



Cross-layer Loss Discrimination Algorithms for MEC in 4G networks

Mamoutou Diarra, Thierry Turetti, Walid Dabbous, Amine Ismail

► To cite this version:

Mamoutou Diarra, Thierry Turetti, Walid Dabbous, Amine Ismail. Cross-layer Loss Discrimination Algorithms for MEC in 4G networks. 2020. hal-03001893

HAL Id: hal-03001893

<https://inria.hal.science/hal-03001893>

Preprint submitted on 12 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cross-layer Loss Discrimination Algorithms for MEC in 4G networks

Mamoutou Diarra
Inria, Université Côte d'Azur
Ekinops France
mamoutou.diarra@inria.fr

Thierry Turletti
Inria, Université Côte d'Azur
France
thierry.turletti@inria.fr

Walid Dabbous
Inria, Université Côte d'Azur
France
walid.dabbous@inria.fr

Amine Ismail
Ekinops France
amine.ismail@ekinops.com

Abstract—Traditional loss-based Congestion Control Algorithms (CCAs) suffer from performance issues over wireless networks mostly because they fail to distinguish wireless random losses from congestion losses. Different loss discrimination algorithms have been proposed to tackle this issue but they are not efficient for 4G networks since they do not consider the impact of various link layer mechanisms such as adaptive modulation and coding and retransmission techniques on congestion in LTE Radio Access Networks (RANs). We propose MELD (MEC-based Edge Loss Discrimination), a novel server-side loss discrimination mechanism that leverages recent advancements in Multi-access Edge Computing (MEC) services to discriminate packet losses based on real-time RAN statistics. Our approach collects the relevant radio information via MEC's Radio Network Information Service and uses it to correctly distinguish random losses from congestion losses. Our experimental study made with the QUIC transport protocol shows over 80% higher goodput when MELD is used with New Reno and 8% higher goodput when used with Cubic.

Index Terms—CCA, LDA, LTE, MEC, RNIS, QUIC, RAN, AMC, CQI, TBS.

I. INTRODUCTION

Most of today's download data flows are controlled by loss-based congestion control algorithms (CCAs) that consider any packet loss as a congestion signal. The default behavior of such congestion control algorithms once being notified of a packet loss is to decrease their sending rate by reducing the current window size (e.g., RENO, Cubic). Although such a technique is appropriate when losses are caused by congestion, it degrades performance in case of random errors caused by lossy wireless links [1]. Therefore, in order to avoid unnecessary rate throttling, loss discrimination algorithms (LDAs) have been proposed to help transport layer protocols distinguish (random) wireless losses from congestion losses in wireless networks, and especially for WiFi networks [1], [2]. However, proposed techniques used by existing LDAs are not efficient in LTE (Long Term Evolution) networks mainly because of different mechanisms used in the radio access network (RAN). Indeed, most LDA approaches proposed in the literature either rely on RTT increase or buffer occupancy measurements in order to differentiate losses. These two techniques are not accurate in LTE for the following reasons: first, detecting a loss at link layer requires at least 8ms in LTE FDD (Frequency Division Duplexing) and another 8ms for the retransmission. Meanwhile in WiFi, link layer loss detection is almost immediate (i.e., within a few μ s) and retransmission may take similar time depending on the number of connected stations. Based on that, it is clear that RTT variations during

link error losses are more significant in LTE, which doesn't necessarily mean congestion. Second, unlike wireless access points, LTE base stations (eNBs) are generally provisioned with large buffers in order to accommodate rapid changes in link capacity and to compensate for the additional processing time per UE. Therefore, those buffers must be appropriately filled in order to reach optimal link utilization. In that case, relying only on buffer occupancy to infer congestion losses can lead to link under-utilization. Furthermore, unlike Ethernet-based networks where a transport layer packet data unit (PDU) is directly encapsulated in a single link layer PDU (except in cases involving IP fragmentation), the LTE link layer subdivides the layer-4 PDU in several transport blocks whose size depends on the actual radio conditions (encoded in the Channel Quality Indicator (CQI)) and on the number of allocated radio resources. As indicated in 3GPP TR 36.912 [3] (see Figure 1), only one transport block can be sent per transmission time interval (TTI), i.e., every millisecond, and since each transmission is controlled by an independent Hybrid Automatic Repeat Request (HARQ) process, the number of transport blocks that can be sent without interruption (i.e., without waiting for an HARQ ACK or NACK) depends on the total number of HARQ processes. Based on that, it becomes

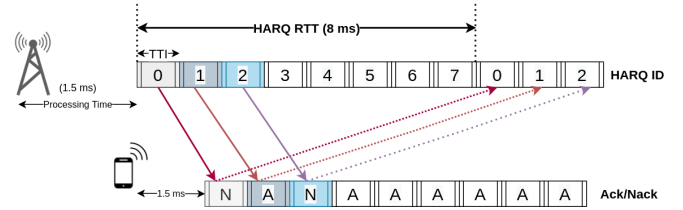


Fig. 1: RAN latency introduced by HARQ RTT and processing times at eNB and UE for LTE-FDD.

clear that the time required to send a transport layer PDU is inversely proportional to the Transport Block Size (TBS).

Consequently, a large mismatch in size between transport blocks and layer-4 PDU and/or retransmissions at lower layers can lead to congestion in the RAN. The challenge in this context is to highlight to what extent the involved radio information, mainly CQI and TBS can be used in order to infer congestion losses. Retrieving and making such radio information available to upper layers require either additional services at the base station (eNB) or modifications in the UE. The latter approach is used by recently proposed cross-layer CCAs such as PBE-CC [4] or CQIC [5] that rely on

client-side radio measurements to adjust their sending rate. Although these approaches guarantee higher link utilization, they also introduce computational overhead and additional power consumption at the UE. Instead, our solution doesn't involve the end user but rather relies on ETSI Multi-access Edge Computing (MEC) [6] concepts to collect radio information. Using MEC, applications deployed at the edge can subscribe to a set of MEC services, one of them being the Radio Network Information Service (RNIS) which allows the applications to receive real-time radio information.

In this paper, we highlight to what extent specific information transmitted over LTE RAN, such as CQI and TBS can be used as relevant indicators of congestion in the access network. We leverage the aforementioned radio information at the transport layer by proposing MELD, a new loss notification mechanism for reliable transport protocols. Our proposed approach comes in two variants, MELD-DE (for Download Enhancement) and MELD-ME (for Mixed flows Enhancement). The former discriminates losses and focuses on maximizing goodput while the latter attempts to maintain fairness between short and long flows of the same UE. The two variants are designed for edge delivery scenarios such as short and large file download from servers located at the edge of the network (i.e., close to the RAN). At a basic level, MELD discriminates losses observed at the transport layer by leveraging up-to-date link layer information. For that, it relies on FlexRAN [7], a software defined RAN platform (SD-RAN) that implements RNIS to collect relevant RAN information. Our proposed approach could help improve the performance of any loss-based CCAs regardless of the underlying transport protocol (e.g., TCP or QUIC). In this paper, we rely on QUIC as the transport protocol because it is more resilient to random losses and high RTT variations that are common in wireless networks [8], [9]. We implemented our solution with one of the most complete implementation of IETF QUIC [10] called picoquic [11] and evaluate its performance with controlled experiments done in the R2lab wireless testbed¹.

The remainder of this paper is organized as follows. Section II presents the building blocks of our approach by exposing the impact of LTE RAN mechanisms on reliable transport layer protocols before introducing RNIS and highlighting the reasons why it is relevant in the proposed scheme. Section III explores the related work and highlights their limitations. Section IV introduces our proposed MELD scheme and outlines the basic concepts of its design and implementation. Section V presents and interprets the results of our experiments. Finally, we summarize our contributions and future research directions in Section VI.

II. BACKGROUND

In this section, we first give an overview of some relevant LTE FDD radio access parameters and highlight the transport layer issues they introduce. Then we provide a brief overview of a particular MEC service that can be used to mitigate those

issues, namely the RNIS, which inspired us to propose the MELD approach.

A. LTE RAN and L4 issues

Recent papers and experiments [12], [13] show that most reliable transport layer protocols experience throughput degradation and rate throttling over LTE networks. This phenomenon is mainly due to misinterpretations of some LTE radio access techniques and mechanisms by transport layer protocols (e.g., TCP, QUIC). The most important ones are:

Link adaptation and channel coding: Because of fast varying radio conditions, the base station selects for each user equipment (UE), the adequate modulation and coding scheme (MCS) depending on the reported downlink CQI. Hence TCP and QUIC cannot adapt to fast varying link capacity.

Retransmissions at lower layers: In LTE, besides the forward error correction performed by channel coding, some retransmission techniques are used in Radio Link Control (RLC) and Medium Access Control (MAC) layers, respectively Automatic Repeat Request (ARQ) and Hybrid Automatic Repeat Request (HARQ). ARQ at RLC layer allows retransmissions of missing blocks due to residual HARQ errors (e.g., due to misinterpreted NACKs or retry threshold reached) when using the Acknowledge Mode (AM). These retransmissions are interpreted at the transport layer as RTT increase or bufferbloat signals.

Resource allocation and transport block size: At a basic level, LTE RAN resources are organized in term of physical resource blocks (PRBs). Typically, a PRB lasts one slot (0.5ms) and is composed of 12 sub-carriers, each of which composed of 7 Resource Elements (i.e., 7 symbols). A PRB is the smallest unit of resources that can be allocated to a UE (User Equipment) and it is always allocated in pairs (i.e., 1ms for 2 consecutive slots). The transport block corresponds to the number of bits transmitted per TTI (Transmission Time Interval i.e., 1ms for LTE), hence its size, called Transport Block Size (TBS), depends both on the number of allocated PRBs and on the modulation scheme (e.g., BPSK, 16QAM, etc.). A large difference between TBS and transport layer segment size (MSS) can lead to several retransmissions and/or congestion.

Large buffers: In order to hide packet losses and retransmissions at the physical layer and to accommodate the varying network conditions, LTE networks are generally provisioned with extremely large buffers (e.g., in-flight bytes larger than 1200kB in [12]), which can penalize short flows in presence of bufferbloat.

B. RNIS as a Service

As specified in ETSI MEC standards [6], RNIS provides real-time radio network related information to mobile edge applications. The MEC applications that subscribed to this service can then leverage up-to-date information to dynamically adapt their behaviors to the current RAN conditions. The use of RNIS in an as-a-service manner has been evaluated in [14], with an implementation of a publish-subscribe scheme

¹R2lab Testbed: <http://r2lab.inria.fr>

based on RabbitMQ that is suitable for real-time use-cases. For instance, using a similar approach, the ETSI RAVEN POC [15] showcased a video optimization MEC application that dynamically adjusts the quality of the video streams according to the radio conditions of the users. Similarly, our MELD plugin allows picoquic to exploit relevant radio information (i.e., CQI, TBS and TxQueue length) in order to differentiate at the transport layer congestion losses from random transmission losses. Unlike other LDAs proposed in the literature, our solution takes into account the inherent characteristics of LTE RAN as inputs in the decision-making process.

III. RELATED WORK

As observed by Junxian Huang et al. in [12], more than 12% of TCP flows in traces collected from a commercial LTE network experience undesired slow start due to the loss of a single packet. More precisely, they identified that retransmission timeout (RTO) expired due to several ARQ/HARQ retransmissions or bufferbloat, and proposed a solution to update the RTO upon receipt of duplicate ACKs. However, with such a solution, the detection of real congestion may be delayed since TCP cannot distinguish between congestion and random losses with duplicate ACKs.

In the literature, several LDAs [16], [17], [18] have been proposed in order to detect random losses in wireless networks. Samaraweera suggests NCPLD [19], a non-congestion packet loss detection scheme that implicitly identifies the type of packet loss using the variation of delay experienced by TCP packets. TCP Veno [2], [1], proposed by Fu et al. estimates the number N of excess packets in the bottleneck buffer. In case of packet loss, Veno declares random loss if $N < 3$. LDA_EQ [1], similar to TCP Veno, estimates queue usage using information available to TCP. In case of packet loss, congestion is declared when the estimated queue usage is larger than a certain threshold. Ben-Jye Chang and Yi-Hsuan Li proposed a cross-layer-based adaptive TCP algorithm for 4G networks [20]. Their approach detects bottleneck location by analyzing client-side cross-layer radio information included in TCP ACKs and discriminates losses based on bottleneck buffer occupancy estimated via delay variation measurements. However, their solution doesn't show significant performance gain under low loss rates (i.e., $\leq 1\%$), which is the most frequent case in commercial LTE networks [21].

Just like NCPLD, TCP Veno or LDA_EQ most LDAs (e.g., West, JTCP, RELDS [1]) are either based on RTT variations or on an indication about queue usage. In a LTE context, these two techniques are not appropriate since the RTT varies with TBS and radio conditions (CQI) and queue length can also grow due to retransmissions at lower layers.

Other alternatives to loss-based could be model-based CCAs based on available bandwidth estimation (ABE), which reduce their window size in a more intelligent fashion in case of random losses. The most recent algorithm in this category is BBR [22], which targets an operation point where a Bandwidth Delay Product (BDP) worth of data in flight is maintained

while keeping bottleneck buffers empty (i.e., with minimal delays). Although Google claims that BBR is 2700 times faster than CUBIC on fixed networks under 1% loss [23], authors in [24], [25] demonstrate that the two CCAs have similar performance over LTE links. Indeed, algorithms that limit queue occupancy can be effective against bufferbloat but they are not optimal in LTE networks for which buffering is required to take advantage of fast variations in link capacity. Therefore, as suggested by Feng Li and al. [24] a tuning in such CCAs might be needed in order to keep buffers appropriately filled in LTE networks. Although recently proposed client-centric CCAs such as PBE-CC [4] or CQIC [5] can overcome BBR limitations in LTE and adapt to fast variations in link capacity by measuring the available bandwidth in the RAN, they require client-side modifications and introduce additional energy consumption in the UE.

Solutions involving Active Queue Management (AQM) techniques are also proposed in the literature to help mitigate bufferbloat. However, the analysis in [16] suggests that the most appropriate buffer management scheme to use depends on the application. For instance, CoDel (Controlled Delay) is mostly effective when the application needs both throughput and low latency (already achieved by BBR).

In summary, there is not any solution available in the literature to accurately discriminate random losses while taking into account the ever-changing nature of LTE radio link capacity. In this paper we aim to demonstrate that this problem can be mitigated by correctly interpreting the values of certain RAN information at the transport layer, namely, CQI, TBS and per UE transmission queue (TxQueue) length at the base station. Such a server-side cross-layer approach is only possible in MEC scenarios where the end-server is deployed at the edge.

IV. LOSS DISCRIMINATION VIA RADIO NETWORK INFORMATION SERVICE

In this section, we first highlight the correlation between radio information mentioned in previous sections and LTE loss differentiation. Then we describe the design and implementation of our proposed solution based on the analysis of RAN information.

A. Radio information as congestion signal

In LTE, each UE sends (periodically or not) a Channel Status Information (CSI) report that takes at least 8ms to reach the base station as indicated in 3GPP TR 36.912. The CSI report includes the UE computed CQI value that corresponds to a predefined MCS at the base station side. As illustrated in [26], a low CQI value indicates a low SINR, hence less bits per symbol in the modulation. CQI, therefore, has a direct influence on the transport block size, regardless of the number of allocated PRBs. So, the number of transport blocks required to transmit a single transport layer PDU highly depends on the CQI value. Consequently, this number must be less than or equal to the number of HARQ processes in order to send the whole PDU without interruption, thus minimizing the risk of bufferbloat at the BS (high buffering) and eventual RTO

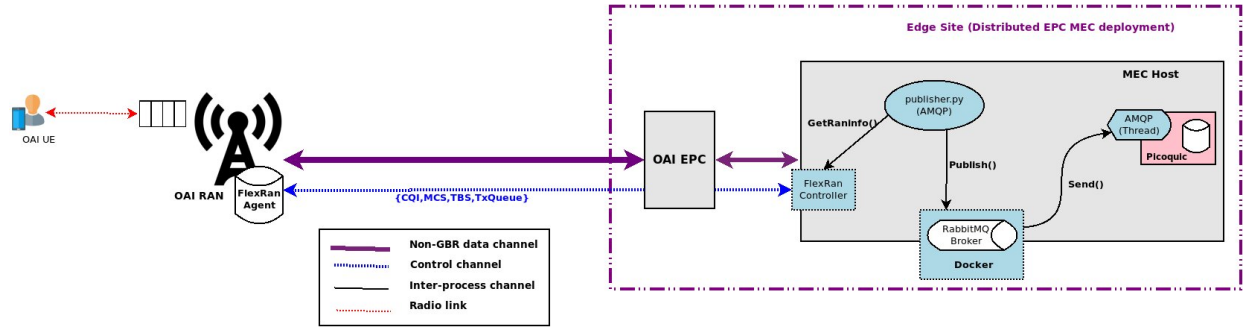


Fig. 2: MELD experimentation setup.

expiration. Let S_{TB} be the optimal size of transport blocks so that a transport layer PDU can be transmitted without interruption, T_{PDU} the size of transport layer PDU (TCP or QUIC) and N_{HARQ} , the number of HARQ processes. In that case, at least one transport layer PDU is transmitted without interruption (i.e., within HARQ RTT) only if:

$$TBS \geq S_{TB} \quad \text{where: } S_{TB} = \frac{T_{PDU}}{N_{HARQ}} \quad (1)$$

Thus, a TBS value less than S_{TB} is likely to introduce queuing delay and congestion. To put it another way, packet losses during low TBS state (i.e., low CQI and/or small number of PRBs assigned to UE) have high probability to cause congestion in the radio network. Conversely, packet losses occurring during high TBS state (i.e., high CQI and/or sufficient number of PRBs assigned to UE) can be attributed to fast random fluctuations of the SINR. Mapping between CQI and MCS is vendor-specific [27], once this mapping is performed and the number of allocated PRB is known, the corresponding TBS can be retrieved from Table 7.1.7.2.1-1 in [28]. However, even with the right TBS, checking only Equation 1 for loss discrimination can penalize short flows since long flows will keep increasing their window size as long as the base station send buffer is not filled. On the other hand, some buffering is necessary to accommodate rapid changes in transport block size and to compensate mismatch between arrival and sending rate at the base station. Therefore, in order to maintain fairness between different possible data flows of the same UE, it is necessary to make sure that the send queue allocated to the UE at the BS does not exceed the optimal threshold Q_{th} . This threshold must be large enough in order to guarantee full link utilization in case of favorable radio conditions. For that reason, let S_{TBm} be the maximum TBS under the current allocated PRBs (at CQI=15), the BS send queue size for a given UE must satisfy the following:

$$S_{queue} \geq Q_{th} \geq S_{TBm} * N_{HARQ} \quad (2)$$

Enforcing Equations 1 and 2 guarantees optimal link utilization, lower RTT for short flows and accurate congestion detection in case of packet losses. In the subsequent section, we describe MELD, a LDA scheme implemented on top of picoquic that checks the two aforementioned conditions after collecting the required radio information.

B. Loss differentiation in MELD

In this section, we introduce MELD, a novel LTE-friendly LDA based on picoquic and the FlexRAN. At a basic level, MELD listens to RAN information through FlexRAN in order to decide whether a lost packet should be retransmitted silently or reported to the congestion controller. As illustrated in Figure 2, we use a local LTE environment based on Open Air Interface (OAI) and FlexRAN.

1) *Design and Implementation:* MELD is built on top of picoquic on the server side, the client does not require any modification. Similar to the model described in [14], [7], the FlexRAN controller gets real-time RAN statistics from the FlexRAN agent every millisecond via the control channel. The relevant RAN information is then retrieved by a Python process and published on an Advanced Message Queuing Protocol (AMQP) topic as illustrated in Figure 2. The picoquic server deployed at the MEC host then uses a separate thread to subscribe to the topic in order to receive real-time changes from FlexRAN agent. Basically, AMQP requires two main components to work at the server, namely a message broker and a publisher. The publisher in our case is our Python process that retrieves the relevant RAN data from the controller and the broker is a rabbitmq container that broadcasts the published message to the subscribed applications. Once the picoquic server receives the radio information (CQI, MCS, TBS, Txqueue length) via AMQP, it computes the TBS threshold (S_{TB}) and the queue threshold (Q_{th}) in case of packet loss(es). Then it decides whether it should notify the CCA or just retransmit the lost packet(s) silently. This process is actually enforced by modifying the picoquic loss notification logic. Note that S_{TB} is the value of TBS guaranteeing the transmission of one transport layer PDU without interruption; an example mapping between CQI, MCS and TBS is given in Table II in the case of 25 allocated PRBs.

2) *Overhead of FlexRAN notifications:* In order to use all the available bandwidth for user traffic, we dedicate a separate control channel for FlexRAN. Our approach based on analysis of up-to-date radio information requires FlexRAN agent (collocated with the eNodeB) to report RAN statistics to the controller (in the MEC host) every TTI (i.e., every ms).

Authors in [7] conclude that the network overhead introduced by such continuous agent-controller communication mode increases with the number of UEs and experiments show that it can reach up to 100 Mbps for 50 UEs.

3) *MELD LDA*: Our MELD loss discrimination algorithm is invoked at each packet loss event. As illustrated in Algorithm 1, TBS_{th} and Q_{th} are computed using the downlink CQI ($dlcqi$), the downlink TBS ($dltbs$), the length of the BS per UE send queue ($txQueueLen$) the number of allocated PRBs ($nprb$), the number of HARQ processes ($nharq$) and the size of the transport layer PDU ($stpdu$) downlink sending queue size ($txQueueSize$) collected via FlexRAN. In case of loss, the congestion controller is notified when TBS is less than TBS_{th} (which could result from a low CQI or a decrease in the number of allocated PRBs) and when the sending queue at the BS ($txQueue$) exceeds Q_{th} . The latter condition prevents large bufferbloat and guarantees fairness between short and long flows. In all other cases, the packet is silently retransmitted, i.e., with a high CQI value or/and sufficient PRBs.

Algorithm 1: MELD LDA logic

```

Result: NotifyCCA
1 NotifyCCA  $\leftarrow$  TRUE;
2 if Pktloss then
3    $TBS_{th} \leftarrow \text{ComputeTbsThresh}(dltbs, nharq, stpdu);$ 
4    $Q_{th} \leftarrow \text{ComputeQThresh}(nprb, dlcqi, dltbs);$ 
5   if ( $dltbs \geq TBS_{th}$ ) && ( $txQueueLen \geq Q_{th}$ ) then
6     NotifyCCA  $\leftarrow$  1;
7     exit();
8   end
9   if ( $dltbs < TBS_{th}$ ) then
10    NotifyCCA  $\leftarrow$  1;
11    exit();
12  else
13    NotifyCCA  $\leftarrow$  0; /* Retransmit lost packets silently */
14  end
15 end

```

V. EXPERIMENTATION AND RESULTS

To evaluate the performance of our approach, we rely on an edge delivery application based on MELD. Table I describes the characteristics of our experimentation setup on the R2lab wireless testbed. We generate controlled interference in the RAN using an Additive White Gaussian Noise (AWGN) generator that introduces 0.8% random packet loss [21]. We analyze the introduced packet loss rate profile on continuous ICMP ping traffic sent at 560kbps, (10 runs of 120 seconds), see Figure 3. The loss distribution suggests that packets lost with this controlled interference are mostly due to fast variations of the SINR between CSI reports, which are known for hindering AMC performance [29]. During all the following experiments, we use the same level of interference and evaluate the performance of a 20MB file download over the 15Mbps radio link.

We repeat each test 10 times in order to account for the variability of link capacity. We first evaluate the global goodput using a loss discrimination algorithm based only on TBS, as suggested by Equation 1; then we analyze the proposed MELD algorithm. Figures 4-6 show the different test results with a 90% confidence interval.

TABLE I: OAI Experimentation setup characteristics.

Parameters	Values
LTE BW	5MHz
# of PRBs	25
E2E RTT	55ms
L4 protocol	QUIC
Max TBS	2292 Bytes
Link Capac.	15 Mbps

TABLE II: Measured CQI values and TBS for 25 PRBs.

CQI index	MCS	TBS (Bytes)
15	28	2292
14	28	1980
13	28	1764
12	28	1572
11	28	1239

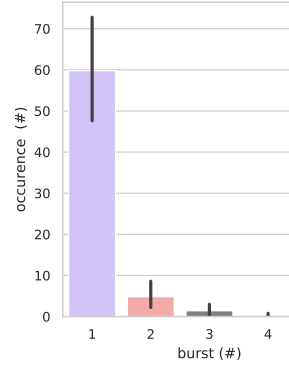


Fig. 3: Burst loss occurrences

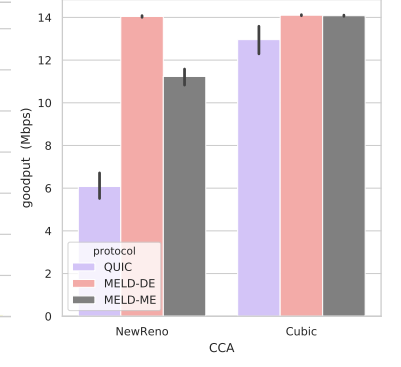


Fig. 4: CCA performance with MELD-DE/ME

A. TBS-only loss discrimination

In this first experiment, we take the decision to notify lost packets to CCAs only based on the transport block size (MELD-DE). The mapping between CQI, MCS and TBS is performed using values from Table II which correspond to values reported by FlexRAN in our local setup.

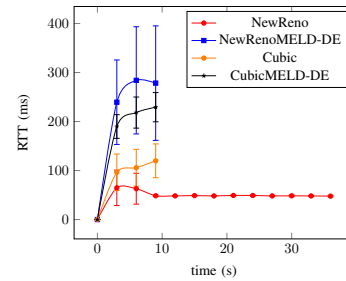


Fig. 5: RTT for MELD-DE vs legacy QUIC

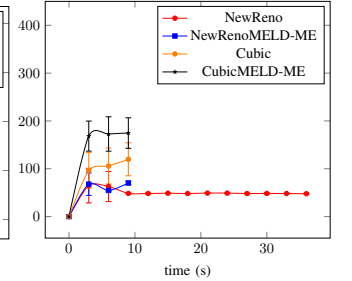


Fig. 6: RTT for MELD-ME vs legacy QUIC

We assume that the channel quality is very poor or the UE is not assigned enough resources when the selected MCS corresponds to a TBS less than S_{TB} . Since N_{HARQ} is equal to 8 and the maximum QUIC datagram size is set to 1440 Bytes, the optimal size of transport blocks (S_{TB}) must be greater or equal to 180 Bytes (1440 bits) in order to validate Equation 1. In other words, CCA is notified of the loss and congestion window reduction is applied whenever $TBS < 180$ B. The test results with NewReno using the TBS threshold show a sig-

nificant performance gain. As illustrated in Figure 4, a 131% increase in goodput is observed, mostly due to the fact that the congestion window is not halved in case of random losses. On the other hand, the same experimentation with Cubic does not double the goodput. We only observe 9% performance gain over the legacy version (see Figure 4). Since Cubic already uses an aggressive growth function, the use of LDA increases the link utilization to a near optimal value. The difference in download time can be observed in Figure 5. Results show that flows under MELD-DE have shorter duration because they exploit more bandwidth. However, regarding the RTT increase (see Figure 5), we can observe significant RTT increase mainly due to buffering. The large RTT increase implies that MELD-DE doesn't take queue length into account.

B. TBS and queue length-based loss discrimination

This experiment evaluates the MELD-ME algorithm, proposed in Section IV, in terms of average goodput and RTT increase. The results show for New Reno, over 80% increase in goodput (see Figure 4) when compared to legacy picoquic. As expected, taking queue length into account to prevent RTT increases negatively affects the global throughput. As illustrated in Figures 4 and 6, MELD-ME prevents large RTT increase (bufferbloat) for NewReno and Cubic at the cost of slightly lower goodput. On the other hand, with Cubic, although a 8% increase is observed in the goodput, the measured RTTs are above legacy Cubic RTTs during the whole connection. Such a behavior was expected since Cubic growth function is more aggressive than NewReno Additive Increase Multiplicative Decrease (AIMD). As shown in the test results, for both CCAs, the RTT is kept under a lower value which was not the case with MELD-DE. This behavior confirms the efficiency of Equation 2 which allows MELD-ME to be fair to short flows even when the UE is in ideal radio conditions (i.e., with CQI=15).

As shown in the different results, MELD-DE outperforms MELD-ME in term of goodput but struggles to maintain a reasonable RTT increase for short flows. On the other hand, MELD-ME exhibits lower RTT increase and fairness at the cost of slightly lower performance.

VI. CONCLUSION

In this paper we demonstrate that certain information transmitted over LTE Radio Access Network, such as Channel Quality Indicator (CQI) and Transport Block Size (TBS) can be exploited as relevant congestion signals at the transport layer. To the best of our knowledge, our proposed MELD scheme is the first loss discrimination algorithm for LTE that exploits such information to proactively discriminate packet losses. Our performance evaluation using MELD-DE and MELD-ME LDAs show significant improvement for loss-based congestion controllers. MELD-DE is adapted for sheer download scenarios where RTT increase and fairness to short flows can be ignored. On the other hand MELD-ME can be used in mixed scenarios (i.e., when a UE is downloading short and long flows at the same time) since it improves the goodput

while minimizing RTT increase. Our proposed algorithms are implemented as plugins integrated into picoquic. The code and explanation of all the necessary steps to reproduce our results can be retrieved at [30]. We hope this paper paves the way for future work on leveraging MEC services to improve transport layer protocols performance in mobile networks. While the MELD approach has been proposed for LTE, we believe that it could also be exploited in 5G networks with potential minor changes.

REFERENCES

- 1 Park, M. and Chung, S., "Analyzing effect of loss differentiation algorithms on improving TCP performance," in *Proc of ICACT*, vol. 1, 2010.
- 2 Cheng Peng Fu *et al.*, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE JSAC*, vol. 21, no. 2, 2003.
- 3 3GPP, "Feasibility study for further advancements for e-utra (lte-advanced) (rel. 16)," in *3GPP TR 36.912 V16.0.0*, 2020.
- 4 Xie, Y. *et al.*, "PBE-CC: congestion control via endpoint-centric, physical-layer bandwidth measurements," in *preprint arXiv:2002.03475*, 2020.
- 5 Zhong, Z. *et al.*, "Performance evaluation of ccqic and tcp bbr in mobile network," in *Proc. of ICIN*, 2018, pp. 1–5.
- 6 ETSI, "Multi-access edge computing (mec); radio network information api," in *GS MEC 012 V2.1.1*, 2019.
- 7 Foukas, X. *et al.*, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," in *ACM CoNEXT*, 2016.
- 8 Wolsing, K. *et al.*, "A performance perspective on web optimized protocol stacks: Tcp+tls+http/2 vs. quic," in *Proc. of ANR Workshop*, 2019.
- 9 Langley, A. *et al.*, "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Proc. ACM SIGCOMM*, 2017.
- 10 Piraux, M. *et al.*, "Observing the evolution of quic implementations," in *Workshop on the EPCI of QUIC*, New York, NY, USA, 2018.
- 11 Huitema, C., "picoquic," 2018, Software. [Online]. Available: <https://github.com/private-octopus/picoquic>
- 12 Huang, J. *et al.*, "An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance," *ACM SIGCOMM*, 2013.
- 13 Robert, R. *et al.*, "Behaviour of common tcp variants over lte," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.
- 14 Arora Sagar *et al.*, "Exposing radio network information in a MEC-in-NFV environment: the RNISaaS concept," in *Proc. of NETSOFT*, 2019.
- 15 Sabella, D. *et al.*, "A Hierarchical MEC Architecture: Experimenting the RAVEN Use-Case," in *IEEE VTC Spring*, 2018.
- 16 Varma, S., *Internet congestion control*. Morgan Kaufmann, 2015.
- 17 Fu C.P., L. S., "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE JSAC*, 2003.
- 18 Xu K., A. N., "TCP-Jersey for wireless communications," *IEEE JSAC*, 2004.
- 19 Samaraweera, N. K. G., "Non-congestion packet loss detection for TCP error recovery using wireless links," *IEE Proc. Com.*, vol. 146, 1999.
- 20 Chang, B.-J. *et al.*, "Cross-layer-based adaptive tcp algorithm in 4g packet service lte-advanced relaying communications," in *AISA - Vol. 1*, 2013.
- 21 Chen, Y.-C. *et al.*, "Measuring cellular networks: Characterizing 3g, 4g, and path diversity," in *Conf. Int. Tech. Alliance*, 2012.
- 22 Cardwell, N. *et al.*, "Bbr: Congestion-based congestion control," *ACM Queue*, vol. 14, September-October, pp. 20 – 53, 2016.
- 23 —, "congestion control comes to gcp your internet just got faster," *Google Cloud*, 2017.
- 24 Li, F. *et al.*, "TCP CUBIC versus BBR on the Highway," in *PAM*, 2018.
- 25 Claypool, M. *et al.*, "BBR: An Implementation of Bottleneck Bandwidth and Round-trip Time Congestion Control for ns-3," in *WNS3*, June. 2018.
- 26 Khirallah, C. *et al.*, "Design of Bandwidth and Energy Efficient Video Broadcasting Services over LTE/LTE-A," in *IEEE WCNC*, April 2013.
- 27 Roman, K. *et al.*, "An accurate approximation of resource request distributions in millimeter wave 3gpp new radio systems," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, 2019.
- 28 3GPP, "Lte evolved universal terrestrial radio access (e-utra); physical layer procedures," in *TS 36.213 version 14.2.0*, 2017.
- 29 Sassioui, R. *et al.*, "Harq and amc: Friends or foes?" *IEEE Transactions on Communications*, vol. 65, no. 2, 2017.
- 30 Diarra, M. *et al.*, "MELD," 2020, Software. [Online]. Available: <https://github.com/madi223/MELD>