguishing the packet type. 3GPP states that all the packets that are aggregated to one flow must be treated equally and, therefore, discarding packets with delay-sensitive requirements

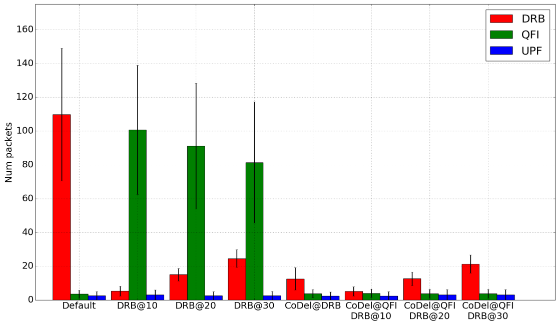

Fig. 3. 1st scenario: Average queue occupancy, *ping* interval of 10 ms.

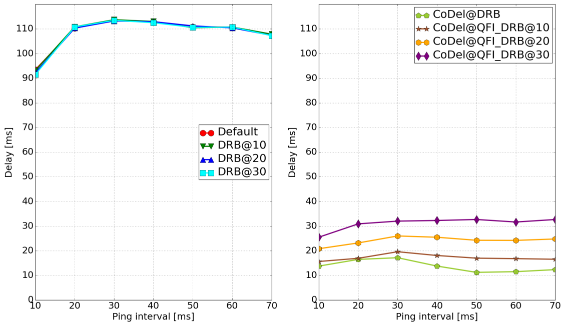

Fig. 4. 1st scenario: Average RTT for delay-sensitive flow.

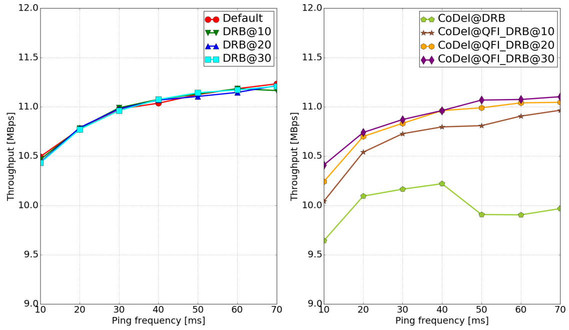

Fig. 5. 1st scenario: Average throughput of TCP flow.

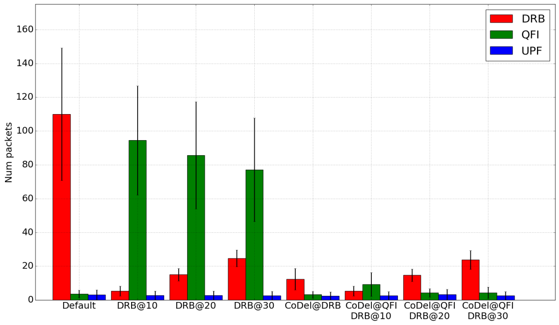

Fig. 6. 2nd scenario: Average queue occupancy, *ping* interval of 10 ms.

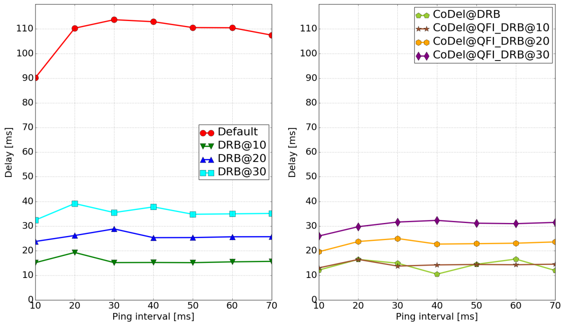

Fig. 7. 2nd scenario: Average RTT for delay-sensitive flow.

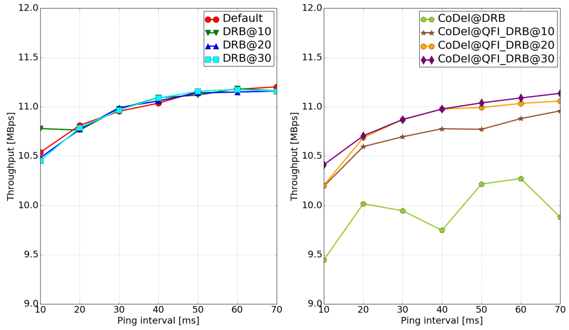

Fig. 8. 2nd scenario: Average throughput of TCP flow.

happens. However, in this scenario, just 0.79% of all the packets emitted are discarded by CoDel. From Figs. 4 and 5, it can be extracted that, a queue size limit of 20 packets at DRB in conjunction with CoDel at QFI substantially reduces the delay while keeping the throughput high, leading to an appropriate balance between both metrics.

The experimental results from the second scenario are shown in Figs. 6, 7 and 8. In the first case, all the packets accumulate at the DRB queue, following the same trend as in the first scenario. No significant reduction in the delay can be obtained from the segregation of the flows in different QFI flows, as observed in Fig. 7, if the DRB queue is not limited.

However, the throughput remains fully utilized as observed in Fig. 8. In the second, third and fourth cases, the queue at the DRB is limited to 10, 20 and 30 packets, respectively. In these cases, the packets corresponding to the delay-sensitive traffic benefit from the flow segregation and are enqueued into the DRB queue in a Round Robin manner without suffering the delay associated to the TCP flow in the congested QFI buffer. This approach reduces the latency drastically as can be seen from Fig. 7.

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

This case is comparable to the scenario at [10], where the traffic is segregated in two different flows before being forwarded to the lower layers for prioritization purposes. The throughput is kept high as all the transmission opportunities are used (Fig. 8).

Another solution is shown at the fifth case, where CoDel is implemented at the DRB. The CoDel mechanism maintains the DRB buffer occupancy low as observed in Fig. 6. However, Fig. 8 shows that, in this case, throughput cannot be maximized for the same reasons aforementioned for the same case.

For the last solution and the sixth, seventh and eighth cases, our proposals are evaluated, where CoDel is implemented at both QFI queues, while the DRB queue is limited to 10, 20 and 30 packets, respectively. CoDel successfully maintains the QFI queue occupancy level low, discarding some packets, while all the packets from the delay-sensitive flow are forwarded as they do not exceed the target time. From Fig. 8, it can be observed that a 10 packet queue at DRB decrements the throughput, while the limited queues of 20 and 30 packets are close to the maximum achievable throughput. The delay increases as the DRB queue limit rises as observed in Fig. 7.

Maintaining the bulky and delay-sensitive traffic segregated in different QFIs leads to good TCP throughput and reduced delay as shown in Figs. 7 and 8. However, the number of QFIs per UE and DN are limited, thus, some services will inevitably share QFIs in real deployments. Therefore, the second scenario presented in this paper is not scalable. Hence, a good solution for the first scenario is also critical for 5G systems. Moreover, due to 5G's channel capacity variability in the radio access, determining the optimal limited queue size can be challenging, and overdimensioning the queue will inevitably lead to larger sojourn times than necessary. Hence, an adaptive approach such as the AQM method proposed in this paper is needed for 5G. While achieving such dynamic, CoDel has only discarded 0.5% of the delay-sensitive flow packets in the evaluated scenarios. Hence, if deployed at the correct entity, our proposed solution is not detrimental to the throughput, while achieving low delays.

VI. Conclusion

Sharing of queues by different services with QoS criteria is an unavoidable phenomenon in 5G networks, for which an exponential increase of traffic is expected. A congested system will be challenging for low-latency services that have to guarantee time constraints. This paper shows the benefits that AQM can bring to the 5G network, exploring the new QoS scenario with the recently included SDAP entity. In this work, non-3GPP compliant solutions have been avoided. We evaluated CoDel with limited buffer sizes at different layers and entities. Through physical experiments, we show that AQM mechanisms and limited queues can reduce the low-latency traffic delay by a factor of 4 by reducing the queue occupancy, while maintaining the competing TCP flow's throughput close to the achievable maximum. We empirically demonstrate that AQM mechanisms as well as intelligent buffer limitations will be key enablers in the future 5G QoS scenario.

Acknowledgment

This work has been supported in part by the EU Horizon 2020 research and innovation program under grant agreement No. 675806 (5GAuRA), in part by the 5G-PICTURE project, co-funded by the European Commission 5G-PPP Programme under grant number No. 762057, and in part by the Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement from the Generalitat de Catalunya.

References

[1] "The bufferbloat project," https://www.bufferbloat.net/projects/.

[2] X. Che and B. Ip, "Packet-level traffic analysis of online games from the genre characteristics perspective," *J. Network and Computer Applications*, vol. 35, pp. 240–252, 01 2012.

[3] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2429–2453, thirdquarter 2018.

[4] 3GPP, "NR, Radio Link Control (RLC) specicication," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.322, Jan. 2019, version 15.4.0.

[5] A. K. Paul, H. Kawakami, A. Tachibana, and T. Hasegawa, ""An AQM based congestion control for eNB RLC in 4G/LTE network"," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2016, pp. 1–5.

[6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug 1993.

[7] W. chang Feng, K. G. Shin, D. D. Kandlur, and D. Saha, "The blue active queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 513–528, Aug 2002.

[8] K. Nichols and V. Jacobson, "Controlling queue delay," *Queue*, vol. 10, no. 5, pp. 20:20–20:34, May 2012.

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

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

[11] 3GPP, "System architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.501, Dec. 2018, version 15.4.0.

[12] "Saegw administration guide," https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21-8_6-2/SAEGW-Admin/21-8-SAEGW-Admin/21-8-SAEGW-Admin_chapter_0111010.html.

[13] Marcus Ihlar, Ala Nazari, Robert Skog, "Low latency, high flexibility - virtual aqm," https://www.ericsson.com/en/ericsson-technology-review/archive/2018/virtual-aqm-for-mobile-networks, 2018.

[14] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '95. New York, NY, USA: ACM, 1995, pp. 231–242.

[15] "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm," RFC 8290, Jan. 2018.

[16] "Openwrt," https://openwrt.org/.

[17] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Service Data Adaptation Protocol (SDAP) specification," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 37.324, Sep. 2018, version 15.1.0.

[18] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, "The flow queue codel packet scheduler and active queue management algorithm," Internet Requests for Comments, RFC Editor, RFC 8290, Jan. 2018.

[19] "Byte queue limits," https://lwn.net/Articles/469652/.

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