Computer Networks 236 (2023) 110008

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Realistic assessment of transport protocols performance over LEO-based communications

Fátima Khan [a,*], Cristina Hervella [a], Luis Diez [a], Fátima Fernández [b], Néstor J. Hernández Marcano [c], Rune Hylsberg Jacobsen [d], Ramón Agüero [a]

[a] Communications Engineering Department, Universidad de Cantabria, Santander, Spain
[b] IoT and Digital Platforms Department, Ikerlan Technology Research Centre, Arrasate/Mondragón, Spain
[c] Gatehouse Satcom, Nørresundby, Denmark
[d] Aarhus University, Aarhus, Denmark

ARTICLE INFO

Keywords:
QUIC
LEO
Scheduling
LMS
TCP

ABSTRACT

We study the performance exhibited by the transport protocols, Transport Control Protocol (TCP) and QUIC, over realistic satellite networks. We propose a novel methodology, which combines real implementation (exploiting virtualization techniques) and simulation, to carry out systematic and repetitive experiments. We modify the default operation of the *ns-3* framework and we integrate the dynamism that characterizes satellite communication links, particularly Low Earth Orbit (LEO). We carry out a thorough assessment over different setups, changing the operating frequency band and packet buffer lengths. In addition, we ascertain the impact of using the multi-streaming feature that QUIC integrates. The results show that QUIC yields lower delays than TCP, although it might suffer from higher jitter in particular setups. In addition, the results evince that using multiple streams in QUIC does not yield a relevant gain for the default Round-Robin (RR) scheduler. We propose more appropriate scheduling strategies, which are able to yield better performances with unbalanced traffic. Even if the behavior of transport protocols over non-terrestrial-networks might not be always appropriate, the obtained results evince that QUIC can definitively bring benefits when compared to TCP. Furthermore, we have shown that optimal scheduling policies yields a fairer performance when using multiple flows, having unbalanced traffic loads.

1. Introduction

The presence of Non-Terrestrial Networks (NTN) in forthcoming cellular technologies, 5G and Beyond 5G (B5G) networks, is expected to be remarkably more relevant than in previous systems. They are considered as pivotal elements of such network deployments, since they will allow providing connectivity services to remote areas (satellite) as well as quickly deploying communication resources in certain places, using Unmanned Aerial Vehicle (UAV). In short, the impact of aggregating NTN networks to more traditional cellular architectures is beneficial in terms of reliability, scalability, coverage, and service continuity.

On the other hand, these benefits come with various challenges. For instance, it is well known the poor performance exhibited by traditional transport protocols, most notably Transmission Control Protocol (TCP), over wireless links. This is expected to become more prominent with the advent of 5G and B5G systems [1], including NTN. Furthermore, legacy transport protocols do not behave well with the traffic patterns that characterize novel services.

In order to address the aforementioned shortcomings, the scientific community is making an effort to design and develop new transport protocols, which could overcome some of the limitations exhibited by TCP. One of the most relevant examples of such efforts is the QUIC protocol, originally promoted by Google, which has been recently standardized [2]. Among its advantages, we can highlight that QUIC enables multi-streaming, which avoids Head-of-line (HOL) blocking. In this sense, when a loss affects a particular stream, the rest of active streams do not get affected.

With the aim of increasing system efficiency, *scheduling* algorithms emerge to manage resources, increasing the performance and reducing the delay. During a QUIC connection, the capacity is first limited by the congestion control mechanisms, which establish when there is a transmission opportunity. Then the scheduling policy decides which stream frames will be selected to build a QUIC packet. The default operation of the QUIC protocol is the well-known RR strategy, where packet data

* Corresponding author.
E-mail address: khanf@unican.es (F. Khan).

https://doi.org/10.1016/j.comnet.2023.110008
Received 14 April 2023; Received in revised form 24 July 2023; Accepted 30 August 2023
Available online 4 September 2023

slices are equally assigned to each stream, always following the same order, so handling all streams without any priority whatsoever.

In this paper we address the evaluation of the combination of QUIC and NTN and, in particular, we study the performance of this transport protocol over Low Earth Orbit (LEO) communications. By exploiting a novel methodology, which permits mimicking the behavior of the wireless links that characterize the underlying connectivity, we can extract conclusions on the performance of such protocol over these networks. We first compare the performance of QUIC and TCP, and then we study the behavior of different scheduling strategies, to ascertain the potential benefits that they might bring.

The contributions of this paper are:

- We introduce a novel methodology that changes the legacy behavior of the *point-to-point* link in *ns-3* to adapt its features during an experiment, according to different events. This permits us to capture the dynamism of NTN underlying links. In addition, the proposed framework can easily deploy custom topologies, embracing several Inter-Satellite Link (ISL) links, with different features. In this sense, we can include disconnection times, which might be common, for instance, for crossed-ISL.
- We use the aforementioned solution, as well as virtualization techniques (Docker containers), to assess the performance of the QUIC protocol over LEO satellite network scenarios. We consider different configurations, changing the buffer lengths, the number of streams, the distribution of capacity between them, and we compare the performance with that obtained when using TCP.[1] At a second glance, we analyze the performance over a NTN scenario with various hops and different link configurations, which leverage a more realistic setup.
- We also look at the application layer, by configuring multiple streams to mimic web traffic, where one connection is able to send different objects. We consider three schedulers besides the one included in QUIC, RR; we implement the well-known Fair Queuing (FQ), and Weighted Fair Queuing (WFQ) schedulers, and we propose a novel one, based on Lyapunov's theory. The performance of such schedulers is then analyzed in terms of buffer occupancy, throughput, and delay.

The rest of the paper is structured as follows: Section 2 discusses the related work, pointing out how this paper is positioned against it. Section 3 depicts the general scenario that we consider, and Section 4 introduces the proposed methodology. The scheduling strategies that we integrate into the QUIC protocol are described in Section 5, while Section 6 discusses the results obtained after an extensive experiment campaign. Finally, we conclude the paper in Section 7, where we provide an outlook of our future work.

2. Related work

In the last years the interest for NTN, in general, and LEO constellations, in particular, has remarkably increased. One of the main reasons is the recent interest in exploiting them as an enabling technology for 5G and beyond 5G networks [3,4]. Liu et al. introduce in [5] a Satellite Access Network (SAN), and they pay special attention to new research lines leveraged by this communication paradigm. Apart from the architectural perspectives, there exist works that analyze the technical convergence of terrestrial and non-terrestrial networks. In this sense, Leyva-Mayorga et al. provide in [6] an overview of the integration of LEO constellation within the 5G/B5G access network, and they focus on characterizing the physical links of the resulting SAN.

[1] The developed evaluation framework, as well as the data that was obtained during the experiments carried out for this paper have been made publicly available in a GitHub repository: https://github.com/tlmat-unican/Lightweight-ns-3-link-simulation.

Standardization bodies have also started considering satellites as enablers for B5G networks. In [7] the authors survey current efforts in the integration of NTN and 5G technologies. The paper first describes current 3GPP standardization approaches, to afterwards discuss main challenges, such as: (1) architectural options; (2) network management techniques; and (3) required modifications in users' terminals. Similarly, Darwish et al. study in [8] how standardization bodies define LEO-based radio access network, paying special attention to 3GPP New Radio (NR) standards. There are also papers proposing technical solutions for LEO-based access networks. The authors of [9] introduce an uplink scheduling solution for massive Machine Type Communications Narrow Band Internet of Things (mMTC NB-IoT) services supported by LEO constellations. It does not require modifications at the NB-IoT devices, and features techniques to reduce the differential Doppler shift.

As can be seen, the aforementioned papers set the grounds of how LEO constellations can be used for service provisioning in 5G/B5G networks. In this context, a second group of works focuses on the capabilities of SAN in general, and LEO in particular, to enable satellite-assisted edge/cloud services, which is an application scenario of interest in our work.

Xie et al. provides in [10] an overview of Satellite-Terrestrial integrated Edge Computing Networks (STECN), which combines satellite-terrestrial networking with edge computing to improve the Quality of Service (QoS). The paper analyzes design principles, key functionalities, as well as some challenges of this approach. In the same line, the authors of [11] propose the combination of Mobile Edge Computing (MEC) with LEO, promoting the so-called LEO Satellite Edge Computing (LSEC), and they investigate resource allocation and computation offloading in the LSEC network. Based on the need of providing services to ground users, Zhang et al. exploit in [12] Orbital Edge Computing (OEC) by developing an algorithm to allocate resources, which shows a good performance in terms of computational cost. Zhou et al. focus on multicast-based multimedia streaming in the next generation of wireless and mobile networks [13]. They also discuss the utilization of hierarchical non-terrestrial networks, consisting of LEO satellites, High-Altitude Platforms (HAPs), and User Equipment (UE). However, existing multi-cast protocols for NTN cause long delays and jitter, due to packet loss events, and [13] proposes a reliable multi-cast protocol, namely Overhearing NOMA Repairing Reliable Multi-cast (ONRM), to reduce delay.

Other works assume a full integration of satellites and existing terrestrial networks. This is the case of [14], which considers a topology with base stations and LEO-based small cells, as well as terrestrial and satellite backhaul links. Over such topology, the authors formulate an optimization problem to minimize the whole network energy consumption, while keeping QoS constraints.

Although these works share the application framework with ours, the goal is different and complementary. Most of them aim to provide network intelligence to improve the integration with LEO satellites. On the other hand, we pursue the performance evaluation of existing (TCP) and coming (QUIC) transport protocols over these networks. We thus now focus on works that have studied the behavior of QUIC and how it stands compared to TCP.

For instance, Shreedhar et al. assess in [15] QUIC performance for web, cloud storage and video workloads in uncontrolled environments. However, the authors do not pay much attention to the underlying access network. Similarly, Qian et al. study in [16] QUIC's performance. In this case, the work focuses on the interaction with LTE-A access networks. The authors study the behavior of different congestion control algorithms, and they extend the regular QUIC operation with its multi-path functionality, by developing a prototype that allows dual WiFi and LTE-A communications. Closer to the edge/cloud computing scenario, Dizdarević and Jukan benchmark QUIC performance for IoT-edge-cloud continuum in [17]. The authors perform the evaluation over two different scenarios: (1) IoT-edge, and (2) IoT-edge-cloud. The assessment, which uses Raspberry Pi devices, focuses on the cloud architecture, while little attention is paid to the access network.

As can be observed, there is a gap in the evaluation of transport protocols in general, and QUIC in particular, over SAN and, more specifically, over scenarios that integrate terrestrial and non-terrestrial networks. There are only a few works that have addressed this evaluation.

In the past, Tsonuda et al. [18] analyzed the performance of Stream Control Transmission Protocol (SCTP) over LEO networks, using adaptive multi-stream. Besides, the multi-homing feature of SCTP has been also exploited over LEO networks in [19] to improve handover procedures. If we focus on QUIC, Yang et al. tackle in [20] the interplay between LEO networks and QUIC, but from a different perspective. They focus on Multipath QUIC (MPQUIC), and they propose a performance model for LEO (handover, outage, etc.). Martin and Khademi [21] discuss the suitability of Bottleneck Bandwidth and Round-trip propagation time (BBR) congestion control for QUIC over Geostationary Equatorial Orbit (GEO) satellite communication networks. The authors explore the potential of high-throughput GEO satellites, and how the increasing deployment of user-space QUIC protocol and BBR congestion control can impact Internet paths traversing satellite links, due to the fact that the transport performance depends on the choice of congestion control and QUIC implementation. The work that seems closer to ours is the one carried out by Yang, Li and Wu [22]. They model the wireless channel with Satellite Tool Kit (STK) through Matlab and then use the corresponding traces to analyze QUIC's performance over LEO constellations. They also exploit emulation over real devices, but they use high-capacity wired links, whose characteristics (delay and loss rate) are statistically modified based on the previous analysis. We consider that our evaluation approach is rather different, since: (1) they do not consider buffer lengths, and so disregard self-inflicted delay; (2) we assume that losses at lower layers are captured by channel capacity fluctuations, capturing the fact that a Signal to Interference & Noise Ratio (SINR) reduction would lead to lower capacity, so that our losses mostly occur due to buffer saturation; (3) they focus on small files (web pages of a few tens of kB) while we analyze the performance for large files transfer; (4) last, by considering channel capacity, we are able to analyze the impact of the application rate.

Finally, it is worth mentioning that not many studies have addressed the analysis of scheduling techniques to orchestrate the multi-stream functionality provided by QUIC. In this sense, it is worth mentioning the work of Cui et al. [23]. In this work, the authors exploit QUIC multi-streaming to mitigate the delay of Moving Picture Experts Group (MPEG) Dynamic Adaptive Streaming over HTTP (DASH) video streaming caused by head-of-line blocking. They designed a deadline-based scheduler at the DASH level; if a video segment does not reach the receiver within the specified time, the proposed scheme switches to a simple transmission, disabling QUIC streams and prioritizing the first requested segment. As can be observed, their solution is complementary to ours, since the scheduler presented herewith is at QUIC level and so it would be activated after the one in [23]. In addition, our proposed solution is not bounded to any specific application, so that both schemes could be complementary. Finally, there are few papers that have tackled scheduling techniques [24,25], but in the scope of multi-path communications, not as a means to manage the multiple streams a QUIC connection might create.

All in all, we believe that this work fills an existing gap in the evaluation of transport layer protocols, in particular QUIC, over novel access network topologies, embracing LEO satellite constellations.

3. Application scenario

We consider communications over LEO satellite networks. The underlying topology embraces two distinct link types: (1) LMS, which is established between the ground station and the first satellite and between the last satellite and the other end ground station; and (2) ISL, between two consecutive satellites in the corresponding constellation.

Under the aforementioned topology, two setups are investigated. In the first one, communication is established between a satellite and a ground station (or vice versa) comprising a single LMS link. In this configuration information is received from (or sent to) a satellite, for instance, if the LEO network is used to deploy edge computing. On the other hand, we will also consider end-to-end communications between two ground stations. The main difference between both setups is that in the latter, the information must traverse two LMS links as well as a number of ISLs.

The LMS model is taken from the work by Fontán et al. [26]. In that work an extensive measurement campaign was carried out, embracing several scenarios, environments, and configurations. The authors observed that the link between ground stations and satellite (or vice versa) may experience different conditions, caused (among others) by the mobility that characterizes LEO constellations. Based on the measurements, the authors model the channel variations using a three-state Markov Chain, whose states correspond to: (l) Line of Sight (LoS), with ideal propagation conditions; (m) mid-shadowing, where the conditions are worse; and (d) deep-shadowing, with the most severe connectivity conditions. It is worth noting that approaches based on Markov chains have been also used in other works, such as [27,28] or [29], to model satellite links.

The model adopted in [26] for the LMS link is depicted in Fig. 1(a). It represents a discrete Markov chain with three states, which correspond to the line-of-sight, mid-shadowing and deep-shadowing conditions, as well as the corresponding transitions between them. In their paper, Fontán et al. also include the parameters of the transition matrix for several channel instances and different operating frequency bands, resulting in up to 85 different configurations. In order to reflect the characteristics of the LMS links, we have modified the legacy operation of the model, using a continuous Markov chain instead of the original discrete approach, to allow channel transitions to happen at any point in time. Then, the sojourn time at each state follows an exponential distribution, with mean [30]:

$$\overline{\tau_i} = \frac{\delta}{1 - p_{ii}} \tag{1}$$

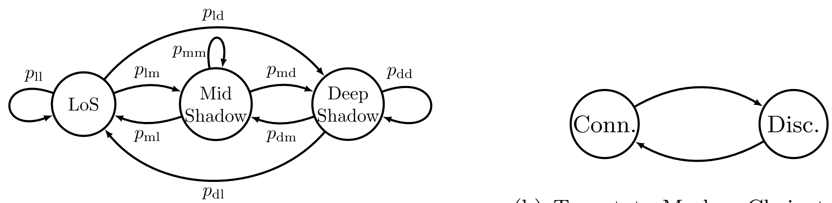
where the sub-indexes $i \in \{l, m, d\}$ indicate the corresponding channel state. line-of-sight (l); mid-shadowing (m); and deep-shadowing (d). In addition, p_{ii} holds for the probability of staying at the same state and δ is the corresponding slot time, when channel transitions might happen. Besides, in this continuous version of the LMS channel model, whenever a particular state is left, the next one is selected with probability:

$$\alpha_{ij} = \frac{p_{ij}}{p_{ij} + p_{ik}} \tag{2}$$

where $i, j, k = \{l, m, d\}$ and $i \neq j \neq k$.

As already mentioned, the model for the LMS link is based on the work by Fontan et al. [26]. They proposed an empirical model, which considers a discrete Markov Chain, whose parameters were based on real measurements. The modification that we use in this work was originally proposed in [30], where we showed that the behavior of the underlying links, which were based on continuous Markov Chains, captures quite well the one exhibited by the original model. The benefit of the time-based configuration that we included is that it allows using such model for any type of traffic pattern. Opposed to our previous work, we exploit this characteristic herewith to study the impact of the dynamic nature of these links over the performance of transport layer protocols and mechanisms. For that, we integrate the proposed model in the framework of the *ns-3* simulator, as we depict in Section 4.

On the other hand, the ISLs are modeled with a 2-state Markov Chain, as depicted in Fig. 1(b). In this case, it is assumed that the connection between two satellites has a constant rate (*Connect* state), but there can be temporary disconnections (*Disconnect* state), due to lack of direct visibility between satellites, which are constantly moving.



(a) Three-state Markov Chain for the LMS channel

(b) Two-state Markov Chain to emulate temporary ISL disconnections

Fig. 1. Markov chain models for LMS and ISL links, which have been used for the performance evaluation.

We also use a continuous process to model transitions between both states, regardless of the particular traffic patterns. This approach was proposed by Zhu et al. in [29]. Thanks to this, and to the possibility of having background traffic, we could also mimic certain LEO connectivity features, such as crossed-ISL links, that appear due to the continuous movement that characterizes these nodes.

It is important to remark that, although they share some characteristics, both models are not alike the well-known Gilbert–Elliot, since we used the different states to modify the capacity of the channels, rather than to create packet erasures.

4. Analysis methodology

This section describes the analysis approach adopted for the evaluation of transport solutions over LEO-based communications. We will first describe the overall methodology and the platform that was used. Then, we will focus on the implementation of the satellite links over *ns-3*, which is exploited to emulate the characteristics of the underlying connectivity.

4.1. Evaluation test-bench

The platform used for the evaluation is depicted in Fig. 2. It combines a real implementation of the transport protocols with simulated connectivity. In this sense, the channel modeling focuses on the capacity variation as a consequence of long-scale fading. As can be seen in the figure, the applications that generate and consume traffic, either using QUIC or TCP, are packaged into Docker containers. We have used different Docker networks to isolate one application from each other, so that they become reachable only through the virtual bridges generated by Docker. The applications, both client and server, are implemented in `GO` programming language. For the evaluation we use the own host TCP implementation, as well as the `quic-go` 0.15.1[2] version.

In order to ensure that all the traffic exchanged between the applications goes through the simulator, the containers are configured to use one simulated node as the default gateway for the outgoing traffic. As can be seen in Fig. 2, the left-hand container and the first node N_1 share the same IP sub-network, so that they are reachable. It is worth noting that the Carrier Sense Multiple Access (CSMA) link that is used to connect the *tap-bridge* with N_1 is provided with infinite transmission capacity and no delay, so it becomes transparent to the traffic. This way, the left-hand container, together with N_1, are used to mimic a ground station. Then, the LEO satellites are emulated through nodes, and the satellite links, either LMS or ISL, with point-to-point links having different properties. Since the main focus of this work is the transport protocol performance, we foster a lightweight modeling approach, which does not aim to precisely capture the lower layer mechanisms and characteristics. We assume that there exists a physical layer able to manage access to a shared medium, regardless of the underlying technology (CDMA or FDMA). We also consider that every link, for a particular situation, has a certain capacity, which is shared between eventual competing flows.

4.2. ns-3 channel implementation

The LMS channel has been integrated by modifying the point-to-point network device implementation of *ns-3*. Such devices, which are included in the `PointToPointNetDevice` class, consist of a buffer that receives traffic from the protocol stack, and an interface characterized by a certain transmission rate, RTT, and error probability. Thus, the simulator originally uses the transmission rate and packet length to schedule the transmissions of packets through the interface. In addition, the transmission rate is usually set at the beginning of the simulation and it remains unaltered afterwards.

On the other hand, the `PointToPointNetDevice` class has a public method (`SetDataRate`) that allows changing the rate during the simulation. However, we noticed some issues when using it in extreme cases. In particular, when the transmission is set to 0 (for instance if we want to emulate a temporary disconnection) to be afterwards increased, the device was not able to restart the transmission of queued packets. To overcome this limitation, whenever there is a rate change, and the previous value was 0, we check the occupancy of the buffer. If there were any waiting packet, its transmission would be started again (function `TransmitStart` is called), so that the device recovers its normal operation. In any case, it is worth noting that the transmission rate change is not preemptive, and packet transmission events are not thus re-scheduled upon a rate change.

For each link in the scenario, one independent Markov Chain is implemented to modify the transmission rate in a point-to-point connection. This way, when a link changes to a new state, the simulator calls the function `SetDataRate` of the corresponding point-to-point device and schedules an event to leave the state according to the dwell time distribution. This process is repeated until the end of the simulation, regardless of the link and whether or not it is currently being used.

In order to simplify the configuration of scenarios, we have developed a set of utilities to create a custom topology based on a generic *ns-3* scenario. It is fed with a JSON configuration file that completely specifies the network topology. In particular, it defines the number of links in the scenario, as well as their type and features. So far, we have implemented link types that adopt the aforementioned 3-state and 2-state Markov Chain models, as well as ideal links with a static configuration. In addition, the simulation of crossed ISLs can be achieved by adjusting link properties, incorporating background traffic, and simulating temporary disconnections. By configuring each link with its specific properties, it becomes possible to accurately capture different constellation geometries. We plan to exploit these features of the proposed methodology in our future work.

For simplicity, the particular settings of the models (*i.e.* transition matrices, average dwell times, among others) are implemented in the *ns-3 C++* code, and the configuration file indicates the model that needs to be loaded for every link. In our future work, we will generalize the implementation, so that the link models could be also fully defined through configuration. We will also analyze the possibility to include random large-scale fading.

In all cases, the configuration file also establishes the value of the legacy parameters of the point-to-point link, and devices, which do not

[2] https://github.com/lucas-clemente/quic-go

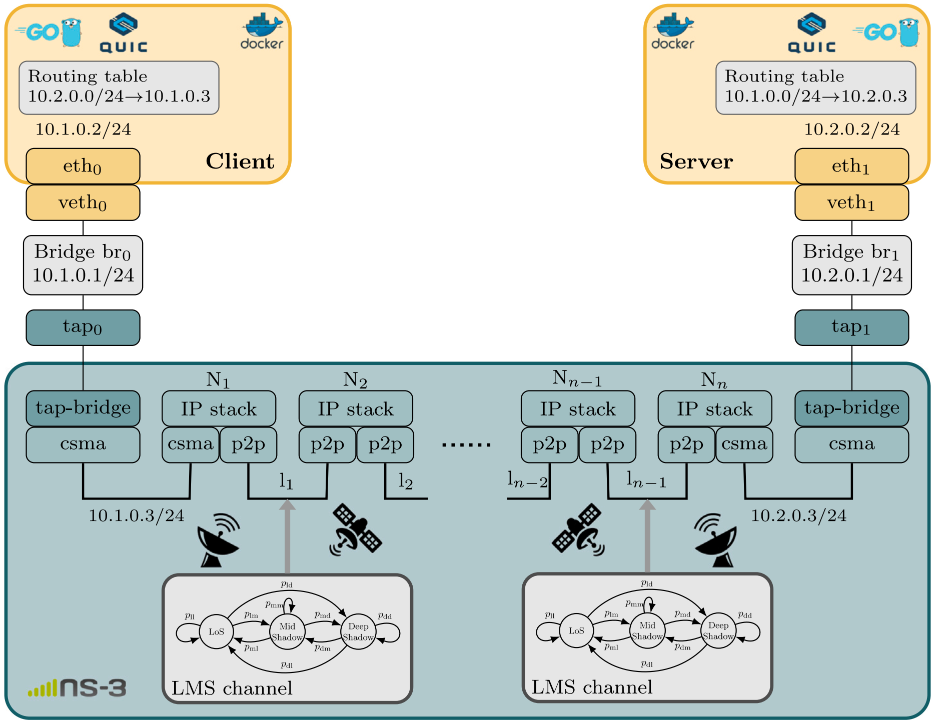


Fig. 2. Diagram of the evaluation platform, which integrates *ns-3*, Docker containers and the LMS and ISL models.

depend on the adopted model: error rate, delay, MTU, and size of the buffer.

The modified *ns-3* implementation with the link models and automation utilities has been made publicly available in a GitHub repository.[3]

5. Multi-stream QUIC application

Although QUIC and TCP are connection oriented protocols, and they both provide functionalities to ensure reliable end-to-end communications (mainly congestion control, flow control, in order delivery), they treat traffic flows in different ways. In the case of TCP, there is one traffic flow per connection, while QUIC introduces the *stream* concept. QUIC streams are independent traffic flows sent over the same connection. Within each stream in-order-delivery is ensured, while the congestion control is jointly applied to the overall connection. Thus, the connection transmission capacity is shared among streams using a scheduler. In the following we describe the architecture of the QUIC application that has been used, and the proposed scheduler proposed. In addition, we briefly describe the legacy policies that will be used for performance comparison.

5.1. QUIC application architecture

As mentioned before, we use `quic-go` version 0.15.1. This QUIC implementation does not provide buffering for application data. In this sense, when the application sends data of a particular stream through a QUIC connection, it keeps a single MTU for each stream before it is sent to the User Datagram Protocol (UDP) buffer, as illustrated in Fig. 3. In addition, similar to TCP sockets, send functions are blocking. For this reason, when the application rate of one stream exceeds the capacity provided by the combination of congestion-control and scheduling, the application stalls, thus impacting the actual traffic generation.

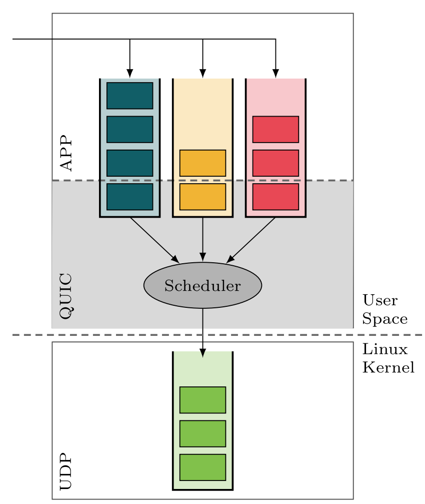


Fig. 3. Architecture of the proposed application, featuring a buffer mechanism.

In order to avoid this behavior, we have implemented a QUIC application that features a simple buffering mechanism, by using two independent threads for each stream. The first one is responsible for generating traffic, according to its configuration, and allocate it in the corresponding stream buffer. In parallel, the second thread gets data from the buffer and sends it through the QUIC connection as soon as possible. To avoid race conditions, stream buffers are protected by locks, so that threads cannot read and write simultaneously. The resulting application architecture is depicted in Fig. 3.

5.2. Stream schedulers

The QUIC implementation used in this work uses RR as its default scheduler. For each transmission opportunity, it selects a stream, following a sequential order. Besides, when a stream does not have

[3] https://github.com/tlmat-unican/Lightweight-ns-3-link-simulation

enough bytes to fill the packet, the scheduler selects the next one, in order.

We have also implemented FQ and WFQ for comparison purposes. Both schedulers assume that there are C resources to be shared among N streams; in our case the resources are the number of bytes of the QUIC packet. Any stream i has a number of bytes to send r_i. With FQ all streams are given the same number of resources ($\frac{C}{N}$). In addition, if there are streams demanding fewer resources, $r_i < \frac{C}{N}$, their demand is thus fully satisfied, and the remaining resources are equally distributed among the rest of streams. The WFQ scheduling is similar to FQ, but in this case the assignment for a stream i is weighted by a pre-configured parameter, $c_i = \frac{w_i \cdot C}{\sum_{\forall i} w_i}$. This allows prioritizing particular streams.

The aforementioned schedulers exhibit some limitations. On the one hand, RR and FQ cannot prioritize particular streams, since their goal is to manage all streams alike. On the other hand, WFQ is able to provide some prioritization, but the weights need to be configured beforehand, and it would not be therefore able to adapt to traffic variations.

In order to overcome these limitations, we propose a BP based scheduling, which is based on Lyapunov's theory [31]. This policy aims to ensure stream queue stability for any traffic and capacity (channel and congestion control) conditions, provided capacity limits are respected in average. It is worth noting that, different to WFQ, the proposed solution does not require pre-configuration.

We model the stream scheduling as a queuing system. In this sense, $Q_k(t)$ corresponds to the occupancy (in bytes) of the buffer application for the kth 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 congestion control. Then, each queue is updated as:

$$Q_k(t+1) = \max[Q_k(t) - b_k(t), 0] + a_k(t) \quad (3)$$

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 $\alpha_k(t)$, while $a_k(t)$ is an arbitrary 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 [31] that this problem is solved using the BP algorithm. Hence, at every slot we have to take a decision that optimizes the following problem:

Problem 1.

$$\max_{\alpha(t)} \sum_{k=1}^{N} Q_k(t) \cdot b_k(t) \quad (4)$$

$$s.t. \quad \alpha \in \mathcal{A} \quad (5)$$

As can be observed, in Problem 1 queue occupancy has constant coefficients in each time slot, leading to a constrained linear problem. In practice, the problem solution boils down to selecting the streams with the highest queue occupancy in descending order at each scheduling decision. These are then processed in descending order, until the transmission capacity for the corresponding QUIC packet is reached. If we look at it as a max-weight problem, the optimization would aim at maximizing the previous objective function, which is directly influenced by the value of $b_k(t)$.

6. Performance evaluation

This section discusses the results that were obtained during the evaluation of TCP and QUIC, including the various scheduling strategies, over LEO scenarios. Their main configuration parameters are summarized in Table 1. As can be seen, we use two different bands, *Ka* and *S*, each of them characterized by the transition probability matrix, the average dwell time in each state, and the corresponding transmission rates. The maximum transmission rates, which correspond to line-of-sight situation, are 4 and 80 Mbps for *S* (2–4 GHz) and *Ka* (26.5–40 GHz) band, respectively. Additionally, mid- and deep-shadowing capacities are set to 50% and 20% of such maximum capacity.

Table 1
Scenario setup.

Ka band LMS parameters from [26, Table XVII]	
LMS rate	[80, 40, 16] Mbps
Average link rate	45.33 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
δ	100 ms
S band LMS parameters from [26, Table XVII]	
LMS rate	[4, 2, 0.8] Mbps
Average link rate	2.32 Mbps
LMS transition matrix	$\mathcal{P} = \begin{pmatrix} 0 & 0.94076 & 0.059243 \\ 0.77084 & 0 & 0.22916 \\ 0.49418 & 0.50582 & 0 \end{pmatrix}$
LMS sojourn time	[0.5485, 0.4992, 0.3529] s
δ	100 ms
Application and buffer	
Buffer sizes	[7, 15, ∞] Packets
Packet size	1000 Bytes

In the following we will first compare the performance of QUIC and TCP over LMS links, using the default configurations. Then, we will focus on the QUIC protocol, analyzing the behavior of the schedulers described Section 5 over LMS links, considering asymmetric traffic flows. This will be further extended by analyzing the impact of ISL disconnections over an end-to-end scenario.

6.1. TCP and QUIC performance over single a LMS link

Before discussing the average performances of TCP and QUIC over LMS links, Fig. 4 illustrates the variability exhibited by a particular LMS channel instance, and its impact over the buffer occupancy, congestion window (CWND), and traffic delay. Here, the buffer refers to the one in the *point-to-point* device (see Fig. 2), and its occupancy thus indicates the adaptation of the congestion control to the channel variation. The results are obtained from a single QUIC transmission, with one stream and constant application traffic rate, over the *Ka* band with infinite buffer capacity. The background colors of Fig. 4 represent the channel state, and the lines the instantaneous evolution of the different parameters.

As can be seen in Fig. 4(a) the buffer size increases during the connection, which entails a file transfer, with slow variations that are correlated to the congestion window, and fast variations that depend on the particular channel state. At the same time, we observe that the packet delay follows the same trend as the buffer occupancy, showing large variation (*jitter*) when the channel shifts between states. On the other hand, Fig. 4(b) shows the evolution of the congestion window in the very same experiment. As can be observed, CWND grows up while the buffer is inflating, until a expiration time event happens, where the congestion window decreases its value, alike the buffer occupancy. All in all, Fig. 4 shows that the LMS link is rather challenging for transport protocols to adapt to the actual transmission capacity, which has strong impact on the application performance, particularly on the delay and jitter.

We now compare the performance obtained by QUIC and TCP over various LMS links for different configurations. Both transport protocols use CUBIC as their congestion control mechanism. Fig. 5 shows the packet delay and application throughput obtained in both bands, *Ka* and *S*, and for different buffer sizes. The results are averaged over 30

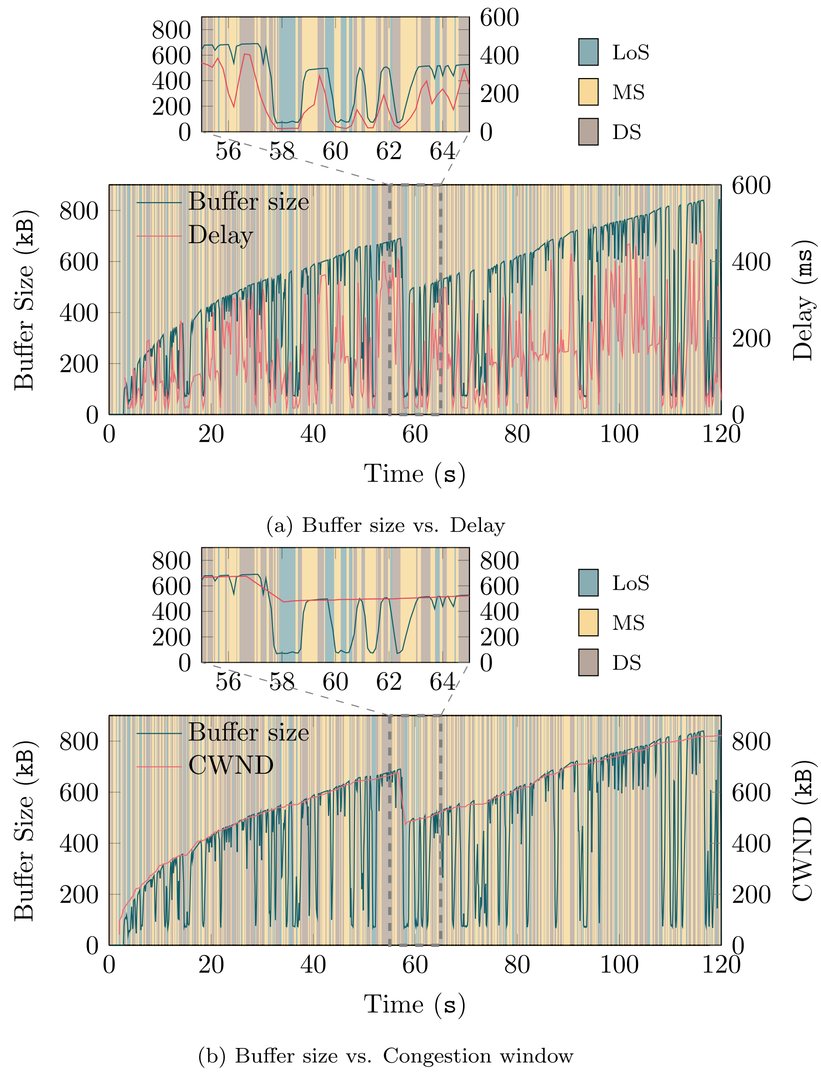


Fig. 4. Example of the evolution of the buffer size, delay and congestion window over a LMS channel with an infinite capacity buffer.

independents runs, each experiment comprising transmissions active for 60 s. In all cases, the application generates traffic at the average link capacity. When using QUIC we represent the results that were obtained for different number of active streams, using the default scheduler (RR).

As expected, we can observe that the buffer size has a strong impact over the delay. However, the results show that the performance of TCP is more severely hindered, showing a very high delay with an infinite buffer. Since both protocols use the same congestion control, this difference is caused by the congestion detection improvements that are implemented in QUIC [32]. On the other hand, the average rate is fairly stable, regardless of the protocol or buffer size. As for the multi-stream capability, the results evince that it has little impact on the protocol performance. This comparison indicates that QUIC is a more suitable transport solution for highly varying channels, such as LMS links.

6.2. QUIC schedulers performance over LMS

After comparing QUIC and TCP, we now focus on the performance of QUIC over LMS links when using different schedulers. In order to capture how they impact the protocol behavior, QUIC is configured for all cases with two streams, having different application rates: the rate of the first stream is twice the rate of the second one. Furthermore, we configure different overall sending rates, as a ratio of the average channel capacity. We restrict the evaluation to the *Ka* band, which has a higher average capacity (∼ 45 Mbps). Table 2 summarizes the traffic configurations used in this analysis.

Table 2
Application traffic configuration.

Common setup	
Average channel capacity (*Ka*)	45 Mbps
Packet size	1000 Bytes
Packet distribution	Poisson
Traffic time generation	60 s
Scenario 1 (50%)	
App. data rates	$Stream_1$: 15 Mbps; $Stream_2$: 7.5 Mbps
Scenario 2 (80%)	
App. data rates	$Stream_1$: 24 Mbps; $Stream_2$: 12 Mbps
Scenario 3 (100%)	
App. data rates	$Stream_1$: 30 Mbps; $Stream_2$: 15 Mbps

As for the schedulers, it is worth noting that WFQ is configured to provide a higher priority (×2) to the first stream, reflecting the average traffic load distribution.

First, Fig. 6 shows the evolution of the application buffer using the different streams (rows) and traffic rates (columns). As expected, we

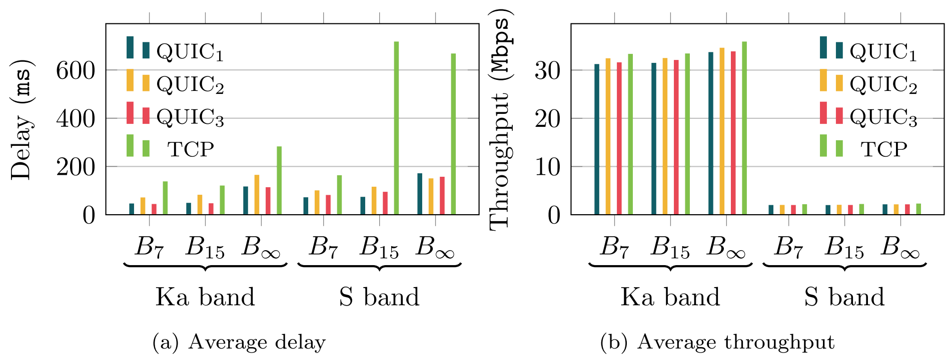


Fig. 5. Performance comparison between TCP and QUIC over a single LMS channel. Average results are obtained from 30 independent executions where traffic is generated during 1 min at a rate equal to the average channel capacity.

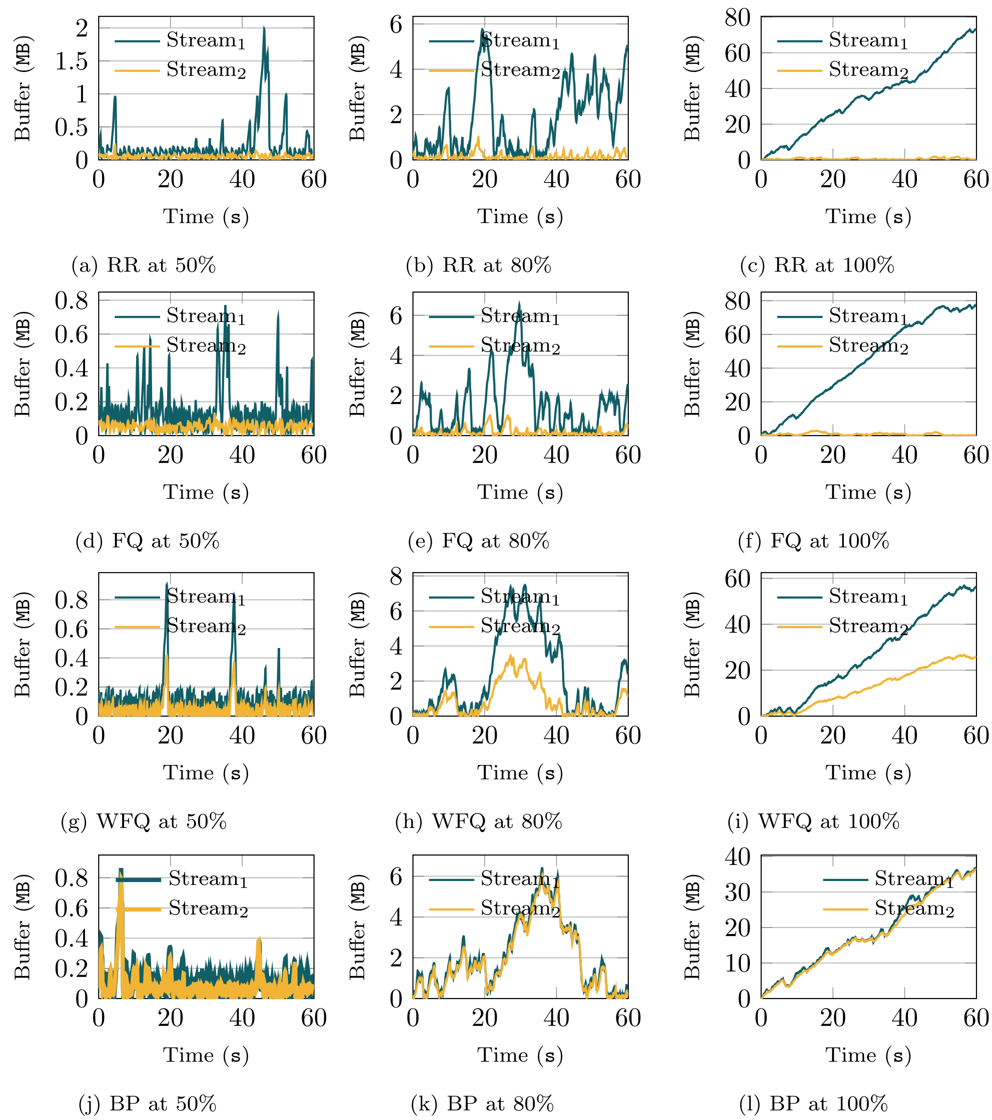
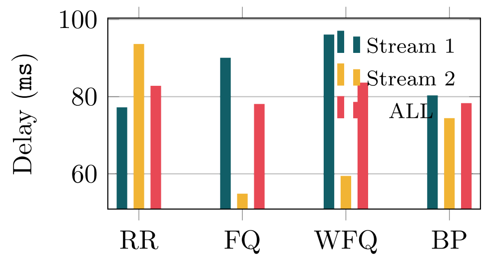
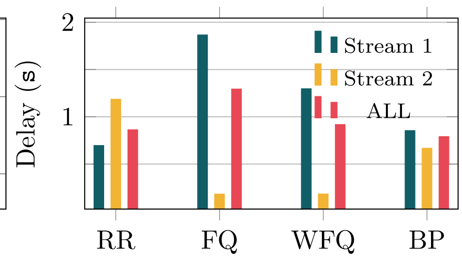


Fig. 6. Queue buffer evolution of different streams over time, using both different scheduling algorithms and binary sending rates, on a LMS link.

observe that RR and FQ (first and second rows) are not able to adapt to the traffic unbalance, leading to rather different application buffer occupancy. This behavior is more accused as the application traffic rate gets closer to the average channel capacity. On the other hand, WFQ yields a fairer distribution between the streams, leading to better balanced buffer occupancy as the traffic rates increases. Nevertheless, it does not optimally exploit situations where the traffic rate does not exactly correspond to its average value. This sub-optimal behavior is corrected over time when the channel is not saturated, but in circumstances close to saturation this effect is not observed, and WFQ is not able to yield a fair distribution of the shared resources. On the other hand, the proposed policy, based on the BP algorithm, harmonizes the

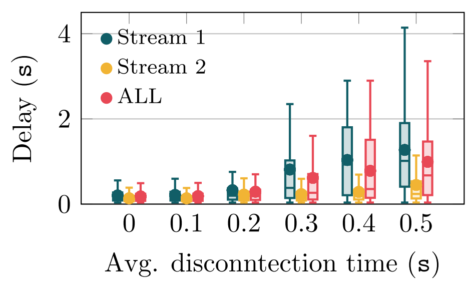


(a) Traffic rate at 50% of average capacity

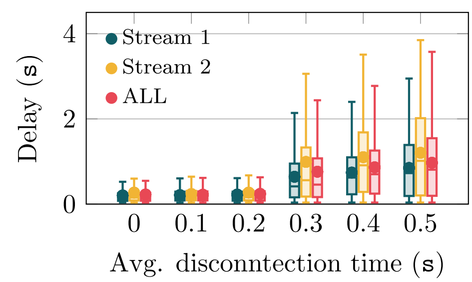


(b) Traffic rate at 80% of average capacity

Fig. 7. Average delay experienced over a Ka band LMS link, when using the different schedulers.



(a) Round Robin



(b) Back Pressure

Fig. 8. Delay distribution with RR and BP over an E2E channel with interruptions.

buffer occupancy on both streams, regardless of the channel variation, even when the traffic reaches the channel capacity, due to the fact that it checks the buffer occupancy when building the QUIC packet.

We now analyze the impact of the schedulers over the application performance. We will focus on the delay, since all the experiments showed that the difference in terms of throughput was almost negligible. Fig. 7 shows the average delay experienced by application packets, for traffic loads of 50% and 80% of the average channel capacity. The results are obtained from 30 independent experiments, each of them comprising connections lasting 60 s. To better illustrate the behavior we show the average delays experienced by each stream independently, and that affecting the overall connection. As can be observed, the static schedulers (RR, FQ and WFQ) yield rather unbalanced delay between both streams, regardless of the traffic load. Interestingly, the default RR implementation shows a fairer delay distribution, due to the better usage of the total transmission capacity. On the other hand, the proposed BP algorithm is able to harmonize the delay of both streams, almost to the same value. In addition, the proposed solution yields a slight overall delay reduction.

6.3. QUIC schedulers performance for E2E scenarios

We now broaden the performance analysis of the scheduling policies over an end-to-end setup, embracing two *Ka* LMS links, which emulate connectivity conditions between ground stations and satellites, connected by a single ISL. As mentioned in Section 3, the ISL is modeled with a two state Markov Chain, depicted in Fig. 1(b), to mimic interruption situations in the communication.

Following the same setup as before, we configure two unbalanced QUIC streams, where the rate of the first one is twice as high as the second. Since we introduce interruption situations with different average time, so decreasing the overall network capacity, the aggregated traffic rate is set to 50% of the average LMS capacity.

The ISL link is modeled with exponentially distributed dwell times: an active period (Connect) with 80 Mbps transmission rate and average sojourn time of 5 s; and an idle state that emulates link disconnections and whose average dwell time is increased from 0 to 0.5 s. For each value of the average interruption time, we run an independent simulation, which lasts 120 s, which allows a sufficient number of state transitions to obtain meaningful results. It is worth noting that the execution only stops when all bytes are received at the receiving application.

Fig. 8 depicts the distribution of the average end-to-end delay of both streams for two scheduling strategies, the proposed BP and the default RR, as we increase the average interruption time at the ISL link. As expected, we observe that the average end-to-end delay steadily increases, as well as its sparsity. It is worth noting that this variability is slightly higher when using the BP algorithm, which on the other hand leads to a more balanced behavior for the streams, since it achieves harmonizing the buffers for all configurations. This can be clearly observed for an average disconnection time of 0.5 s, where the BP solution yields similar values for the two streams, whereas with RR they significantly differ.

7. Conclusion

In this work we have first assessed the performance of transport protocols (TCP and QUIC) over LEO networks. We have developed a novel methodology that combines virtualization techniques through Docker containers and real protocol implementations, with a simulation framework in *ns-3* to enable systematic experiments, featuring realistic modeling of the underlying connectivity. The results evince that QUIC outperforms the traditional TCP when operated over channels exhibiting high variability.

We have broadened the analysis to thoroughly characterize the behavior of different scheduling strategies, which exploit the multi-streaming capability QUIC features. The default Round-Robin algorithm used by the QUIC implementation has shown not to be an optimal solution for unbalanced traffic across various streams. Besides, the well known Fair Queuing approach is able to keep the transmission bit rate, without severely hindering the average delay, but it cannot adapt to situations of unbalanced traffic.

On the other hand, the WFQ solution yields a good balance between the buffer length of the active streams. However, it is not able to adapt to varying conditions, and the corresponding weights would need to be set in advance. The proposed scheme, based on the Back-Pressure algorithm has shown to stabilize all active streams' queue lengths, just using the occupancy state of such buffers.

The proposed methodology allows increasing the complexity of the underlying connectivity, by adding disconnection times at the ISL links (as was discussed in the paper) or background traffic. This could be exploited, for instance, to model the features of different LEO networks, such as crossed-ISL links, which might be caused by continuous satellite movements in different constellations. We will also tackle the enhancement of the proposed models, by adding new features, such as random large-scale fading, on top of the dynamism captured by the proposed Markov Chain.

In our future work we plan to exploit the methodology to broaden the analysis, and to include other elements, such as congestion control mechanisms. In addition, we will explore the adoption of novel scheduling algorithms and explicitly account for the traffic delay. We will also explore the possibility of using external capacity traces to emulate link behavior within *ns-3*, obtained either from real networks or from high precision simulation frameworks, which cannot be used directly with real applications. This new feature would allow broadening the evaluation and further validate the obtained results.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

Acknowledgments

This project was funded the Spanish Government (Ministerio de Economía *y* Competitividad, Fondo Europeo de Desarrollo Regional, MINECO-FEDER) by means of the project SITED: Semantically-enabled Interoperable Trustworthy Enriched Data-spaces (PID2021-125725OB-I00); and by EU Horizon 2020 research and innovation program, Drones4Safety-agreement No. 861111.

References

[1] R. Poorzare, A. Calveras, Open trends on TCP performance over urban 5G MmWave networks, in: Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, in: PE-WASUN '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 85–92, http://dx.doi.org/10.1145/3416011.3424749.

[2] J. Iyengar, M. Thomson, QUIC: A UDP-based multiplexed and secure transport, RFC 9000, 2021, http://dx.doi.org/10.17487/RFC9000, URL https://rfc-editor.org/rfc/rfc9000.txt.

[3] I.F. Akyildiz, A. Kak, S. Nie, 6G and beyond: The future of wireless communications systems, IEEE Access 8 (2020) 133995–134030, http://dx.doi.org/10.1109/ACCESS.2020.3010896.

[4] W. Jiang, B. Han, M.A. Habibi, H.D. Schotten, The road towards 6G: A comprehensive survey, IEEE Open J. Commun. Soc. 2 (2021) 334–366, http://dx.doi.org/10.1109/OJCOMS.2021.3057679.

[5] S. Liu, Z. Gao, Y. Wu, D.W. Kwan Ng, X. Gao, K.-K. Wong, S. Chatzinotas, B. Ottersten, LEO satellite constellations for 5G and beyond: How will they reshape vertical domains? IEEE Commun. Mag. 59 (7) (2021) 30–36, http://dx.doi.org/10.1109/MCOM.001.2001081.

[6] I. Leyva-Mayorga, B. Soret, M. Röper, D. Wübben, B. Matthiesen, A. Dekorsy, P. Popovski, LEO small-satellite constellations for 5G and beyond-5G communications, IEEE Access 8 (2020) 184955–184964, http://dx.doi.org/10.1109/ACCESS.2020.3029620.

[7] M. Hosseinian, J.P. Choi, S.-H. Chang, J. Lee, Review of 5G NTN standards development and technical challenges for satellite integration with the 5G network, IEEE Aerosp. Electron. Syst. Mag. 36 (8) (2021) 22–31, http://dx.doi.org/10.1109/MAES.2021.3072690.

[8] T. Darwish, G.K. Kurt, H. Yanikomeroglu, M. Bellemare, G. Lamontagne, LEO satellites in 5G and beyond networks: A review from a standardization perspective, IEEE Access 10 (2022) 35040–35060, http://dx.doi.org/10.1109/ACCESS.2022.3162243.

[9] O. Kodheli, S. Andrenacci, N. Maturo, S. Chatzinotas, F. Zimmer, An uplink UE group-based scheduling technique for 5G mMTC systems over LEO satellite, IEEE Access 7 (2019) 67413–67427, http://dx.doi.org/10.1109/ACCESS.2019.2918581.

[10] R. Xie, Q. Tang, Q. Wang, X. Liu, F.R. Yu, T. Huang, Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues, IEEE Netw. 34 (3) (2020) 224–231, http://dx.doi.org/10.1109/MNET.011.1900369.

[11] K. Wei, Q. Tang, J. Guo, M. Zeng, Z. Fei, Q. Cui, Resource scheduling and offloading strategy based on LEO satellite edge computing, in: 2021 IEEE 94th Vehicular Technology Conference, VTC2021-Fall, 2021, pp. 1–6, http://dx.doi.org/10.1109/VTC2021-Fall52928.2021.9625072.

[12] Y. Zhang, C. Chen, L. Liu, D. Lan, H. Jiang, S. Wan, Aerial edge computing on orbit: A task offloading and allocation scheme, IEEE Trans. Netw. Sci. Eng. 10 (1) (2023) 275–285, http://dx.doi.org/10.1109/TNSE.2022.3207214.

[13] E. Zhou, M. Kobayashi, T. Fujihashi, M.A. Alim, S. Saruwatari, M. Nishi, T. Watanabe, Reliable multicast streaming for hierarchical non-terrestrial network, in: 2022 9th International Conference on Wireless Networks and Mobile Communications, WINCOM, 2022, pp. 1–6, http://dx.doi.org/10.1109/WINCOM55661.2022.9966478.

[14] Z. Tang, H. Zhou, T. Ma, K. Yu, X.S. Shen, Leveraging LEO assisted cloud-edge collaboration for energy efficient computation offloading, in: 2021 IEEE Global Communications Conference, GLOBECOM, 2021, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM46510.2021.9685309.

[15] T. Shreedhar, R. Panda, S. Podanev, V. Bajpai, Evaluating QUIC performance over web, cloud storage, and video workloads, IEEE Trans. Netw. Serv. Manag. 19 (2) (2022) 1366–1381, http://dx.doi.org/10.1109/TNSM.2021.3134562.

[16] P. Qian, N. Wang, R. Tafazolli, Achieving robust mobile web content delivery performance based on multiple coordinated QUIC connections, IEEE Access 6 (2018) 11313–11328, http://dx.doi.org/10.1109/ACCESS.2018.2804222.

[17] J. Dizdarević, A. Jukan, Experimental benchmarking of HTTP/QUIC protocol in IoT cloud/edge continuum, in: ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1–6, http://dx.doi.org/10.1109/ICC42927.2021.9500675.

[18] H. Tsunoda, N. Kato, A. Jamalipour, Y. Nemoto, Performance evaluation of SCTP wth adaptive multistreaming over LEO satellite networks, in: 2007 International Workshop on Satellite and Space Communications, 2007, pp. 150–154, http://dx.doi.org/10.1109/IWSSC.2007.4409407.

[19] Z. Zhang, Q. Guo, Z. Gao, A prediction based SCTP handover scheme for IP/LEO satellite network, in: 2010 6th International Conference on Wireless Communications Networking and Mobile Computing, WiCOM, 2010, pp. 1–4, http://dx.doi.org/10.1109/WICOM.2010.5601455.

[20] W. Yang, S. Shu, L. Cai, J. Pan, MM-QUIC: Mobility-aware multipath QUIC for satellite networks, in: 2021 17th International Conference on Mobility, Sensing and Networking, MSN, 2021, pp. 608–615, http://dx.doi.org/10.1109/MSN53354.2021.00093.

[21] A. Martin, N. Khademi, On the suitability of BBR congestion control for QUIC over GEO satcom networks, in: Proceedings of the Workshop on Applied Networking Research, ANRW '22, Association for Computing Machinery, New York, NY, USA, 2022, http://dx.doi.org/10.1145/3547115.3547194.

[22] S. Yang, H. Li, Q. Wu, Performance analysis of QUIC protocol in integrated satellites and terrestrial networks, in: 2018 14th International Wireless Communications & Mobile Computing Conference, IWCMC, 2018, pp. 1425–1430, http://dx.doi.org/10.1109/IWCMC.2018.8450388.

[23] C. Cui, Y. Lu, S. Li, J. Li, Z. Ruan, DASH+: Download multiple video segments with stream multiplexing of QUIC, in: 2022 Tenth International Conference on Advanced Cloud and Big Data, CBD, 2022, pp. 66–72, http://dx.doi.org/10.1109/CBD58033.2022.00021.

[24] X. Shi, L. Wang, F. Zhang, B. Zhou, Z. Liu, Pstream: Priority-based stream scheduling for heterogeneous paths in multipath-QUIC, in: 2020 29th International Conference on Computer Communications and Networks, ICCCN, 2020, pp. 1–8, http://dx.doi.org/10.1109/ICCCN49398.2020.9209682.

[25] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, R. Steinmetz, Multipath QUIC: A deployable multipath transport protocol, in: 2018 IEEE International Conference on Communications, ICC, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICC.2018.8422951.

[26] F.P. Fontan, M. Vazquez-Castro, C.E. Cabado, J.P. Garcia, E. Kubista, Statistical modeling of the LMS channel, IEEE Trans. Veh. Technol. 50 (6) (2001) 1549–1567, http://dx.doi.org/10.1109/25.966585.

[27] A. Chen, C. Chang, 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, http://dx.doi.org/10.1109/VTC.2001.957168.

[28] R. Hermenier, C. Kissling, 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, http://dx.doi.org/10.1109/IWSSC.2009.5286440.

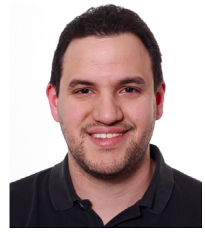
[29] Y. Zhu, M. Sheng, J. Li, R. Liu, Performance analysis of intermittent satellite links with time-limited queuing model, IEEE Commun. Lett. 22 (11) (2018) 2282–2285, http://dx.doi.org/10.1109/LCOMM.2018.2866570.
[30] N.J.H. Marcano, L. Diez, R.A. Calvo, R.H. Jacobsen, On the queuing delay of time-varying channels in low Earth orbit satellite constellations, IEEE Access 9 (2021) 87378–87390, http://dx.doi.org/10.1109/ACCESS.2021.3089005.
[31] M.J. Neely, Stochastic Network Optimization with Application to Communication and Queueing Systems, Morgan and Claypool Publishers, 2010.
[32] J. Iyengar, I. Swett, QUIC loss detection and congestion control, RFC 9002, 2021, http://dx.doi.org/10.17487/RFC9002, URL https://www.rfc-editor.org/info/rfc9002.

Fátima Khan received his B.Sc. from University of Cantabria in 2021 and she is currently finishing her M.Sc. Since 2022 she is a researcher with the Communications Engineering Department at the University of Cantabria. Her research focuses on transport layer optimization in non-terrestrial-networks (NTN) for 5G and B5G architectures.

Cristina Hervella received the B.Sc. and M.Sc from the University of Cantabria, Spain, in 2020 and 2022 respectively. In 2019, Cristina did an internship in IT department of Solvay Chemicals Inc., where she collaborated in the development of industrial applications and services. She is currently a working as IT Consultant at Evenbytes S.L. Her research interests lie in the areas of cloud service development and provision.

Luis Diez received his M.Sc. and Ph.D. from University of Cantabria in 2013 and 2018 respectively. He is currently Assistant Professor at the Communications Engineering Department in that University. He has been involved in different international and industrial research projects. His research focuses on future network architectures, resource management in wireless heterogeneous networks, and IoT solutions and services. He has published more than 40 scientific and technical papers in those areas, and he has served as TPC member and reviewer in a number of international conferences and journals. As for teaching, Dr. Diez has supervised 15 BSc and MSc Thesis, and he teaches in courses related to cellular networks, network dimensioning and service management.

Fátima Fernández received the degree in telecommunication technologies engineering and the M.Sc. degree in telecommunication engineering from the University of Cantabria, Santander, Spain, in 2017 and 2019, respectively. She is currently pursuing the joint Ph.D. degree with Ikerlan Technology Research Centre in collaboration with the University of Cantabria. In 2019, she was a Fellow Researcher with the Communications Engineering Department, University of Cantabria. Her area of work is mainly centered around congestion control algorithms for transport layer protocols.



Néstor J. Hernández Marcano is a 5G Research and Development Engineer at Gatehouse Satcom since 2022, specializing in the 3GPP 5G NB-IoT solutions at the system and protocol stack levels. From 2017 to 2021, he was a postdoctoral researcher on small satellite and UAV communications at Aarhus University. He obtained his MSc. in Electronics Engineering, Digital Communications from Universidad Simón Bolívar, Venezuela in 2013, and the PhD in Wireless Communications from Aalborg University in 2016. His research interests focus on wireless communications systems and networks, error/erasure correcting codes, space-air-ground integrated networks, machine learning and network security. He has published more than 30 papers in these areas in international conferences and journals, where he is also a regular reviewer and TPC member, and has 15+ years of experience in academic and industrial projects.

Rune Hylsberg Jacobsen is a Professor at the Department of Electrical and Computer Engineering, Aarhus University. He has a master's degree in science in physics and chemistry and has a PhD degree (1997) in laser physics and optoelectronics from Aarhus University, Denmark. His research interests include computer networking, wireless communications, network security, cooperative intelligent systems, and satellite communications. His professional career spans more than 14 years in the telecommunications and IT industry, where he has been responsible for the management of research and development. He has published more than 85 peer-reviewed journal and conference papers in areas of communications and its applications.

Ramón Agüero Calvo received his MSc in Telecommunications Engineering (1st class honors) from the University of Cantabria in 2001 and the PhD (Hons) in 2008. He is currently a Professor at the Communications Engineering Department in that university. His research focuses on future network architectures, especially regarding the (wireless) access part of the network and its management. He has published more than 230 scientific papers in such areas and he is a regular TPC member and reviewer on various related conferences and journals. Ramon Aguero serves in the Editorial Board of IEEE Communication Letters (Area Editor since 2023), IEEE Open Access Journal of the Communications Society, IEEE Systems Journal, Wireless Networks (Springer). Dr. Aguero has supervised 6 PhDs and more than 80 BSc and MSc thesis. He is the main instructor in courses dealing with Networks, and Traffic Modeling, both at BSc and MSc levels. Since 2016, he is the Head of the IT Area (deputy CIO) at the University of Cantabria.