Supporting Smart Farming through Bandwidth Adaptation in Satellite Communications

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

Cristian Coreggioli
Department of Mathematics
University of Padua
Padua, Italy

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

Salah E. Merzougui
Department of Mathematics
University of Padua
Padua, Italy
merzougu@math.unipd.it

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

Lasse B. Schmidt
Department of Computer Science
University of Aarhus
Aarhus, Denmark
201805994@post.au.dk

ABSTRACT

Access to the Internet is a crucial enabler for many of the Sustainable Development Goals (SDGs) of the United Nations. Unfortunately, a significant part of the world's population is left behind due to the lack of access to a reliable and affordable Internet connection. Satellites have the potential to impact the current market of Internet services significantly. In particular, Low Earth Orbit (LEO) satellites promise high-bandwidth without compromising latency. They can be employed in 5G Non-Terrestrial Networks (e.g., IoT connectivity, connected autonomous driving, communication in rural areas, and more). Smart farming and precise agriculture (even remotely controlled), especially in underdeveloped areas, are compelling use cases for LEO satellites. In these scenarios, high bandwidth and low latency are required to facilitate both quick transmission of images/videos and prompt remote control of drones, tractors, actuators, etc. This study compares different TCP protocols based on their performance over satellite communication in a smart farming case study. It also proposes and analyzes a solution leveraging on a limited buffer size to maintain a high throughput while lowering per-packet delays.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

LEO Satellite Communications, TCP Protocols, Congestion Control, Smart Farming

ACM Reference Format:
Armir Bujari, Cristian Coreggioli, Mirko Franco, Salah E. Merzougui, Claudio E. Palazzi, and Lasse B. Schmidt. 2023. Supporting Smart Farming through Bandwidth Adaptation in Satellite Communications. In *ACM International Conference on Information Technology for Social Good (GoodIT '23), September 06–08, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3582515.3609520

1 INTRODUCTION

There is an emerging consensus worldwide that access to the Internet is a basic human right. Indeed, it is strongly related to some fundamental human activities and capabilities (e.g., participation in democracy, freedom of speech, social networking, smart agriculture, etc.) [30]. Unfortunately, a significant part of the population is left behind, especially in developing countries, due to the lack of access to a reliable and affordable Internet connection. The International Telecommunication Union (ITU) estimates that around half of the world population does not have Internet access at home [16]. Access to the Internet is, however, a key enabler for many of the Sustainable Development Goals (SDGs) of the United Nations, the global action plan for the next decade to end poverty and hunger, address basic human needs (e.g., health, education, job opportunities, social protection, etc.), tackle climate changes, and create sustainable societies and communities [27]. Universal and affordable access to the Internet for all is itself one of the SDGs. Unfortunately, although 95% of the global population has access to a mobile broadband connection, the coverage gap remains significant for least developed and developing countries, where 17% of the population cannot access a mobile broadband network [28].

In this context, several low-cost solutions have been proposed to open the Internet to developing regions and provide affordable connections to people living in remote areas with no or limited access to traditional communication networks [18]. One representative example is the adoption of satellites, especially the Low Earth Orbit (LEO) ones, to bring low-latency broadband connectivity to all areas of the world. Famous cases of disruptive technology in the field of satellite connections include the utilization of pico-satellites, also known as *CubeSats*, for Earth remote sensing, as well as the launches of thousands of satellites by companies such as *SpaceX* and

---

GoodIT '23, September 06–08, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0116-0/23/09…$15.00
https://doi.org/10.1145/3582515.3609520

Figure 1: Smart Farming Scenario.

OneWeb, which promise high-bandwidth and low-latency connectivity, thus having the potential to significantly impact the current market of Internet services.

Satellites are currently the only solution available to provide global access to Internet services, even in remote areas (e.g., over the ocean [2]). In particular, Low Earth Orbit (LEO) satellites promise high-bandwidth and low-latency connectivity, making them a crucial tool in reducing the digital divide among various regions of the globe. For these reasons, researchers have proposed numerous applications based on them. For instance, they can be employed in 5G Non-Terrestrial Networks (e.g., for IoT, connected autonomous vehicles, maritime communications, rural areas connectivity, etc.), aeronautical tracking systems, earth observation, and space communications [19].

Smart farming, generally depicted in Figure 1, embodies an intriguing application of LEO satellites for their potential in supporting the SDGs and for the peculiar characteristics of the generated network traffic. For instance, consider the scenario where high quality images of the crops or videos of the ongoing operations have to be transmitted, elaborated and then acted upon to send back control messages to a remote actuator. In this case, we need both high bandwidth and low per-packet latency to support fast downloads and timely remote control.

In this context, this paper aims to compare some TCP protocols, even specifically devised for satellite links, considering smart farming as a case study and employing throughput and Round Trip Time (RTT) as metrics. We also propose and analyze a solution based on limiting buffers in order to maintain a high throughput while lowering per-packet delays. Our solution does not rely on a specific TCP variant as its use of the advertised window make it applicable to any TCP version. We hence demonstrate that LEO satellites are a promising tool in reducing the digital divide and providing Internet connectivity to all, effectively supporting the SDGs.

The rest of this paper is organized as follows. Section 2 presents a review of the relevant literature. Section 3 overview the compared protocols including our proposed solution. We describe the considered scenario and the parameters employed in the simulations in Section 4, while the results of our experiments are presented in Section 5. Finally, we draw our conclusions and present some future research directions in Section 6.

2 RELATED WORK

TCP was first proposed around the 1970s [10]. Since the beginning, it went through several evolutions [4, 12, 17]. With the popular spread of satellite technologies at the beginning of the twentieth century, researchers have developed variants specifically designed for satellite communication, such as Westwood [13, 14], which is one of the most representative implementations of this genre. Furthermore, latency and congestion control have become a big concern as the world has become increasingly connected since we have shifted from wired connections to wireless ones. To answer these concerns, Google published the bottleneck bandwidth and round-trip time (TCP BBR) congestion control algorithm in 2016 [9].

Nowadays, satellite networks have become an essential part of our network infrastructure due to their massive reach, easy deployment, and high throughput. However, satellite latency remain an issue. Satellites orbit between 1,000 to 36,000 km above the Earth, so the physics for communication between terrestrial hosts using a satellite means ranges between 40 to 600 ms RTT at a minimum depending on the satellite orbit, which is a challenge for TCP-based protocols [5, 24].

Obata *et al.* [25] assessed TCP performance over genuine (rather than simulated) satellite networks. They contrasted New Reno and Hybla with a satellite-oriented TCP congestion management technique (STAR). Experiments with the Wide-band Inter-networking Engineering test and Demonstration Satellite (WINDS) network revealed throughput of around 26 Mb/s and an RTT of approximately 860 ms. TCP STAR and TCP Hybla had higher throughput than TCP New Reno over the satellite link. In the framework of Digital Video Broadcasting-Return Channel via Satellite, Kuhn *et al.* [20] investigated the performance of TCP over random and dedicated access techniques. Utsumi *et al.* [29] created a TCP Hybla analytic model for steady-state throughput and RTT over satellite networks. They validated their model's accuracy using simulated and emulated satellite networks (bandwidth 8 Mb/s, RTT 550 ms, and up to 2% packet loss rates). According to their results, TCP Hybla offers much higher throughput than legacy TCP for packet loss rates greater than 0.0001%.

Giambene *et al.* [15] explored the numerous pathways of terrestrial and satellite infrastructures in order to maximize TCP throughput simultaneously. Bacco *et al.* [3] described the interaction between TCP at the transport layer and random-access techniques at the Media Access Control (MAC) layer for machine-to-machine services over satellite connections. Pokhrel *et al.* [26] investigated the performance of multi-path TCP over a network that included a server linked to a satellite, as well as drones and multiple WiFi access points for Internet access. Liu *et al.* [21] analyzed TCP New Reno's performance via satellite-based links. Claypool *et al.* [11] compared TCP congestion control strategies' performance on a commercial satellite Internet network. The analysis demonstrated that all strategies have equal steady-state bit-rates, but there are considerable disparities in start-up throughput and RTT due to packet queuing in flight. A comprehensive survey about several aspects of satellite connections is presented in [19].

Some works also considered the QUIC protocol. For instance, Adami *et al.* [8] analyzed the network traffic generated by two social media, which employ even QUIC as the transport protocol. Yet, its behaviour in satellite links is not fully known and deserves further investigation [1, 22, 23].

3 COMPARED PROTOCOLS

TCP is a two-way, reliable, byte-stream-oriented, end-to-end transport protocol which provides also flow and congestion control. Unfortunately, its mechanism for congestion control was designed for a reliable medium, where losses were always considered a sign of congestion, and is not appropriate for the current wireless environment. Over the years, several variants of the TCP protocol specifically designed for the wireless environment and satellite links have been developed. We briefly described some representative cases used in our experiments, including our proposed solution based on limiting the buffer size in order to ensure both high throughput and low per-packet delay.

3.1 TCP New Reno

TCP New Reno is one of the first developed versions of TCP. When a packet is lost, TCP New Reno retransmit the packet and generally reduces the transmission speed. In particular, if a packet loss is detected through the expired RTO, it is consider as an indicator of massive congestion, and the new *cwnd* is set to 1. Then, Slow Start phase is applied.
On the contrary, when the sender receives three duplicate ACKs, the segment is retransmitted but the new transmission speed is just half of what it was before the packet loss.

As it is well known, TCP New Reno performs poorly when we have multiple packet losses in one window (it may happens in wireless links), long RTTs and satellite links in general.

3.2 TCP BBR

TCP BBR ("Bottleneck Bandwidth and Round-trip propagation time") is one of the more recent TCP protocols developed in 2016 by Google [9]. According to Google, BBR was developed to operate best on bad wireless connections as this was the only time average users experienced poor internet performance. Its congestion control aims to constantly operate the TCP session right at the point of onset of queuing. To do this, BBR keeps a model of the network, which is constantly updated as it sends packets. This model keeps the maximum recent bandwidth available and the minimum recent RTT. BBR then uses this model to decide how quickly data will be sent and how much data it is willing to allow to simultaneously be present in the network. To ensure an accurate model, it systematically probes the network by sending out data with an increased sending rate of 25%. If more bandwidth is available, BBR updates the model to take advantage of the newly freed-up bandwidth.

3.3 TCP Westwood

TCP Westwood [13, 14] is a pure end-to-end variant of the TCP protocol, devised to improve the performances of the traditional versions (e.g., TCP New Reno), especially in so called *big leaky pipes*, i.e., error-prone satellite links. The key idea is to continuously estimate, at the TCP sender, the packet rate of the connection by averaging the rate of returning ACKs. The estimated connection rate is then used to compute the congestion window and the slow start threshold which are employed after a congestion occurs (after three duplicate acknowledgments or after an expired timeout).

TCP Westwood comes from the idea that if a connection is currently achieving a certain transmission rate, then it can safely use the window corresponding to that rate without causing congestion in the network and without unnecessarily shrinking its transmission speed when some (wireless) loss occurs. It attempts to select the *ssthresh* and a *cwnd* which are consistent with the effective bandwidth used at the time of detected congestion. This approach makes TCP Westwood more robust to sporadic losses, especially in wireless domain.

3.4 A Solution Based on Limiting the Buffer Size

The actual sending window of TCP is computed as the minimum between the congestion window and the advertised window (provided by the receiver in the TCP acknowledgments). If we limit the size of the advertised window to a (small) proper value, congestion will not build up and queuing delays will stay very low. Basically, this solution limits the sending rate to a proper value that ensures an efficient utilization of the available bandwidth, while avoiding to creating queues [6, 7]. The receiver determines the proper value for this limitation by considering ongoing per-packet delays with respect to a minimum RTT (without queuing delays) and the bandwidth present in the bottleneck (which is known as it corresponds the terrestrial access link). This gives a reference value for the receiver buffer size that will then be monitored by TCP to determine the advertised window provided in its acknowledgment packets. This represents a rule of thumbs that can be slightly modified even empirically depending on the instability of the connection, resorting to more or less conservative approaches. In our experiments we considered different buffer sizes at the receiver and, as a representative case, we report here the outcome for this solution in the case with the buffer size set to the bandwidth-RTT product augmented by the size of three TCP packets.

The obtained throughput is comparable to those achieved with the use of a large window size; yet, we also ensure a lower and more stable over time per-packet-delay, by adjusting the value of the advertised window. Since our solution requires only to modify a parameter via software, it is possible to deploy and adapt to different scenarios and applications. Furthermore, as we show in our experiments, it does not rely on a specific TCP variant and can be applied to any TCP version as all of them use the advertised window as a flow control tool.

4 SCENARIO

In this section, we present the considered scenario for our experiment and the parameters employed in the simulations. Our envisaged scenario represents a LEO satellite communication using three different TCP protocols, the first one is legacy protocol (TCP New Reno), the second is designed for satellite links (TCP Westwood) and the third is designed for loss-based congestion control (TCP BBR). We considered six nodes: node 1 represents the server, node 2 represents an antenna (the satellite gateway), node 3 represents the LEO satellite, node 4 represents a second antenna (the satellite terminal), node 5 represents the access point, and node 6 represents the client (Tractors, Drones, Sensors and cameras, etc.). These six nodes are connected through two types of communication links; three terrestrial links represented by link 1 connecting node 1 to node 2, link 4 connecting node 4 to node 5, and link 5 connecting

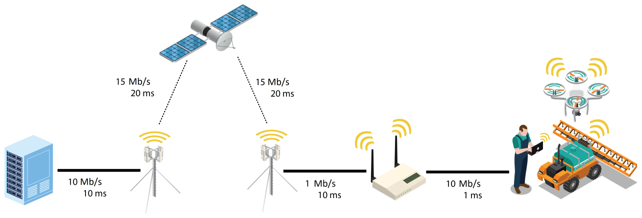


Figure 2: Topology of the Network.

Table 1: Parameters of the Simulations

Parameter	Value
Simulation Time	50 s
Bandwidth of the Bottleneck	1 Mb/s
Propagation Delay of the Bottleneck	10 ms
Maximum Transmission Unit (MTU)	1500 B
Maximum Segment Size (MSS)	1446 B
Data Rate of the Application	9 Mb/s
Versions of TCP	New Reno, BBR, Westwood

node 5 to node 6; and two satellite links represented by link 2 connecting node 3 to node 2, and link 3 connecting node 3 to node 4 (see Figure 2). We implemented the topology under the well-known ns-3 simulation tool, version NS-3.36.1, installed on Ubuntu operating system version 20.04 LTS, where each link corresponds to a Point-to-Point connection with a specific data rate and delay, as shown in Figure 2.

We set the delay of the terrestrial links as follows: link 1 and link 4 have a delay of 10 ms and link 5 has a delay of 1 ms. The data rates were set to 10 Mb/s for links 1 and 5 and 1 Mb/s for link 4 (the bottleneck of the connection). When we use regular TCP protocols without buffer limitation, the TCP buffer size has been set very high (1GB). We configured the one way delay on both satellite links to 20 ms to simulate the delay typically caused by the altitude of a LEO satellite and the data rate to 15 Mb/s. Then, we implemented a custom TCP application to generate the data traffic. We attached to the application two functions to help us monitor and study the behavior of the RTT and the Congestion Window for each TCP variant. The TCP application creates packets of 1446 B and sends them from the server to the client at a data rate of 9 Mb/s for a duration of 50 s (see Table 1).

5 PERFORMANCE EVALUATION

In our simulations an application sends data from the server to the client, as described in Section 4. The considered experimental settings are the following:

- **First Experiment**: the error rate on the satellite links is equal to 0;
- **Second Experiment**: the error rate of the link between the antenna and the satellite is equal to 0.005;

Table 2: Throughput summary for each considered TCP variant and experiment.

Variant of TCP	First Exp.	Second Exp.	Third Exp.
TCP New Reno	0.956211	0.702409	0.95783
TCP BBR	0,933075	0,926365	0,938628
TCP Westwood	0.93909	0.859734	0.95783

- **Third Experiment**: the error rate of all the links is equal to 0 and the buffer size for all three TCP versions is limited by our solution.

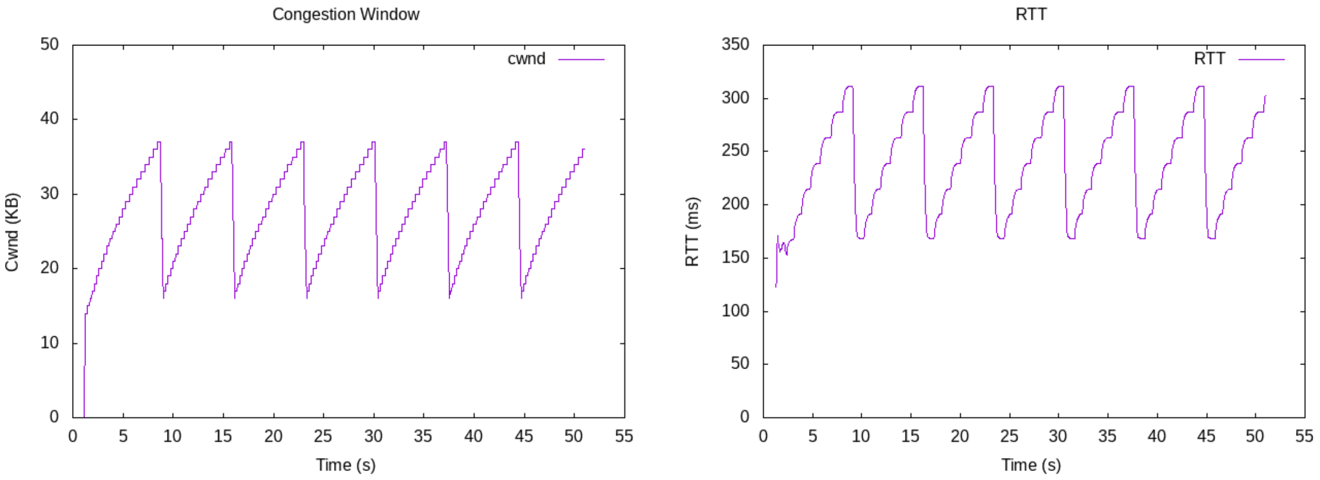
The data rate of the application is 9 Mb/s, while the bandwidth of the bottleneck is 1 Mb/s, so after a certain amount of time, the channel will be saturated, thus causing packet losses. In this way, we aim to create congestion in the network so that we can observe the effects of the congestion control mechanism of TCP. This behaviour is visible from the charts related to the congestion window in Figure 3. We can notice that TCP BBR had 5 packet loss events, while TCP New Reno and TCP Westwood had 6 each. Yet, the throughput achieved by TCP Westwood and TCP BBR is almost the same, while with TCP New Reno we obtain a slightly higher value (see Table 2).

When we introduce a probability of 0.5% to lose packets in the link from the antenna to the satellite, we will have many more peaks in the shape of the congestion window as the TCP needs to retransmit more lost packets (see Figure 4). As mentioned above, each TCP variant differs in the way of computing the congestion window, as we can notice from the charts shown by Figure 4. The most aggressive behaviour is the one of TCP Westwood since the congestion window drops to almost zero at every lost packet. TCP New Reno behaves in a similar way. In contrast, with TCP BBR the values are higher when compared to those of the other variants. Also it is the quickest to get the congestion window back up to its limit again. When looking at the throughput we also see that TCP BBR outperforms the other two protocols in this test, almost maintaining the same throughput achieved in experiment 1. Indeed, this is the TCP version among the three considered here that has the most efficient use of the channel (see Table 2).

Finally, we limit the size of the TCP buffers as mentioned in Section 3.4 to set an upper bound to the sending rate through the advertised window. We notice that the congestion window keeps growing (probably indefinitely) for TCP New Reno and TCP Westwood as no congestion loss will occur (see Figure 5). This happens as the actual sending rate is limited by the advertised window rendering the congestion window's growth useless. However, for TCP BBR, we see that there are still a few packet losses. It is fewer than in the first experiment, but there are still a few. TCP BBR still probes the internet for more bandwidth, just less frequently. In this way, we obtain limited, stable over time and predictable delays while reaching the same (or even better) throughput with respect to the first experiment.

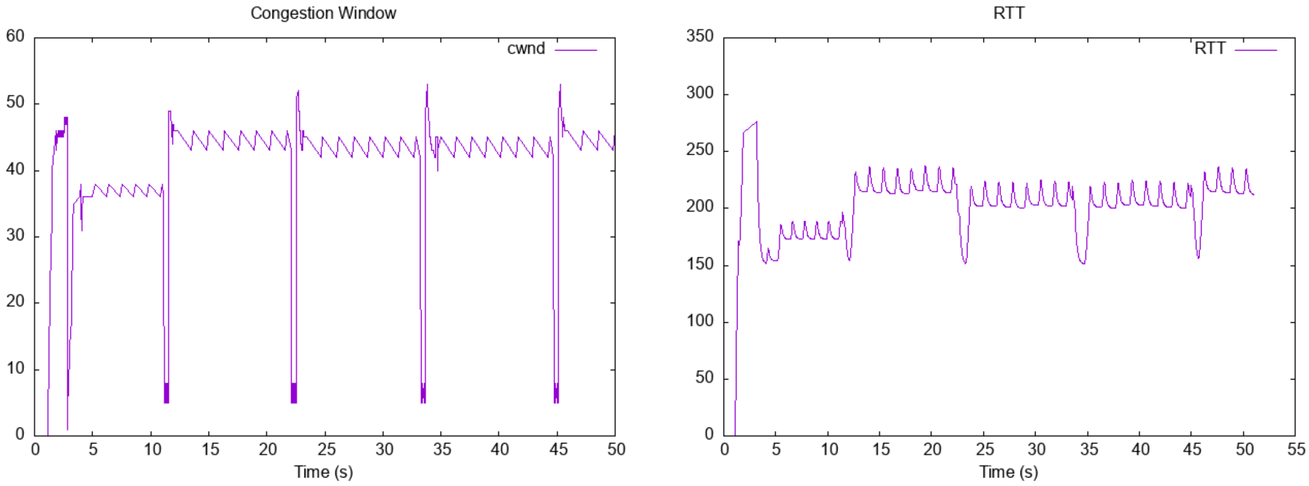
6 CONCLUSION

The increasing recognition of connectivity as a fundamental human right is in stark contrast to the anticipated lack of achievement of ensuring access to reliable and cost-effective Internet connectivity



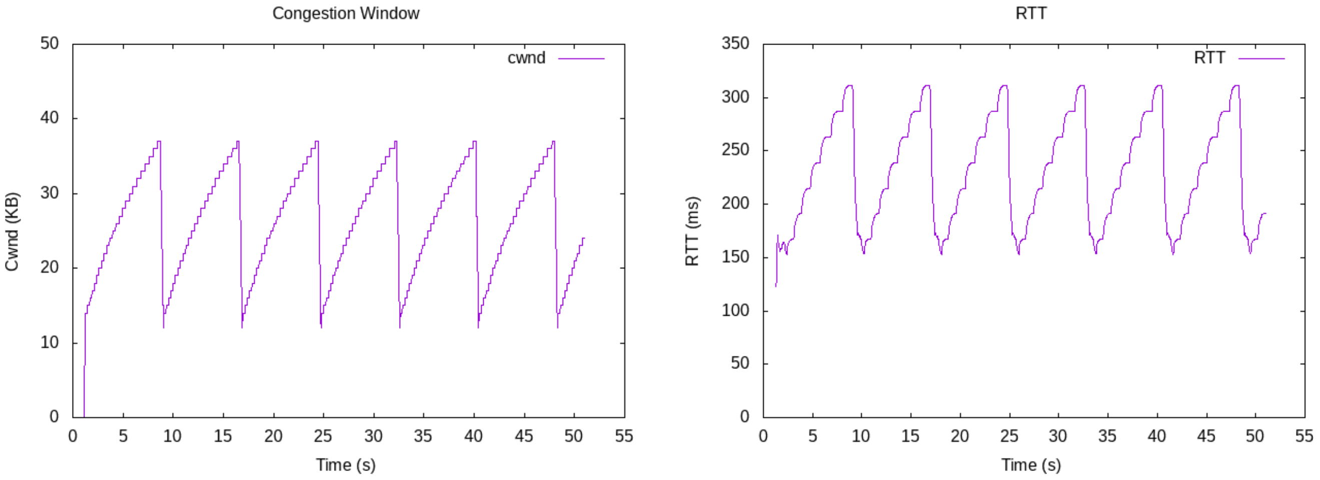
(a) Congestion Window for TCP New Reno

(b) RTT for TCP New Reno.



(c) Congestion Window for TCP BBR.

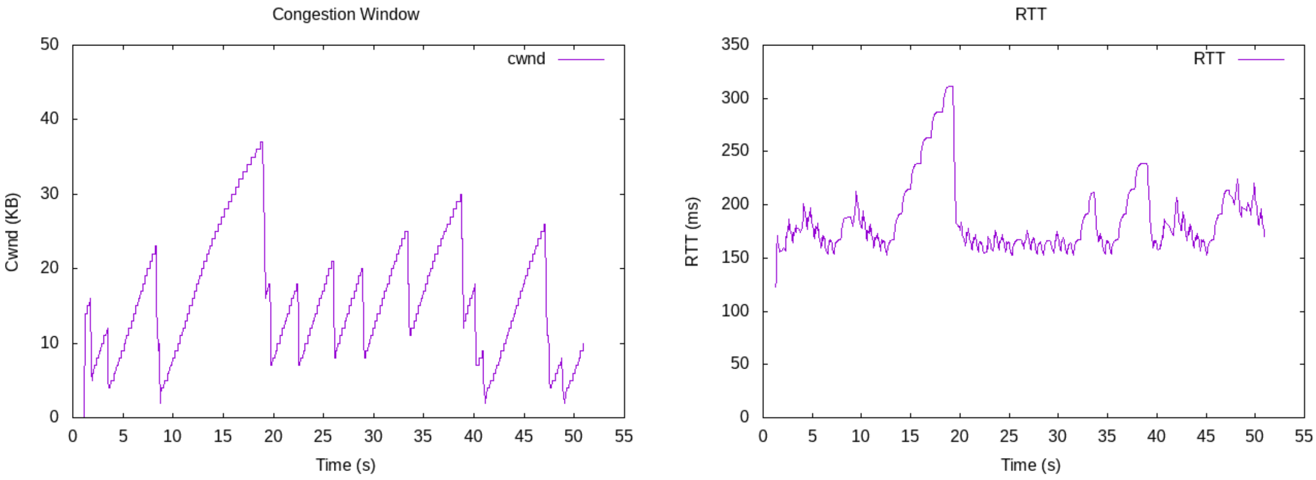
(d) RTT for TCP BBR.



(e) Congestion Window for TCP Westwood.

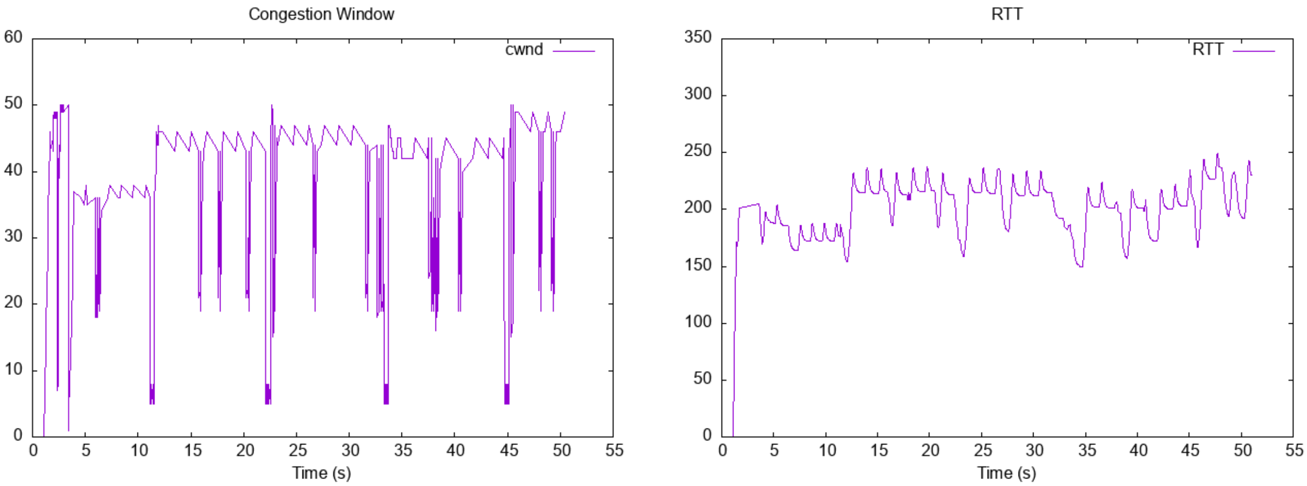
(f) RTT for TCP Westwood.

Figure 3: First Experiment: Congestion Window and RTT for the considered versions of TCP



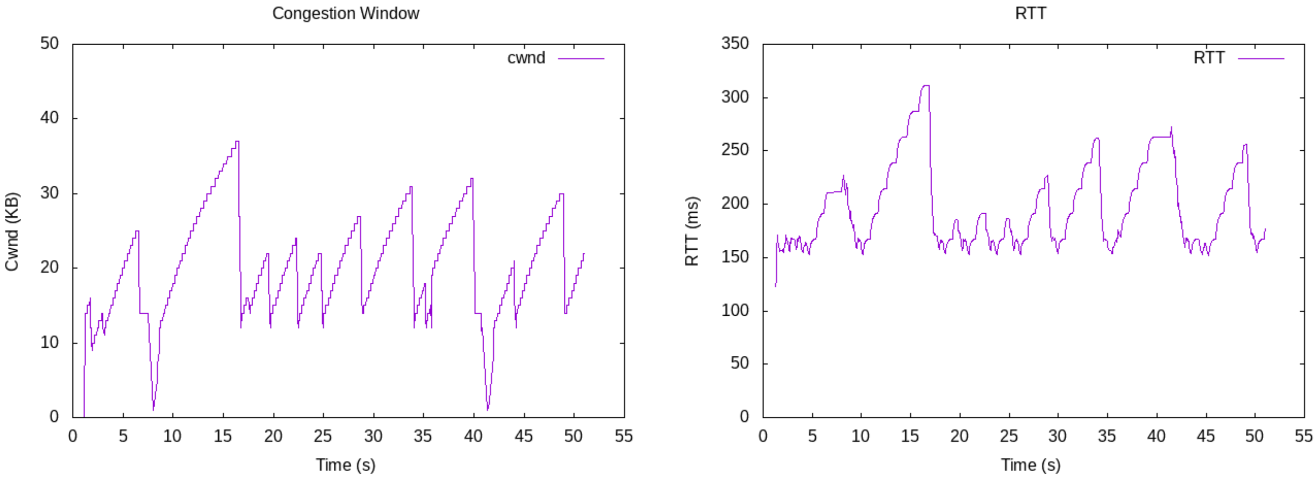
(a) Congestion Window for TCP New Reno

(b) RTT for TCP New Reno.



(c) Congestion Window for TCP BBR.

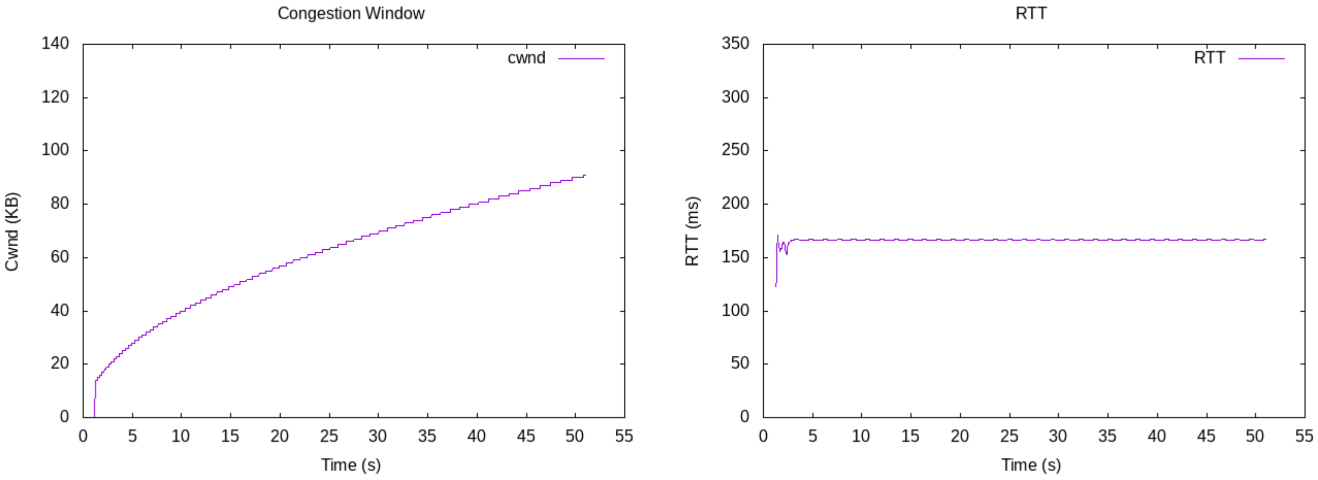
(d) RTT for TCP BBR.



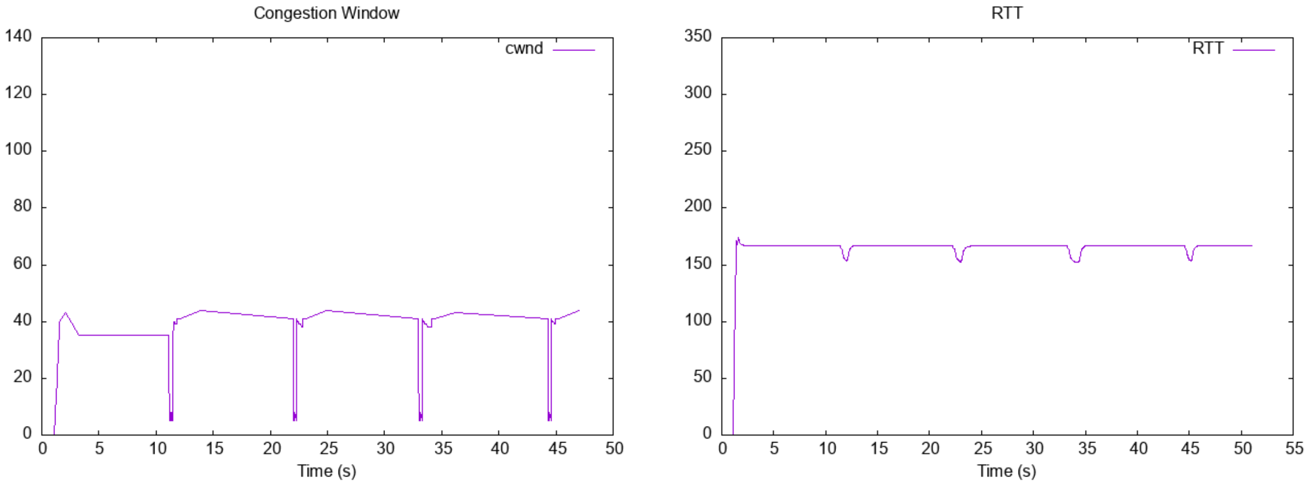
(e) Congestion Window for TCP Westwood.

(f) RTT for TCP Westwood.

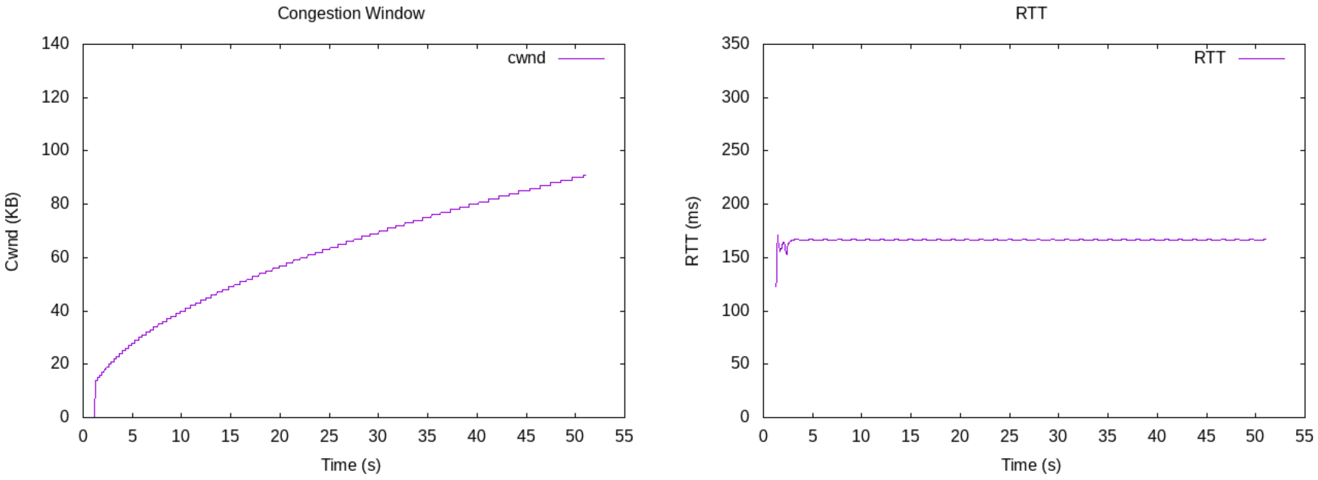
Figure 4: Second Experiment: Congestion Window and RTT for the considered versions of TCP with errors



(a) Congestion Window for TCP New Reno **(b) RTT for TCP New Reno.**



(c) Congestion Window for TCP BBR. **(d) RTT for TCP BBR.**



(e) Congestion Window for TCP Westwood. **(f) RTT for TCP Westwood.**

Figure 5: Third Experiment: Congestion Window and RTT for the considered versions of TCP with limited buffer size.

for all by 2030. In this context, satellites represent a promising solution in providing Internet connectivity even in remote areas, where a terrestrial infrastructure would be difficult to build and not cost-efficient. In particular, LEO satellites can provide both high-bandwidth and low-latency links.

Since the performance of TCP is affected by satellite links, we have analyzed some representative variants, considering throughput and RTT. We have shown how a simple solution limiting the buffer usage and exploiting the advertised window feature present in all TCP versions can be coupled with any of the considered TCP variants to improve RTT while preserving throughput. This allows LEO satellite systems to be employed in underdeveloped areas, without proper terrestrial Internet infrastructure, to support Smart Farming (and other crucial applications for SDGs).

We intend to extend our research in several directions. For instance, we plan to compare the per-packet end-to-end delay to investigate the behaviour of the different versions of the protocol. Moreover, we would like to consider more flows, even the coexistence of satellite and not-satellites ones, and measure properties such as fairness and friendliness. Finally, investigating the performance of QUIC in networks involving satellite links would be interesting.

ACKNOWLEDGMENTS

The authors of this paper acknowledge the support of MUR (Italian Ministry of University and Research) under the PON initiative ex DM 1061 and the support of ESA under the Artes At 6b.084 initiative (QUICoS project).

REFERENCES

[1] A. Abdelsalam, M. Luglio, M. Quadrini, C. Roseti, and F. Zampognaro. 2019. QUIC-proxy based architecture for satellite communication to enhance a 5G scenario. In *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. 1–6.

[2] Abdelmoula Ait Allal, Khalifa Mansouri, Mohamed Youssfi, and Mohammed Qbadou. 2018. Reliable and Cost-Effective Communication at High Seas, for a Safe Operation of Autonomous Ship. In *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. 1–8.

[3] Manlio Bacco, Tomaso De Cola, Giovanni Giambene, and Alberto Gotta. 2019. TCP-Based M2M Traffic via Random-Access Satellite Links: Throughput Estimation. *IEEE Trans. Aerospace Electron. Systems* 55, 2 (2019), 846–863.

[4] Lawrence S. Brakmo and Larry L. Peterson. 1995. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE J. Sel. Areas Commun.* 13 (1995), 1465–1480.

[5] Armir Bujari, Michele Luglio, Claudio E. Palazzi, Mattia Quadrini, Cesare Roseti, and Francesco Zampognaro. 2020. A Virtual PEP for Web Optimization over a Satellite-Terrestrial Backhaul. *IEEE Communications Magazine* 58, 10 (2020), 42–48.

[6] Armir Bujari, Andrea Marin, Claudio E. Palazzi, and Sabina Rossi. 2016. Analysis of ECN/RED and SAP-LAW with simultaneous TCP and UDP traffic. *Computer Networks* 108 (2016), 160–170.

[7] Armir Bujari, Michele Massaro, and Claudio Enrico Palazzi. 2015. Vegas Over Access Point: Making Room for Thin Client Game Systems in a Wireless Home. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 12 (2015), 2002–2012.

[8] Armir Bujari, Claudio E. Palazzi, Giacomo Quadrio, and Daniele Ronzani. 2020. Emerging Interactive Applications over QUIC. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. 1–4.

[9] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2016. BBR: Congestion-Based Congestion Control: Measuring Bottleneck Bandwidth and Round-Trip Propagation Time. *Queue* 14, 5 (oct 2016), 20–53. https://doi.org/10.1145/3012426.3022184

[10] V. Cerf and R. Kahn. 1974. A Protocol for Packet Network Intercommunication. *IEEE Transactions on Communications* 22, 5 (1974), 637–648.

[11] Saahil Claypool, Jae Chung, and Mark Claypool. 2021. Comparison of TCP Congestion Control Performance over a Satellite Network. In *Passive and Active Measurement: 22nd International Conference, PAM 2021*. Springer-Verlag, Berlin, Heidelberg, 499–512. https://doi.org/10.1007/978-3-030-72582-2_29

[12] Kevin R Fall and W Richard Stevens. 2011. *TCP/IP illustrated, volume 1: The Protocols*. addison-Wesley.

[13] M. Gerla, M.Y. Sanadidi, Ren Wang, A. Zanella, C. Casetti, and S. Mascolo. 2001. TCP Westwood: Congestion Window Control Using Bandwidth Estimation. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. 287–297.

[14] M. Gerla, M.Y. Sanadidi, Ren Wang, A. Zanella, C. Casetti, and S. Mascolo. 2001. TCP Westwood: Congestion Window Control Using Bandwidth Estimation. In *GLOBECOM'01. IEEE Global Telecommunications Conference*, Vol. 3. 1698–1702 vol.3.

[15] Giovanni Giambene, Doanh Kim Luong, Tomaso de Cola, Van Anh Le, and Muhammad Muhammad. 2019. Analysis of a Packet-Level Block Coding Approach for Terrestrial-Satellite Mobile Systems. *IEEE Transactions on Vehicular Technology* 68, 8 (2019), 8117–8132.

[16] International Telecommunication Union (ITU). 2020. Measuring Digital Development. Facts and Figures. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2020.pdf. Accessed 21 January 2023.

[17] V. Jacobson. 1988. Congestion Avoidance and Control. In *Symposium Proceedings on Communications Architectures and Protocols* (Stanford, California, USA) *(SIGCOMM '88)*. Association for Computing Machinery, New York, NY, USA, 314–329. https://doi.org/10.1145/52324.52356

[18] Keith Kirkpatrick. 2018. Bringing the Internet to the (Developing) World. *Commun. ACM* 61, 7 (jun 2018), 20–21.

[19] Oltjon Kodheli, Eva Lagunas, Nicola Maturo, Shree Krishna Sharma, Bhavani Shankar, Jesus Fabian Mendoza Montoya, Juan Carlos Merlano Duncan, Danilo Spano, Symeon Chatzinotas, Steven Kisseleff, Jorge Querol, Lei Lei, Thang X. Vu, and George Goussetis. 2021. Satellite Communications in the New Space Era: A Survey and Future Challenges. *IEEE Communications Surveys & Tutorials* 23, 1 (2021), 70–109.

[20] Nicolas Kuhn, Huyen Chi Bui, Jérôme Lacan, José Radzik, and Emmanuel Lochin. 2013. On the Benefits of Random Access Methods on TCP Performance over DVB-RCS2. *CoRR* abs/1301.5931 (2013). arXiv:1301.5931 http://arxiv.org/abs/1301.5931

[21] Jiong Liu, Zhu Han, and Wei Li. 2020. Performance Analysis of TCP New Reno over Satellite DVB-RCS2 Random Access Links. *IEEE Transactions on Wireless Communications* 19, 1 (2020), 435–446.

[22] Michele Luglio, Mattia Quadrini, Simon Pietro Romano, Cesare Roseti, and Francesco Zampognaro. 2020. Enabling an Efficient Satellite-Terrestrial Hybrid Transport Service through a QUIC-Based Proxy Function. In *International Symposium on Networks, Computers and Communications Architectures and Protocols (ISNCC)*. IEEE, 1–6.

[23] M. Luglio, M. Quadrini, C. Roseti, F. Zampognaro, and S. P. Romano. 2020. A QUIC-based Proxy Architecture for an Efficient Hybrid Backhaul Transport. In *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. 144–146.

[24] Gustavo Marfia, Claudio E. Palazzi, Giovanni Pau, M. Gerla, and Marco Roccetti. 2010. TCP Libra: Derivation, analysis, and comparison with other RTT-fair TCPs. *Computer Networks* 54, 14 (2010), 2327–2344.

[25] Hiroyasu Obata, Kazuya Tamehiro, and Kenji Ishida. 2011. Experimental Evaluation of TCP-STAR for Satellite Internet over WINDS. In *10th International Symposium on Autonomous Decentralized Systems*. 605–610.

[26] Shiva Raj Pokhrel, Jiong Jin, and Hai Le Vu. 2019. Mobility-Aware Multipath Communication for Unmanned Aerial Surveillance Systems. *IEEE Transactions on Vehicular Technology* 68, 6 (2019), 6088–6098.

[27] United Nations. 2015. Transforming our World: The 2030 Agenda for Sustainable Development. https://wedocs.unep.org/20.500.11822/9814. Accessed 23 January 2023.

[28] United Nations. 2022. The Sustainable Development Goals Report 2022. https://unstats.un.org/sdgs/report/2022/. Accessed 23 January 2023.

[29] Satoshi Utsumi, Salahuddin Muhammad Salim Zabir, Yu Usuki, Seisho Takeda, Norio Shiratori, Yasushi Kato, and Jeyeon Kim. 2018. A New Analytical Model of TCP Hybla for Satellite IP Networks. *J. Netw. Comput. Appl.* 124 (2018), 137–147.

[30] Stephen B. Wicker and Stephanie M. Santoso. 2013. Access to the Internet is a Human Right. *Commun. ACM* 56, 6 (jun 2013), 43–46.