Exploiting QUIC multi-streaming over NTN: Delay-based scheduling policies

Fátima Khan*, Fátima Fernández†, Luis Diez*, Ramón Agüero*
*Department of Communication Engineering, Universidad de Cantabria
fatima.khan@unican.es,{ldiez,ramon}@tlmat.unican.es
†IoT and Digital Platforms Department, Ikerlan Technology Research Center
ffernandez@ikerlan.es

***Abstract*—We analyze the performance of QUIC over Non-Terrestrial-Networks (NTN). The presence of this transport protocol has strongly increased since it was originally proposed, but its behavior over wireless networks is still an open question. We focus on NTN, which will play a key role in forthcoming cellular systems, and whose underlying connectivity exhibits a highly variable behavior. We show that using multiple streams, which is one of the most relevant advantages of QUIC, yields performance improvements. We then propose a delay-based scheduler, based on dynamic queuing control, and we compare its behavior with that exhibited by legacy solutions. We carry out an extensive experiment campaign exploiting a novel methodology that combines virtualization techniques, real QUIC implementation, and *ns-3* to model NTN links. The results show that the proposed scheduling policies fairly distribute the delay between streams, without jeopardizing the throughput, even under very high load situations.**

***Index Terms*—QUIC, stream multiplexing, scheduling, Non-Terrestrial Networks**

I. INTRODUCTION

The development of 5G and 6G technologies entails the tight integration of novel Radio Access Network (RAN) with the traditional terrestrial architecture. Consequently, Non-Terrestrial Networks (NTN) is now considered an essential element of forthcoming wireless networks, due to their capability to foster reliability, scalability, as well as coverage and service continuity. On the other hand, this integration also brings several challenges, such as the high variability of the underlying wireless links, which requires reassessing the performance of existing protocols.

In this sense, it is well known the poor performance exhibited by traditional transport protocols, most notably Transmission Control Protocol (TCP), over wireless channels, which exhibit a highly variable behavior. We have also witnessed the appearance of alternative transport layer solutions, where QUIC stands as one of the most relevant ones. QUIC was originally promoted by Google and has been recently standardized by the Internet Engineering Task Force (IETF) [1]. It was originally proposed to improve the performance of HTTP traffic, which usually involves short flows, as well as ensuring secure communications.

One of the main advantages of QUIC is that it avoids Head-of-line (HOL) blocking, by enabling multi-streaming. Whenever there is a loss event, it would just affect a particular stream, while the rest would not be affected. To the authors' best knowledge, there are very few works that have studied multi-streaming in QUIC, and none that have proposed schedulers to yield better performances exploiting this feature.

Hence, a second aspect that we tackle in this paper is to assess the benefits that the use of multiple streams might bring to QUIC performance, in particular over highly variable wireless links (NTN). Furthermore, we will develop *scheduling* algorithms that can yield lower delays, and we compare their behavior with benchmark solutions, such as the well-known Round-Robin (RR), which is the default scheduler of the QUIC implementation used in this work. In particular, we propose a novel delay metric, and we design and implement a delay-based scheduler, whose design is based on the *Lyapunov* theory.

We address the evaluation of the combination of QUIC, multi-stream, and scheduling policies over NTN, in particular, Low Earth Orbit (LEO) communications. We exploit a novel methodology, which permits mimicking the behavior of the wireless links that characterize the high variability of the underlying connectivity using the *ns-3* framework. The results evince that the proposed scheduling policies outperform the baseline solutions and that the use of multiple streams in QUIC could thus yield lower delays.

All in all, the main contributions of this paper are:

- We assess the performance of the multi-streaming capacity of QUIC over NTN highly variable wireless channels.
- We propose scheduling policies, based on dynamic queue control theory, which aims to harmonize the delay of heterogeneous traffic flows.
- We use a methodology that combines real protocol implementations (virtualization techniques) with simulation platforms, which allows us to precisely mimic the characteristics of the underlying connectivity, and carry out systematic experiments.
- We compare the performance of the proposed schemes with traditional schedulers. The results evince that they harmonize the delay suffered by heterogeneous traffic flows, without hindering the throughput, even under high-load scenarios.

The rest of this manuscript is structured as follows. Section II discusses related works, pointing out how we differ

979-8-3503-1090-0/23

from them. Section III depicts the proposed system model, which allows us to introduce novel, delay-based, schedulers. Section IV describes the methodology that was used to assess the performance of the QUIC protocol and the proposed schedulers, as well as the corresponding scenario. The results are then presented in Section V. Finally, Section VI concludes the paper, and introduces our future research lines.

II. STATE OF THE ART

As was mentioned before, there do not exist previous works proposing scheduling policies that exploit the multi-streaming feature of QUIC. In fact, there are only a few papers that have actually assessed the performance of this characteristic. On the other hand, some works have studied different approaches and mechanisms based on a delay-based metric, introducing different strategies to manage it.

A first group of papers focuses on congestion control solutions. In this sense, the combination of Bottleneck Bandwidth and Round-trip propagation time (BBR) congestion control algorithm and QUIC is proposed to address the shortcomings observed when TCP was used with loss-based congestion control algorithms (CCAs). In [2], implementing a receiver-driven BBR in cellular networks may face challenges such as compatibility with existing network infrastructure, fairness to other flows, and scalability by applying a more appropriate delivery rate calculated at the receiver. Wang *et al.* assess the performance of QUIC, with BBR, in satellite environments, using dedicated network emulation testbeds [3]. The results show that QUIC with BBR outperforms traditional TCP protocols over satellite links, characterized by long Round-Trip Time (RTT) and high packet loss rates. The work also suggests that QUIC with BBR can help to reduce transmission delay and maintain the throughput, even when packet loss rates increase. The authors also propose that further research is needed to improve the effectiveness of BBR over satellite networks.

Although not particularly intended for QUIC, it is worth mentioning some works that have proposed scheduling schemes that base their strategy on the delay. For instance, Hai *et al.* propose in [4] a delay-optimal Back Pressure (BP) routing algorithm called sojourn-time-based BP (STBP), which uses an accumulated sojourn time-based metric to calculate the weight for each scheduling decision. Their results, which are based on a simulation study, evince that the proposed scheme improves the end-to-end delay while ensuring throughput optimality. In our case, we use dynamic queue control and Lyapunov theory to propose a delay-based scheduler, which only uses local information (from internal buffers), to yield lower delays.

Furthermore, in this paper, we focus on single-path communications, but there exist works that have looked at the scheduling problem when multiple paths are available. We can highlight [5], where the authors propose a synchronizing scheduler that balances the transmission rate of data units over multiple paths of communication to reduce protocol delay and waiting time, while maintaining high network throughput. The authors formally analyze the properties of the proposed scheme, whose performance is then assessed over both Multi-Path TCP (MPQUIC) and Multi-Path QUIC (MPQUIC). In addition, Viernickel et al. introduced in [6] a stream-to-path scheduling policy for MPQUIC, able to minimize the HOL effect and to reduce the time required to establish subflows.

On the other hand, we seek to analyze the improvements that the use of the multi-streaming feature of QUIC, together with delay-based scheduling policies, bring over NTN connectivity.

III. SYSTEM MODEL

We modify the operation of the default scheduler included in the QUIC implementation that will be used afterwards, RR, which sequentially selects a stream at each transmission opportunity. First, Fair Queuing (FQ) and Weighted Fair Queuing (WFQ) have been implemented for comparison purposes. With FQ, we fairly distribute the shared capacity between the competing streams, and whenever the demand of a particular one is satisfied, the surplus is again equally distributed between the rest of active streams. In WFQ the behavior is similar, but in this case, the capacity is distributed according to certain weights, which are used to prioritize certain streams, allocating more resources to them. These three schedulers have some limitations. On the one hand, RR and FQ manage all streams alike, not allowing to establish different priorities. On the other hand, WFQ is able to grant some prioritization, but the weights need to be set beforehand, and it would not be therefore able to adapt to channel or traffic variations.

To overcome this limitation, we first propose a BP scheduler, based on Lyapunov‘s theory [7]. This policy aims to ensure stream queue stability for any traffic and capacity conditions (channel and congestion control), as long as the average capacity limits are respected. It is worth noting that, unlike WFQ, the proposed solution does not require any previous configuration.

We model stream scheduling as a queuing system. $Q_k(t)$ corresponds to the occupancy of the buffer application for the k^{th} stream, at any time t, while the scheduling decision for that stream is defined as $\alpha_k(t)$. The transmission capacity of the connection at any time t is modeled as a random variable $\omega(t)$, whose value depends on the channel conditions, and the congestion control mechanism. Then, each queue is updated as:

$$Q_k(t+1) = \max[Q_k(t) - b_k(t), 0] + a_k(t) \tag{1}$$

where $a_k(t)$ and $b_k(t)$ are the arrival and departure variables, respectively. It is worth noting that $b_k(t)$ actually corresponds to each decision variable $\alpha_k(t)$, while $a_k(t)$ is a random variable over which we have no control whatsoever. At each time instant, we have to take a scheduling decision from a set $\mathcal{A}$ that stabilizes the application queues, ensuring that the transmission capacity is not exceeded, $\sum_k \alpha_k(t) \leq \omega(t)\ \forall t$. Exploiting Lyapunov's theory, it can be shown [7] that this problem is solved using the BP algorithm. Hence, at every

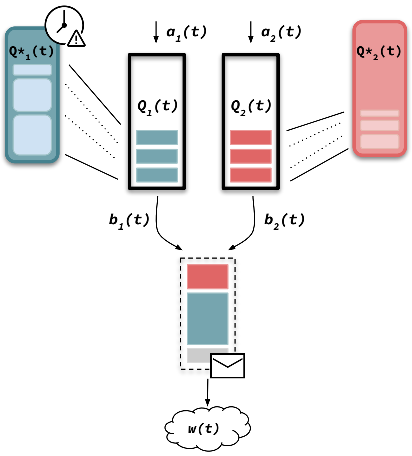


Fig. 1: Delay-based scheduler operation.

slot we have to take a decision that optimizes the following problem:

$$\begin{aligned} \max_{\alpha(t)} \quad & \sum_{k=1}^{N} Q_k(t) \cdot b_k(t) \\ \textbf{s.t.} \quad & \alpha \in A \end{aligned} \tag{2}$$

We now modify the BP algorithm, by introducing a novel delay metric.

Then, with the aim of obtaining the dwell time of each packet, we keep track of their arrival times. For each packet event $a_k(t) \in \boldsymbol{Q}(t)$, we let $d(a_k(t))$ be the time at which packet $a_k(t)$ arrives. The cumulative arrival time of all packets in $Q_k(t)$ is then denoted by:

$$D\left(Q_k(t)\right) = \sum_{p \in Q_k(t)} d(p) \tag{3}$$

As we can see in Figure 1, we consider a virtual buffer, where packets are stored before they are sent, and we monitor their sojourn times. Then, in each slot (t), when a decision is going to be taken, the queue is updated by:

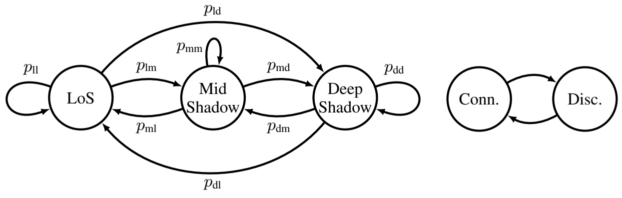
$$\hat{Q}_k(t) = \sum_{p \in Q_k(t)} h_k(t,p) = tQ_k(t) - D\left(Q_k(t)\right) \tag{4}$$

obtaining the sojourn time backlog for each stream k, where $h_k(t,p) = t - d(p)$ represents the sojourn time of p, when it is not removed from the queue. It is worth noting that $Q_k(t)$ corresponds to the number of packets in this queue. Since $b_k(t)$ is the number of departures of stream k, we can denote the cumulative arrival time of those packets as $D\left(b_k(t)\right)$, then the sum of sojourn times of the packet departures is defined by:

$$\hat{b}_k(t) = \sum_{p \in b_k(t)} h_k(t,p) = tb_k(t) - D\left(b_k(t)\right) \tag{5}$$

With the same methodology, we can obtain the sojourn time of the arrival packets, $a_k(t)$. Hence, in each slot, the queue is updated according to:

$$\hat{Q}_k(t+1) = [\hat{Q}_k(t) - \hat{b}_k(t)] + \hat{a}_k(t) \tag{6}$$



(a) (b)

Fig. 2: Markov chain models for LMS (left) and ISL (right) links, used for the performance evaluation.

As in the previous scheduler, in each slot, a decision $\alpha(t)$ is taken, from a set of possible options, $\mathcal{A}$, ensuring to not exceed the transmission capacity $\sum_k \alpha_k(t) \leq \omega(t)\ \forall t$. Therefore, the sojourn time back pressure scheduler can be derived from the following max-weight problem:

$$\begin{aligned} \max_{\alpha(t)} \quad & \sum_{m=1}^{N} \hat{Q}_m(t) \cdot b_m(t) \\ \textbf{s.t.} \quad & \alpha \in A \end{aligned} \tag{7}$$

In practice, the last two problems boil down to selecting, at each scheduling decision, the streams with either higher queue occupancy or delay, in descending order, emptying them until the capacity transmission is reached.

Finally, a maximum delay scheduler has been also implemented, as a simplified version of the previous one. We only contemplate the packet that has the highest delay for all streams. Since all queues follow a First In, First Out (FIFO) policy, we only need to consider the delay of the first packet.

IV. APPLICATION SCENARIO

We assess the performance of the proposed schedulers over NTN, in particular LEO satellite communications. We distinguish two setups: a single Land Mobile-Satellite (LMS) link, between a ground station and a satellite; an end-to-end communication between two ground stations, which entails two LMS links and various Inter-Satellite Link (ISL) links. The latter might suffer from temporary disconnections, due to continuous satellite movement in LEO. We use Markov chains (see Figure 2) to model the two link types: for the LMS link, we start from the work by Fontán *et al.* [8], which considers three different situations/states: (l) Line of Sight (LoS), with ideal propagation conditions; (m) mid-shadowing, where the conditions are worse; and (d) deep-shadowing. For the ISL, we use a two-state chain to capture temporary disconnections. It is worth noting other works [9]–[11] have also exploited Markov chains to model satellite links.

We use the *quic-go* implementation (version 0.15.1), which we modify to integrate the proposed scheduling policies. The platform we use to evaluate their performance is depicted in Figure 3. It blends a real implementation of the QUIC protocol (using virtualization and docker containers) with simulation of the underlying connectivity, exploiting the *ns-3* framework. We modify the operation of the *point-to-point*

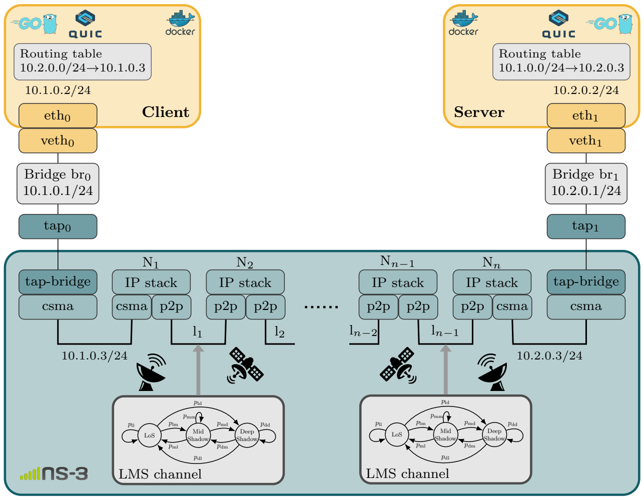


Fig. 3: Diagram of the evaluation platform, which integrates ns-3, docker containers, and the LMS and ISL models.

TABLE I: Scenario setup.

LMS parameters (*Ka* band) from [8, Table XVII]	
LMS rate	$[80, 40, 16]$ Mbps
LMS transition matrix	$\mathcal{P} = \begin{pmatrix} 0 & 0.93156 & 0.068437 \\ 0.34526 & 0 & 0.65474 \\ 0.070012 & 0.92999 & 0 \end{pmatrix}$
LMS sojourn time	$[0.2530, 0.7299, 0.1666]$ s
Average link capacity	45.33 Mbps

link, so that its characteristics can be changed during the experiment, according to the dynamics established by the aforementioned Markov chains. In this way, we are able to mimic the dynamism of both LMS and ISL links.

V. RESULTS

In the first setup we consider a single LMS link. The corresponding parameters are shown in Table I. We use the Ka band, obtaining the transition probability matrix, and the average dwell time in each state from the values found in [8]. We fix the maximum transmission rate, which corresponds to line-of-sight state, to 80 Mbps, while the mid- and deep-shadowing capacities are set to 50% and 20%, respectively. We then extend the results, by comparing the behavior of the scheduler over an end-to-end LEO communication, embracing a single ISL, over which we emulate interruptions.

We have configured QUIC to use two streams in all cases, one of them with twice the load of the other, to illustrate the impact of the scheduling policies over heterogeneous traffics. Traffic is generated following a Poisson model, whose overall rate is established as a ratio of the LMS link capacity (45.32 *Mbps*). Packet lengths are fixed (1000 *Bytes*). Furthermore, the WFQ scheduler is configured with a weight of 2 for the first stream, whose rate is twice the other one.

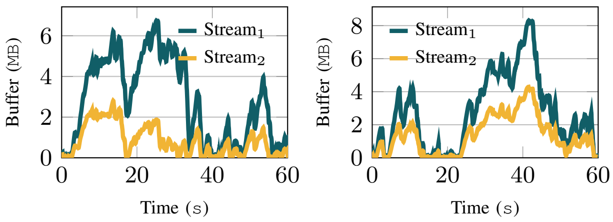
A. Schedulers performance over LMS links

First, Figure 4 shows the evolution of the application buffer, along a single experiment, lasting 60 *s*. We fix the application rate to be 80% of the LMS link capacity (i.e. rather high load), and we plot the buffer evolution for the two streams, and the accumulated sojourn time. It is worth clarifying that the accumulated time is computed as the summation of the sojourn time of all bytes waiting in a stream buffer at a given moment. We observe that BP and delay-based schedulers are able to adapt their operation to the traffic unbalance. BP equalize the buffer occupancy (this is not observed for the two delay-based schedulers), while the two scheduling policies based on delay are able to harmonize the waiting time for the two streams, regardless of their different rates, while BP fails to do so. The behavior of the benchmark solution, WFQ, is as expected, as it was configured to grant twice the capacity to the more demanding stream. We thus see that the proposed scheduling policies, which do not need to be pre-configured, are able to yield a good performance, by adapting its operation to the dynamic changing conditions.

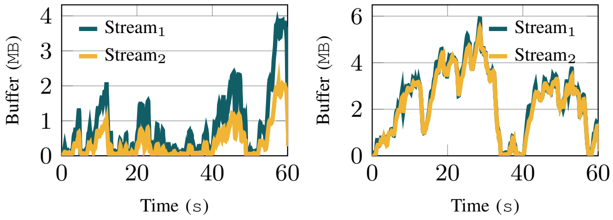
Figure 5 shows the impact of all the schedulers, by measuring both the average delay and the throughput. We average the results of 30 independent experiments, each of them lasting 60 seconds. We consider a low traffic scenario (application rate is fixed at 50% of the LMS link capacity) and a saturation situation, where the application rate equals the LMS link capacity. As can be seen, the static schedulers, RR and FQ, induce a rather unbalanced behavior of the two streams, which is more relevant in the saturation scenario. The default RR implementation shows a slightly fairer distribution, since it is able to use the overall transmission capacity. In addition, the WFQ scheduler is able to equalize the delay in the low load setup, but its behavior is much worse when the load increases. On the other hand, the BP algorithm succeeds in equalizing the delay of both streams, but it induces a higher delay in stream two, since its goal is actually to harmonize the occupancy of both queues. Finally, the proposed delay-based solutions show the expected behavior, as the delay is equalized in both scenarios. It is particularly relevant the fact that even under extremely high load situations (saturation), they are able to yield the same delay for both streams. It is also important to assess whether the observed benefits, in terms of delay, do not hinder the throughput of the communication. For that, Figure 6 shows the throughput observed for the two streams (and the overall) one for the 6 schedulers. Under low load, the impact of the proposed schedulers is almost negligible, since the throughput is almost the same as the one observed for the benchmark schemes. Under the saturation scenario, we see that there is a slight reduction in the performance of the second stream, compared to RR and FQ, but the results also evince that the proposed schedulers, which do not need to be pre-configured, yield higher throughputs than the traditional WFQ, which is not able to adapt its operation to the rapid changes of the underlying connectivity.

B. End to End (E2E) scenario

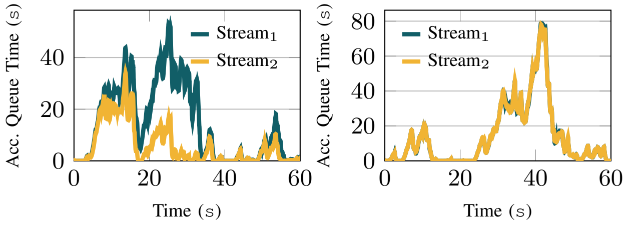
We now broaden the analysis to a E2E scenario, embracing two LMS links and a ISL, which includes temporary disconnections. We use the same application setup as in the previous



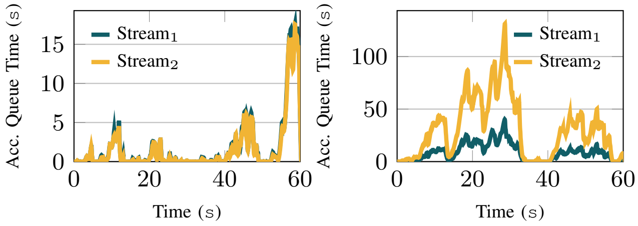
(a) Buffer (WFQ) (b) Buffer (Delay)



(c) Buffer (MaxDelay) (d) Buffer (BP)



(e) Wait. Time (WFQ) (f) Wait. Time (Delay)



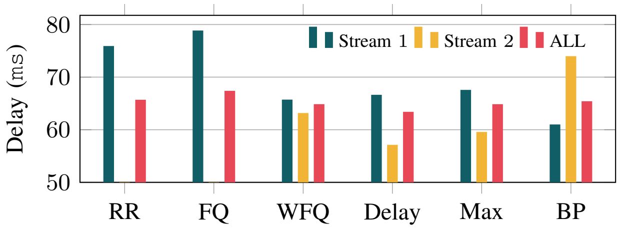
(g) Wait. Time (MaxDelay) (h) Wait. Time (BP)

Fig. 4: Queue buffer evolution of different streams over time, using both different scheduling algorithms and binary sending rates, on a LMS link, under a high load scenario.

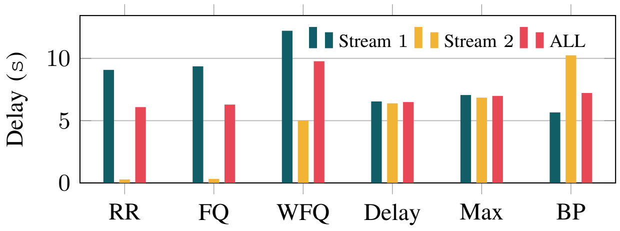
scenario (two streams, one having twice the rate than the other), but we just use the load traffic configuration, since the disconnections at the ISL might effectively reduce the overall end-to-end capacity.

The ISL shifts between connection, with a transmission rate of 80 Mbps, and an average delay of 5 s, and disconnection periods, where the link goes to an inactive status. The average dwell time at the connection situation is 5 s, while it is increased from 0 to 0.5 seconds for the disconnection episodes. In both cases, we use exponentially distributed times at each state. We carry out 5 independent experiments, lasting 300 s for each value of the average dwell time at the inactive situation, to ensure statistically valid performances.

Figure 7 shows the distribution of the average end-to-end delay of both streams (and the overall one) for four scheduling strategies: RR, WFQ, BP and Delay, as we increase the average interruption time at the ISL link. We can again see the good behavior of the proposed scheduling schemes. While the performance is severely affected by the increasing disconnection periods for the two benchmark solutions (RR and WFQ), both BP and the delay-based schedulers are able not only to equalize the delay but to reduce it. The results also evince that their capability of adapting to the changing conditions allows the two proposed schedulers to reduce the variability of the observed delay, thus reducing the perceived jitter at the receiving application.

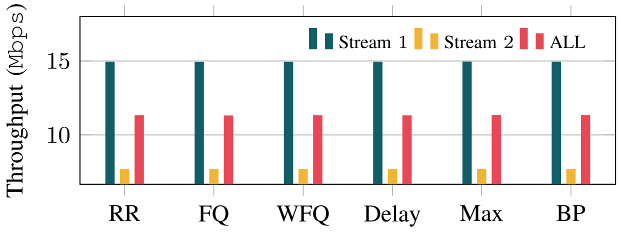


(a) Scenario 1 - Traffic rate at 50%

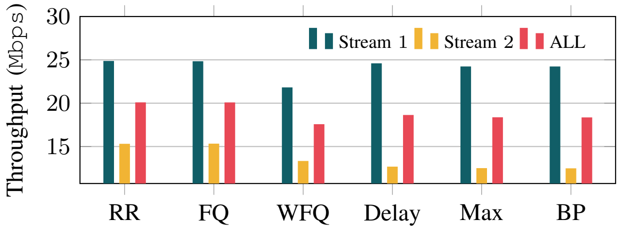


(b) Traffic rate at 100%

Fig. 5: Delay over a LMS link



(a) Traffic rate at 50%

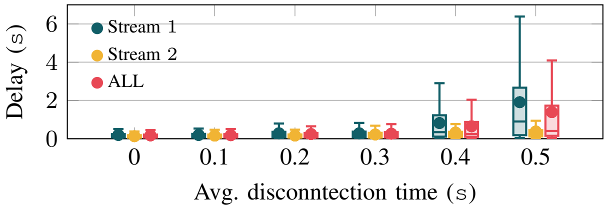


(b) Traffic rate at 100%

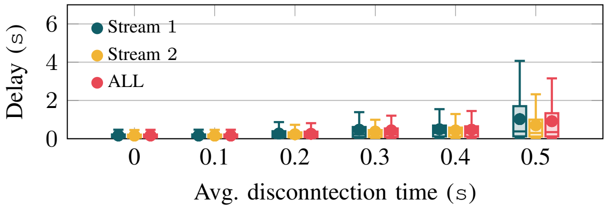
Fig. 6: Throughput over a LMS link.

VI. CONCLUSIONS

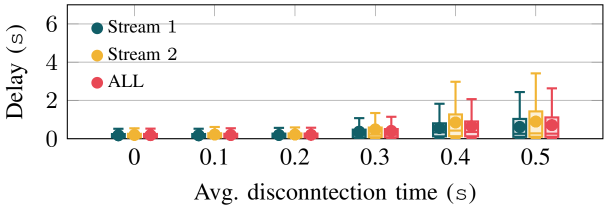
We have proposed a delay-based scheduler for the QUIC protocol, starting from a BP-based scheme, which considers



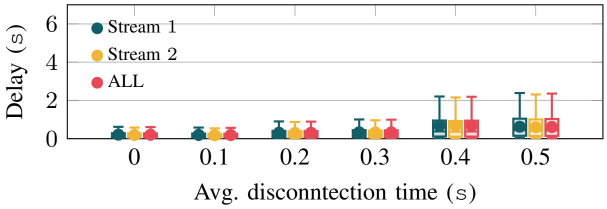
(a) RR



(b) WFQ



(c) BP



(d) Delay

Fig. 7: Delay distribution with RR, WFQ, BP and Delay over an E2E channel with interruptions.

the cumulative waiting time of all packets at the queue. We integrate the proposed scheme in a real implementation of the QUIC protocol, thanks to its multiple streaming feature. By exploiting a novel methodology that combines virtualization techniques, real protocol implementation, as well as realistic modeling of underlying connectivity with simulation frameworks, we have assessed the performance of the proposed scheme, over NTN links, with highly changing characteristics.

We first carried out the analysis over a single LMS link. Traditional and static strategies (RR and FQ) exhibited a sub-optimal behavior for unbalanced traffic conditions. In addition, although the WFQ scheduler was able to harmonize the delay under certain characteristics, it is not able to adapt to rapid connectivity changes and its performance over high load situations was impacted. The proposed schedulers yield a much better behavior since they were able to harmonize the delay, even under relatively high load situations. In addition, they do not severely hinder the throughput of the communications.

We also increased the complexity of the topology, considering an E2E communication, with an ISL having periods of connectivity interruptions. The delay-based scheduler is able to equalize the delay of the two streams and reduce both its average value and variability, even for longer disconnection periods.

In our future work, we plan to extend the characterization to more complex environments, introducing, for instance, background traffic over the ISL. We will also study the interplay of the proposed schedulers with other QUIC mechanisms and extensions, such as congestion control algorithms and multipath communications.

ACKNOWLEDGMENT

This work has been supported by the Spanish (Ministerio de Economía y Competitividad) and Cantabria (Consejería de Industria, Empleo, Innovación y Comercio) Governments, with Fondo Europeo de Desarrollo Regional, FEDER with the project SITED: Semantically-enabled Interoperable Trustworthy Enriched Data-spaces (PID2021-125725OB-I00) and the program “Ayudas a proyectos de investigación con alto potencial industrial de agentes tecnológicos de excelencia para la competitividad industrial TCNIC”. The work of Fátima Fernández was supported by the Basque Government with the EGIA project (KK-2022/00119), Elkartek program.

REFERENCES

[1] J. Iyengar and M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport,” RFC 9000, may 2021. [Online]. Available: https://rfc-editor.org/rfc/rfc9000.txt

[2] H. Haile, K.-J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunstrom, “Wip: Leveraging quic for a receiver-driven bbr for cellular networks,” in *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2021, pp. 252–255.

[3] Y. Wang, K. Zhao, W. Li, J. Fraire, Z. Sun, and Y. Fang, “Performance evaluation of quic with bbr in satellite internet,” in *2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2018, pp. 195–199.

[4] L. Hai, Q. Gao, J. Wang, H. Zhuang, and P. Wang, “Delay-optimal back-pressure routing algorithm for multihop wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2617–2630, 2018.

[5] M. Morawski and P. Ignaciuk, “A synchronizing scheduler for reduced protocol delay in multipath transmission,” in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 553–558.

[6] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, “Multipath quic: A deployable multipath transport protocol,” in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[7] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[8] F. P. Fontan, M. Vazquez-Castro, C. E. Cabado, J. P. Garcia, and E. Kubista, “Statistical modeling of the lms channel,” *IEEE Transactions on Vehicular Technology*, vol. 50, no. 6, pp. 1549–1567, 2001.

[9] A. Chen, C. Chang, and Y. Yao, “Performance evaluation of arq operations with obp and inter-satellite links: delay performance,” in *IEEE 54th Vehicular Technology Conference Proceedings (VTC Fall 2001)*, vol. 4, 2001, pp. 2346–2350 vol.4.

[10] R. Hermenier, C. Kissling, and A. Donner, “A delay model for satellite constellation networks with inter-satellite links,” in *2009 International Workshop on Satellite and Space Communications*, 2009, pp. 3–7.

[11] Y. Zhu, M. Sheng, J. Li, and R. Liu, “Performance analysis of intermittent satellite links with time-limited queuing model,” *IEEE Communications Letters*, vol. 22, no. 11, pp. 2282–2285, 2018.