Use of QUIC Protocol for Efficient Data Transmission over Satellite in Emergency Scenario

Armir Bujari
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
armir.bujari@unibo.it

Mirko Franco
Department of Mathematics
University of Padua
Padua, Italy
mifranco@math.unipd.it

Claudio E. Palazzi
Department of Mathematics
University of Padua
Padua, Italy
cpalazzi@math.unipd.it

Mattia Quadrini
Research and Development
RomARS S.r.l.
Rome, Italy
quadrini@romars.tech

Cesare Roseti
University of Rome "Tor Vergata",
Electronics Engineering Dpt.
Rome, Italy
roseti@ing.uniroma2.it

Francesco Zampognaro*
University of Rome "Tor Vergata",
Electronics Engineering Dpt.
Rome, Italy
zampognaro@ing.uniroma2.it

***Abstract*—During emergency situations, reliable and fast communication is crucial for an effective response and coordination of recovery and rescue operations. However, regular communication means may not be always available nor reliable. In this scenario, constellations of Low Earth Orbit satellites can offer high bandwidth capabilities with (relatively) low latency, making them ideal for supporting disaster management teams on the ground and complementing (or replacing) terrestrial telecommunication infrastructures. In this context, the use of the recently defined QUIC protocol can provide the possibility to seamlessly switch from ground communication to satellite ones when needed. In this paper, we discuss the case study of satellite communication in an emergency scenario, assessing via a simulation study the benefits achievable by QUIC under varying queuing parameters and congestion control algorithm behavior.**

***Index Terms*—QUIC, LEO constellations, multi-access, emergency communications, bottleneck queue.**

I. Introduction

In disaster-struck areas or in general in emergency scenarios, the availability and effectiveness of telecommunication infrastructures cannot usually be taken for granted. Nonetheless, in these circumstances, communications are paramount in supporting personnel (both military and civil) coordination, data exchange, alarm notifications, logistics, and in general to improve operators' situational awareness and to optimize the recovery operations [1], not only for human beings but also in general for valuable assets [2].

Emergency communications are today mostly leveraging digital applications, for voice, multimedia and data transfer, via packet-switched IP networks that are typically wireless (e.g., LTE or 5G) [3], to ensure mobility and high performance using handheld/smart devices. The amount of data to exchange is growing more and more, if considering state-of-the-art applications including augmented reality, high-quality aerial photography (by drones), as well as collection and processing of big data. Furthermore, in-network services and Multi-Access Edge Computing (MEC) are often employed to offer added-value local services without the need of communication with the core side of the network [4], [5]. In this context, satellite communications (SatCom) can play a key role. In fact, once within the SatCom coverage area, digital communication services can be enabled directly with terminals on the ground, regardless of the status of ground infrastructures in the target area. Therefore, it would be possible to use terrestrial mobile networks, if and when available, and handover to SatCom (and vice versa) so to guarantee service reliability and resilience to possible terrestrial network outages. Then, a local node suitable to manage the terrestrial and satellite access, as well as the local breakdown of traffic and running of local/virtual services, becomes essential.

In these conditions, we will consider data transfer for emergency applications by the use of the QUIC protocol [6], a recently defined reliable transport protocol built on top of UDP. QUIC includes and enhances several of the key operating principles of TCP, making it a more robust and a feature-rich alternative to TCP itself. Indeed, QUIC is at the core of the very important and widely adopted HTTP/3 specification [7]. Interestingly, QUIC comes with a path-migration embedded feature, which means that the end-to-end data session can be maintained also when the access networks (and IP addresses in use) change, contrarily from TCP.

In the second part of the paper, we will present a simulation campaign based on Network Simulator 3[1] (NS3), to show QUIC peculiar characteristics and robustness to link change, and identify the most suitable network configuration at the access nodes to ensure an efficient data transfer, as close as possible to the available link capacity.

*Authors in alphabetical order. Corresponding author: F. Zampognaro

[1] https://www.nsnam.org/

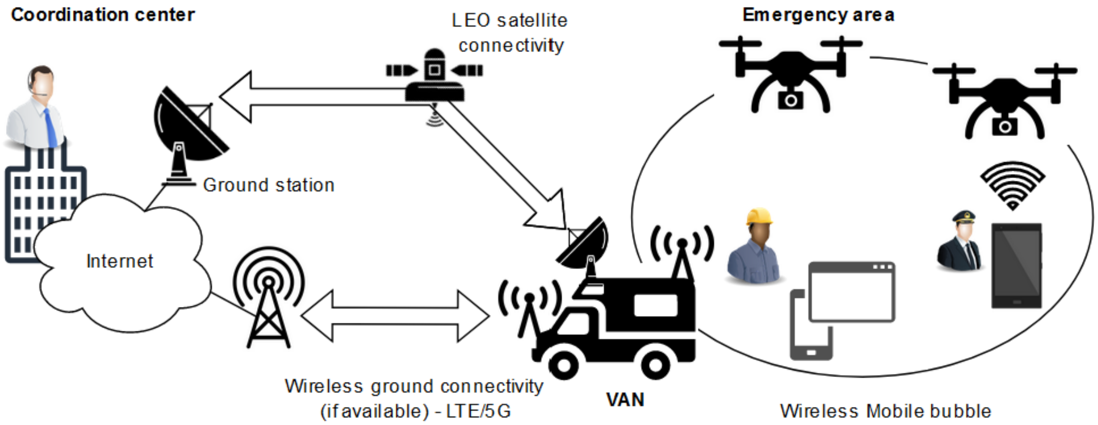


Fig. 1: High level system architecture for data transfer in emergency scenarios.

II. Reference Architecture

Recently, constellations of LEO satellites with a very high number of satellites (hundreds up to thousands), called *mega-constellations*, are commercially available and at a competitive price. The major one can be considered Starlink [8], used in the recent months also as communication support in the Ukrainian conflict. Nonetheless, to provide broadband connectivity in case of LEO megaconstellations, the terminal antenna in use is not small enough to be integrated in a smartphone-sized device. Furthermore, as anticipated, it is very important to consider a MEC node on the target area of interest, suitable to perform local operations on traffic (i.e., mining, data fusion) and offer enhanced network services (i.e., Mission Critical Push to Talk, centralized data storage, etc.).

To address these principal service requirements, we propose an architecture for emergency communications, shown in Figure 1, in which the SatCom antenna is placed in a larger mobile unit, such as a VAN. This backhaul connectivity will ensure a reliable and quick-to-deploy communication with a coordination center away from the emergency area. Furthermore, thanks to the presence of a dedicated access node, we can consider the use (in combination, or as an alternative) of mobile terrestrial networks, LTE or 5G, which may be available and can offer better performance than the SatCom one. The VAN can then enable a multi-access configuration, selecting the best backhaul link to use according to circumstances, and handing over existing connections from one link (we can consider the terrestrial one, if available, the preferred choice) to the other link (i.e., the LEO satellite) if necessary. Then, the VAN can offer connectivity to end-users with common Wi-Fi or licensed standalone local LTE or 5G bubbles, indicated in figure as "Wireless Mobile bubble". Finally, the VAN can also host all local services and equipment for MEC.

In this context of multi-access, we investigate an initial use-case in which a large data file must be sent by an operator device in the emergency area, relayed towards the coordination center in a remote office. It is worth noting that the backhaul connectivity options are managed by different Network Operators, a Mobile Network Operator (MNO) and a Satellite Network Operator (SNO), with different network addressing, NAT configurations and connection characteristics. The data transfer is performed using the QUIC protocol, in the uplink direction, and may suffer from link change during its establishment. Performance analysis of MEC and local added-value services are not considered and will be the subject of future work.

III. QUIC protocol background

QUIC is defined as an Internet standard general-purpose, secure and connection-oriented transport protocol based on UDP, inheriting advantages of the TCP such as a reliable data delivery and congestion control [6]. Moreover, the connection establishment combines the mandatory negotiation of cryptographic (TLSv1.3) and transport parameters, allowing it in some cases to exchange data as soon as possible even during initial phases (0-RTT handshake feature). QUIC connections are identified by a Connection IDentifier (CID), a 64-bit unsigned number randomly generated by the server. In turn, a connection consists of several Streams, an ordered sequence of bytes, which can be either unidirectional (to support live media streams) or bidirectional (suitable for HTTP requests and responses). Finally, a QUIC packet may transport one or multiple frames, belonging to different streams and multiplexed according to priority information provided by the endpoints. The frame type field in the header provides some information about the carried data (user data rather than signalling information such as Acknowledgments – AKCs).

QUIC became (and was conceived in the first place with this goal) the reference transport protocol for HTTP/3 [7], improving flexibility and performance with regard to HTTP/2 over TCP.

A. QUIC Loss Detection and Congestion Control

The QUIC working group in IETF proposed to leverage the extensive know-how of TCP congestion control porting its key working principles to QUIC, as defined in [9]. In this document, the baseline/mandatory sender-side Congestion Control

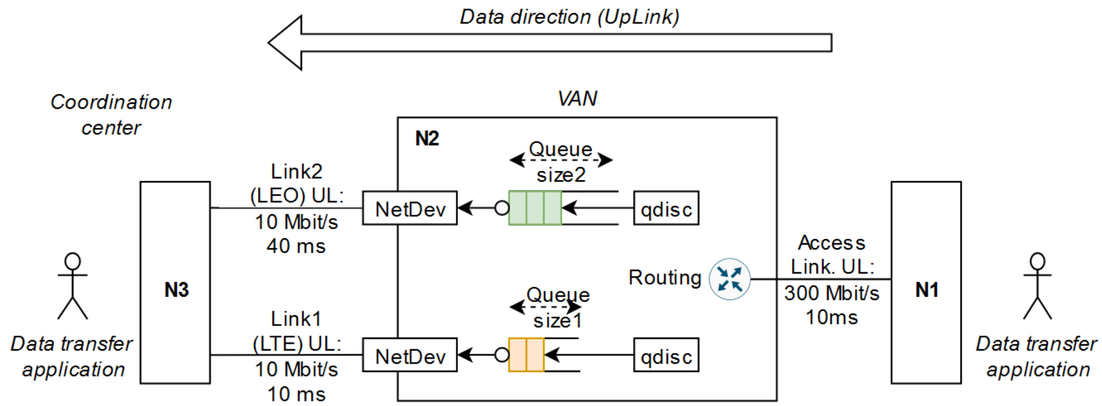


Fig. 2: NS3 simulation setup with 3 nodes and 3 links for modelling the proposed system architecture.

Algorithm (CCA) is defined to be in practice similar to TCP NewReno [10] CCA, with some minor changes. Nonetheless, QUIC aims at a pluggable congestion control mechanism, allowing users to select different CCAs. In this way, other variants of TCP CCA, which offer dedicated optimizations in specific environments, can be adopted. Examples are TCP Westwood [11], suitable to wireless links, TCP Hybla [12] and Vegas [13], showing some benefits in satellite scenarios, up to the recent TCP Cubic [14] and BBR [15], whose goal is to be effective in a large set of configurations. Possible CCA variations in QUIC are left open to implementers so, as consequence, different QUIC protocol stacks offer different CCAs out of the box.

Although inheriting TCP congestion control logic, QUIC improves on that by introducing a different communication environment. It applies some add-on loss-recovery mechanisms such as the Forward-RTO (to efficiently handle spurious timeouts) and Early Retransmit (to accelerate retransmission in case of small windows using a lower number of duplicated ACKs as loss indication), avoiding Fast-Retransmit & Fast-Recovery [16]. Furthermore, QUIC offers a more detailed feedback information for loss detection: on one hand, it uses a monotonically increasing packet number, also for retransmitted packets, to avoid the ambiguity problem (a retransmission generates a new QUIC packet, including lost frames embedded in the original packet); then, QUIC ACKs carry information about the interval time between the reception of a packet and the generation of the corresponding ACK (allowing the sender to better compute the path RTT). Finally, QUIC adopts the selective acknowledgement mechanism, while a QUIC ACK frame can embed multiple ACK blocks. Even if the CCA has been partially redesigned in QUIC, it is still subject to macroscopic effects and possible inefficiencies of TCP NewReno CCA, specifically in relation with end-to-end queues and bottlenecks characteristics [17].

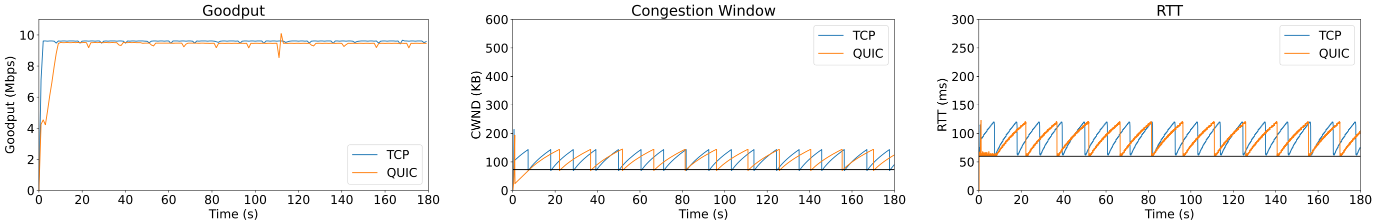
B. Connection migration and resilience to NAT rebinding

One of the advantages in using UDP is the Connection (or Path) Migration feature: the CID reference lives with the connection and not an underlying network sockets. This choice allows end-to-end QUIC connections (including all the streams that are part of it) to tolerate path changes. Only clients can trigger a path migration in the current version of QUIC, and security checks on the new path (refer to Path Validation) should be completed before. QUIC's consistent connection ID allows in particular connections to survive on changes to the client's IP and port, such as those caused by NAT rebindings or by the client changing network connectivity to a new address [18]. QUIC provides automatic cryptographic verification of a rebound client, since the client continues to use the same session key for encrypting and decrypting packets. In conclusion, the CID reference can be used to allow migration of the connection to a new server IP address as well, since it remains consistent across changes in the client's and the server's network addresses.

IV. Simulation Scenario

In this work, the goal is to assess key features of QUIC protocol, as discussed in previous sections, in characterizing and optimizing data transfers within a realistic multi-segment communication network for emergency scenarios.

The main assumption affecting the data transfer is the presence of a central routing node (i.e., the VAN in Figure 1), able to support local wireless communications and bachkaul with either terrestrial or satellite wireless connections. In this setup, we will focus on data uploads to offer an initial evaluation, using QUIC with default parameters and adjusting backhaul links queues (size and type). Even if we can compare QUIC data transfer with TCP data transfer, as soon as the backhaul network changes, TCP connections are reset and, depending on the application, the data transfer shall be resumed or restarted. QUIC path migration instead will make the transfer seamless also in case of network-change events. Please note that in previous work [18], [19], this feature is exploited by performing a proxy operation and delivering TCP traffic within QUIC tunnels, whereas in the present work the QUIC transfer is performed end-to-end. Furthermore, this work is using the open-source network simulator, NS3, a widely-used choice for computer network research, making it possible to run several configurations for fast experimentation, prototyping and design



(a) (b) (c)

Fig. 3: TCP and QUIC data uplink (a) Goodput (b) CWND (c) RTT.

of new CCAs. NS3 offers very accurate models of network device, protocol, and environment/topology to simulate various scenarios and evaluate network architectures and algorithms in a controlled framework. One of the key features of NS3 is its capability to test applications' performance in various topologies, mixing terrestrial and satellite links. The QUIC module for NS3 has been developed out of the NS3 official project to be integrated with some of the recent versions of the simulator [20], [21]. The module offers congestion control algorithm selection, accurate packet loss recovery mechanisms and flow control features based on the TCP ones already available. QUIC is implemented in NS3 as a separate module, that can be integrated with existing applications and the remainder of the simulated TCP/IP stack.

A. Settings

The simulation configuration, including values used to model the data links, is reported in Figure 2. The NS3 simulation scenario was defined assuming the simplest possible topology, suitable for modelling the above mentioned reference architecture. Therefore, we defined 3 nodes and 3 links: N1 is the traffic source, enabling either reliable data transfer over TCP or QUIC. N1 can transfer data assuming a relatively fast wireless bubble, with Uplink (UL) bandwidth of 300 Mbit/s and a latency of 10 ms, in line with e.g., WLAN coverage with current standards such as Wi-Fi 6. Next, N2 is the main router hosted by the VAN node, in which the decision of which link to use is taken according to their availability. The two links are modelled both at 10 Mbit/s Uplink capacity, but with different latency and respectively 10 ms for the terrestrial backhaul and 40 ms for the LEO satellite one. Eventually, N3 represents the traffic sink, enabling an application to make use of the same transport used by N1 application.

Concerning the queues shown in Figure 2, they are associated with the two backhaul links, representing the bottleneck links of the end-to-end path. Such queues are necessary to handle buffering at the bottleneck (according to a queuing discipline, qdisc in figure, assumed by default as First In - First Out – FIFO) before the packets are delivered to the hardware. Indeed, it is known that NewReno CCA is behaving optimally if the queue sizes are approximately equal to the Bandwidth Delay Product (BDP) [16], therefore these queues size may have different values.

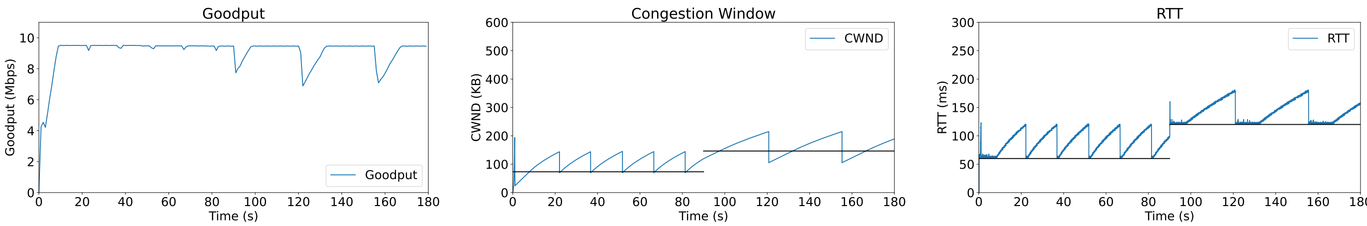
It is worth to remark that smarter queuing policies other than FIFO are available, such as Controlled Delay (CoDel), PIE or Cake, [17], [22]. The goal of these adaptive queues is to reduce standing occupation of the buffer by aggressive CCAs, mitigating the so-called "bufferbloat" effect. Therefore, as further assessment, in addition of identifying the impact of FIFO queue sizes to the data transfer, we will also evaluate the use of CoDel as qdisc. Indeed CoDel [17] is set as the default qdisc in most of the recent Linux distributions.

During the simulation, an handover event from terrestrial to satellite is triggered during the data transfer. This represents the worst case scenario in which the terrestrial connectivity becomes unavailable during an emergency communication. As results of the simulations, we will present the Congestion Window (cwnd) and RTT of either TCP or QUIC connections, as well as the resulting throughput.

V. Results

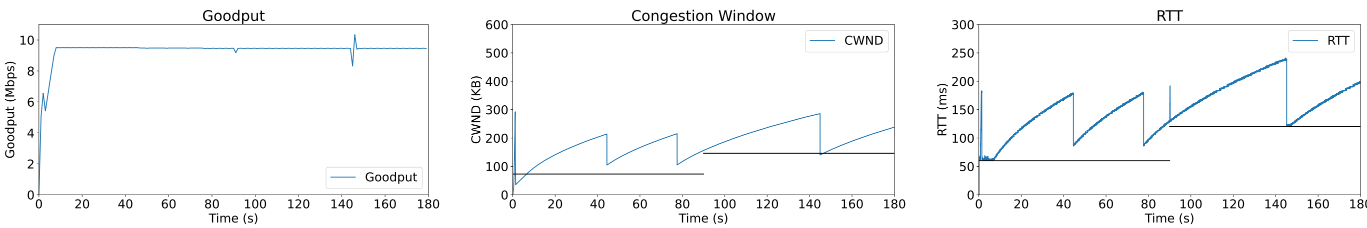
The first test result concerns a comparison of the same NewReno CCA when used by TCP or by QUIC. We present the results in Figure 3, assuming the same network conditions and queue configurations, with the availability, during the whole data upload, of the terrestrial LTE link i.e., without handovers which would interrupt forcibly the data transfer in case of TCP. As expected, the goodput is almost equivalent between TCP and QUIC, as well as the dynamics of CWND and RTT, except some minor differences in the startup and a slightly higher RTT in average. In both cases the "Queue size 1" value is set to the BDP of that link. In the CWND and RTT plots, the black horizontal line represents the reference BDP value and the PHY delay (minimum), respectively.

In the rest of simulation results we will show, as justified before, only QUIC protocol, since we are triggering an handover event to assess the seamingless data transfer also in case of network reconfigurations. The rationale of these tests is to evaluate the protocol reaction to variable network conditions, focusing on the queues setting for optimal performance. We start by showing a connection established for data upload using QUIC, with an handover after 90 s. In this case, we set both queues to the BDP value of the first link. As we can see from results in Figure 4, the data transfer continues but, since the BDP value is increasing but the queue size is constant, after the handover we have some inefficiency. From the cwnd plot, we can see that after the handover the cwnd values falls



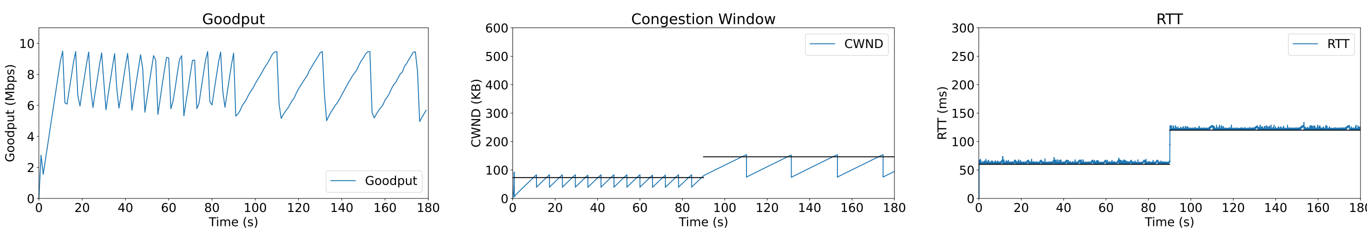
(a) (b) (c)

Fig. 4: QUIC data uplink (a) Goodput (b) CWND (c) RTT with a queue size equal to the BDP of the first link. Handover event at 90 s from the first link towards the LEO link



(a) (b) (c)

Fig. 5: QUIC data uplink (a) Goodput (b) CWND (c) RTT with a queue size equal to the BDP of the second link. Handover event at 90 s from the first link towards the LEO link



(a) (b) (c)

Fig. 6: QUIC data uplink (a) Goodput (b) CWND (c) RTT with an adaptive CoDel queue. Handover event at 90 s from the first link towards the LEO link

periodically below the optimal reference line, with an impact on goodput. In a second case, we set both queues to the BDP value of the second satellite link. Results in Figure 5 show that this choice is allowing the connection to always achieve the maximum goodput, at a cost of an higher cwnd (and RTT) when the transfer happens on the first link. In conclusion, if the queue sizes of the bottleneck links cannot be set individually, it is recommended to use the higher BDP value to ensure good performance, or to use the lower BDP value in case it is necessary to ensure the lowest RTT and degree of interactivity in the network.

Since the configuration of the queues is a critical aspect affecting overall connection performance, either in terms of goodput or RTT, we run a final test using an adaptive CoDel (Controlled Delay) queue, with the same configuration and summarized as follows:

- Maximum queue size: 10k packets
- *target*: 40 ms
- *interval*: 200 ms

We selected these values as an overall goal in controlling delay (at cost of goodput) with a common configuration which is suitable for both links. Indeed, by an analysis of the results in Figure 6, it is clear how CoDel is forcing losses as soon as the stay of a packet in the queue exceeds 40 ms, irrespective of the BDP value of the link in use. This solution is suitable to control the delay and make the network interactive and responsive (owing to the low RTT), sacrificing a little bit the goodput, and it represents a single queue configuration for both links.

The results presented in this section are just an initial assessment of QUIC protocol and bottleneck queues mutual interaction. As future work, we intend to further characterize and define optimal adaptive queue configurations, as well as enhancing the results by considering more CCAs, and specifically BBR [15], Cubic [14] and Wave [23], [24]. We intend to extend the analysis to more variable and randomic handover events, as well as considering different applications for emergency communications rather than just long data uplinks. Furthermore, the design of a satellite-tailored, robust to link change and with low pushing on the queues is being defined, in the research project QUICoS mentioned into the Acknowledgment section. Outcomes of this activity will introduce a more efficient QUIC CCA variant, suitable in particular to critical network conditions and satellite links. It will also be possible to extend the evaluation considering the joint use of the available links, extending existing solutions based on TCP or proxy-based approaches (e.g., [25], [26]).

VI. Conclusion

In this work we introduce the possible use of the QUIC transport protocol for emergency communications, assuming that the backhaul link used to transfer users' applications data in the remote area changes during data transfer. After an initial comparison with TCP Congestion Control Algorithm, the work deals with the impact of the bottleneck queue size selection for data uplinks, before showing the transfer performance when CoDel is used. QUIC demonstrated an high reliability and efficiency in case of network changes and, when paired with adaptive queues, can offer low latency communications at adequate bitrates. Future work is already identified in extending the analysis to MEC solutions, for both uplink and downlink, considering more CCAs and queue configurations. Eventually, a customized CCA for QUIC applicable to the reference scenario will be considered and assessed. Furthermore, beyond the efficacy, we also plan to evaluate the fairness and friendliness of the proposed solution in case of heterogeneous competing flows, [27], [28]

Acknowledgment

The scenarios and results presented in this work are part of the outcomes of the European Space Agency's (ESA) project "QUIC over Satellite" (QUICoS) https://artes.esa.int/projects/quicos, contract n. 4000138640/22/NL/AF. Responsibility of the contents resides with the authors.

References

[1] P. K. Esubonteng and R. Rojas-Cessa, "RESTORE: Low-Energy Drone-Assisted NLoS-FSO Emergency Communications," *IEEE Access*, vol. 10, pp. 115 282–115 294, 2022.

[2] S. D'Auria, M. Luglio, C. Roseti, R. Strollo, and F. Zampognaro, "Real time transmission of cultural heritage 3D survey in case of emergency," in *Proceedings of the 2016 3rd International Conference on Information and Communication Technologies for Disaster Management, ICT-DM*, 2016.

[3] F. Pervez, J. Qadir, M. Khalil, T. Yaqoob, U. Ashraf, and S. Younis, "Wireless technologies for emergency response: A comprehensive review and some guidelines," *IEEE Access*, vol. 6, pp. 71 814–71 838, 2018.

[4] M. C. Filippou, D. Sabella, M. Emara, S. Prabhakaran, Y. Shi, B. Bian, and A. Rao, "Multi-Access Edge Computing: A Comparative Analysis of 5G System Deployments and Service Consumption Locality Variants," *IEEE Communications Standards Magazine*, vol. 4, no. 2, pp. 32–39, 2020.

[5] M. Luglio, M. Marchese, F. Patrone, C. Roseti, and F. Zampognaro, "Performance Evaluation of a Satellite Communication-Based MEC Architecture for IoT Applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, p. 3775 – 3785, 2022.

[6] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9000

[7] M. Bishop, "HTTP/3," RFC 9114, Jun. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9114

[8] M. Albulet, "Spacex non-geostationary satellite system: Attachment a technical information to supplement schedules," *US Fed. Commun. Comm.*, 2016.

[9] J. Iyengar and I. Swett, "QUIC Loss Detection and Congestion Control," RFC 9002, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9002

[10] A. Gurtov, T. Henderson, S. Floyd, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582, Apr. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6582

[11] A. L. Grieco and S. Mascolo, "Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control," *SIGCOMM Computer Communications Review*, vol. 34, no. 2, pp. 25–38, Apr. 2004.

[12] C. Caini and R. Firrincieli, "TCP Hybla: a TCP Enhancement for Heterogeneous Networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 5, p. 547–566, Sep. 2004.

[13] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *SIGCOMM*, 1994.

[14] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *SIGOPS Operating System Review*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[15] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-Based Congestion Control," *Queue*, vol. 14, no. 5, pp. 50:20–50:53, Oct. 2016.

[16] W. R. Stevens, *TCP/IP Illustrated (Vol. 1): The Protocols*. USA: Addison-Wesley Longman Publishing Co., Inc., 1993.

[17] F. Zampognaro, "Enabling CoDel AQM with TCP Cubic connections over satellite links," in *2019 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, 2019, pp. 1–6.

[18] M. Luglio, M. Quadrini, S. P. Romano, C. Roseti, and F. Zampognaro, "Enabling an efficient satellite-terrestrial hybrid transport service through a QUIC-based proxy function," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020, pp. 1–6.

[19] A. Abdelsalam, M. Luglio, M. Quadrini, C. Roseti, and F. Zampognaro, "QUIC-proxy based architecture for satellite communication to enhance a 5G scenario," in *2019 International Symposium on Networks, Computers and Communications, ISNCC 2019*, 2019.

[20] A. De Biasio, F. Chiariotti, M. Polese, A. Zanella, and M. Zorzi, "A QUIC Implementation for NS-3," in *Proceedings of the 2019 Workshop on Ns-3*, ser. WNS3 '19, 2019, p. 1–8.

[21] U. Paro, F. Chiariotti, A. A. Deshpande, M. Polese, A. Zanella, and M. Zorzi, "Extending the NS-3 QUIC Module," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '20, 2020, p. 19–26.

[22] G. Carlucci, L. De Cicco, and S. Mascolo, "Controlling Queuing Delays for Real-Time Communication: The Interplay of E2E and AQM Algorithms," *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 3, jul 2018.

[23] A. Abdelsalam, M. Luglio, N. Patriciello, C. Roseti, and F. Zampognaro, "TCP Wave over Linux: A disruptive alternative to the traditional TCP window approach," *Computer Networks*, vol. 184, p. 107633, 2021.

[24] A. Abdelsalam., M. Luglio., C. Roseti., and F. Zampognaro., "TCP Wave Resilience to Link Changes - A New Transport Layer Approach Towards Dynamic Communication Environments," in *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications - DCNET, (ICETE 2016)*, INSTICC. SciTePress, 2016, pp. 72–79.

[25] A. Abdelsalam, M. Luglio, C. Roseti, and F. Zampognaro, "Analysis of Bandwidth Aggregation Techniques for combined use of Satellite and xDSL broadband links," *Int. J. Satell. Commun. Netw.*, vol. 37, no. 2, p. 76–90, feb 2019.

[26] ——, "TCP connection management through combined use of terrestrial and satellite IP-Based links," in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, 2017, pp. 37–42.

[27] G. Marfia, C. E. Palazzi, G. Pau, M. Gerla, and M. Roccetti, "TCP Libra: Derivation, Analysis, and Comparison with other RTT-fair TCPs," *Computer Networks*, vol. 54, no. 14, pp. 2327–2344, 2010.

[28] A. Bujari, A. Marin, C. E. Palazzi, and S. Rossi, "Analysis of ECN/RED and SAP-LAW with Simultaneous TCP and UDP Traffic," *Computer Networks*, vol. 108, pp. 160–170, 2016.