Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: https://oatao.univ-toulouse.fr/20713

To cite this version :

Tauran, Bastien and Lochin, Emmanuel and Lacan, Jérôme and Arnal, Fabrice and Gineste, Mathieu and Kuhn, Nicolas Controlled Delay Scheduler for VoIP over LEO constellations on LMS channels. (2018) In: 9th Advanced Satellite Multimedia Systems Conference, 10 September 2018 - 12 September 2018 (Berlin, Germany).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Controlled Delay Scheduler for VoIP over LEO constellations on LMS channels

Bastien Tauran, Emmanuel Lochin, Jérôme Lacan

Université de Toulouse ISAE-SUPAERO TéSA
Toulouse, France
firstname.lastname@isae-supaero.fr

Fabrice Arnal, Mathieu Gineste

Thales Alenia Space,
Toulouse, France
fistname.lastname@thalesaleniaspace.com

Nicolas Kuhn

Centre National d'Etudes Spatiales,
Toulouse, France
firstname.lastname@cnes.fr

***Abstract*—Satellite transmissions can suffer from high channel impairments, especially on the link between a satellite and a mobile end-user. To cope with these errors, physical and link layer reliability schemes have been introduced at the price of an end-to-end delay increase resulting in high jitter. Unfortunately, both the delay and the jitter negatively impacts on multimedia traffic. As a matter of fact, not taking into account the channel state greatly decreases the Quality of Experience (QoE) of VoIP users. In this paper, we propose to solve this issue by scheduling data transmission as a function of the channel condition. We first investigate existing scheduling mechanisms and analyze their performance for VoIP traffic with the objective to lower both latency and jitter, which are the most important metrics to achieve a consistent VoIP service. We select the best candidate among several schedulers and propose a novel algorithm specifically designed to carry VoIP over LEO constellations. Our simulations show that in some scenarios, we double the QoE of VoIP users.**

I. Introduction

Low Earth Orbit (LEO) satellite constellations allow to provide internet access to isolated or rural areas anywhere on earth. Different kind of traffic can be transmitted through these constellations: best-effort, VoIP or video. LEO delays are in the same order of magnitude than on terrestrial networks, but the high channel constraints, mostly on Land Mobile Satellite (LMS) channels [1], [2], implies the need of efficient reliability mechanisms. However, their use has an impact on the end-to-end transmission delay and jitter.

To cope with these high channel impairments on the LMS channel, reliability mechanisms have been introduced [3], [4], [5]. One of the most efficient is Hybrid Automatic Repeat reQuest (HARQ) which combines forward error-correction codes and link layer retransmission. This mechanism is introduced on the LMS link between the last satellite on the packet route and the ground receiver. We consider in this paper type II HARQ, which is commonly deployed inside physical layers. Depending on the channel quality at the moment of the transmission, the duration of the decoding of a packet on the ground receiver might vary and the cost in terms of capacity (linked to the redundancy ratio) might also vary on the LMS link. At last but not least, this obviously also increases the jitter of the communication.

When several users are transmitting data over a LMS channel, they compete for capacity over the same link which becomes the bottleneck of the network. If no scheduling policy is enabled, all packets are queued in a FIFO manner and dequeued without taking into account the quality of the channel before transmission. This problem is not new, scheduling transmission to optimize the use of the LMS link has been deeply tackled by the satellite networking community [6], [7], [8]. In particular in [9], [10], different metrics to schedule the packets, such as the throughput or the waiting time of the packets in the queue are investigated. However and in our context, our contribution leads to assess which scheduler would allow to deploy a VoIP service while maximizing user QoE. We review in the following a set of potential scheduling mechanisms that could be used in our context considering the service we seek to obtain for the VoIP flows. This state of the art leads us to the choice of Proportional Fairness, due to the way this scheduler optimizes the channel capacity. During our evaluations, we observed that queuing management should be conjointly considered with the scheduling. Thus, our proposal, Controlled Delay Scheduler (CoDeS), is based on the joint use of a scheduler and a queue management scheme.

To deploy such a service, one possible solution could have been to use a Call Admission Control (CAC) [11]. However, the high variability of the channel over time makes CAC not suitable. Ensuring each user to always have a minimum capacity guaranteed would involve to continuously monitor the channel to assess a kind of worst case of the channel capacity. In our case, the channel can in rare occasions reach an attenuation of 60 dB, and taking this worst case as a reference would imply a very low number of users that can share the channel, which would result in an under-utilization of the link capacity. Our proposal makes a compromise between the number of users sharing the channel with average good conditions, and the use of the LMS channel capacity.

In Section II, we detail the scenario used in our simulations, the LMS reliability scheme and the networking conditions. We also present the different schedulers tested. In Section IV, we present the results measured on the most important metrics such as throughput, latency, losses and jitter. We then analyze these results in Section V. Following this analysis, we propose CoDeS in Section VI and present the benefits obtained in our context.

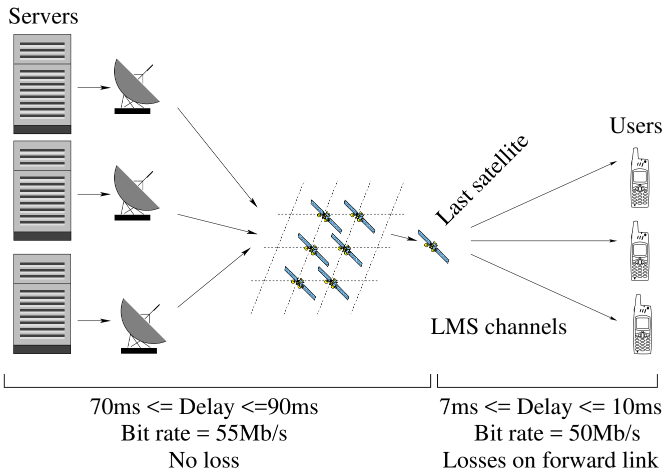

Fig. 1: Model for a satellite constellation

II. Scenario

This section presents the satellite scenario, how we simulate the satellite environment, the different schemes that are considered throughout this paper and in particular: the reliability mechanisms on the LMS channel to deal with the high error rate, the kind of traffic used and the schedulers tested.

A. Satellite environment

We have chosen Network Simulator 2 (ns-2) to simulate the satellite environment. The scenario is composed of 66 LEO satellites at an altitude of 800 km, ensuring a global coverage of any point on earth at any time. Due to the movement of the satellites and route changes, the transmission delay varies from 70 ms to 90 ms within the satellite constellation (data obtained with `SaVi` [12] and detailed in the following). Moreover, except on the LMS forward link, we consider an error free path, as shown in Figure 1, between the sender and the last satellite. Finally, messages are sent from the server to mobile receivers through the satellite constellation.

In our simulations, several transmissions are passing through the constellation, from different sources, to different destinations, but having always the same last satellite on their route path. The last hop is from the same satellite to different ground receivers, involving independent LMS channels. In this scenario, the last satellite and the LMS channels are the bottleneck of the network.

To mimic the topology illustrated in Figure 1, we use three nodes to represent the sender, the last satellite on the message route, and the receiver. The satellite constellation is simulated by varying the delays between the nodes. We used `SaVi` [12] to get the parameters of the previously described LEO satellite constellation. Then we simulate the constellation to assess the evolution of the delays between the sender and the last satellite, and between the last satellite and the ground receiver. From this point, all ns-2 simulations are played using the three nodes and the temporal traces generated. The simulations are run over a LMS channel between the last satellite and the ground gateway in an Intermediate Tree Shadowed (ITS) environment. In our simulations, we use channel model where the Doppler shift has been estimated and compensated, thus we do not need to take it into account in this study. We vary the average quality of this channel by setting a reference Signal Noise Ratio (SNR) ranging from 7 dB to 13 dB. During the simulations, the link quality changes over the time around this SNR reference value. Each simulation lasts 100 s with a varying number of flows performing. Each flow uses the same temporal trace but with different offset values to simulate different route paths and transmission delays. In the same way, each flow uses the LMS channel with a different offset value to simulate independent channels. Moreover, we improved the confidence of the results by running several simulations starting the temporal traces at different times to obtain a consistent statistical set and analysis.

B. Hybrid Automatic Repeat reQuest

We previously explained that HARQ schemes are used to mitigate link-layer impairments on the LMS channels. These schemes aim to compensate the high error rate characterizing such channels and optimize their usage by combining both ARQ and Forward Error Correction codes. We choose a recent proposal called Adaptive-HARQ [13], which is an improvement of type II HARQ. This scheme uses mutual information to compute the optimal number of bits to send at each retransmission and allows up to 3 retransmissions in case of erased packets. The principle detailed in Figure 2 is as follows: the receiver side of the HARQ link stores the bits received while a packet has not been decoded. Each time useful or redundancy bits are received, the algorithm computes whether enough data has been received to decode the packet. If not, a negative ACK is sent to the HARQ sender, asking for more redundancy bits. In the worst case, the packets are recovered at most $70ms$ after entering HARQ module, which corresponds to the time needed to get both the first transmission and the 3 retransmissions.

The varying transmission time leads to increase the flow jitter, each additional retransmission of HARQ makes the packet decoded 20 ms later, highly degrading the quality of the communication with UDP or generating DUPACK with TCP which leads to poor performance [14]. The solution proposed in this paper, in addition to optimize the LMS channel capacity, must minimize the jitter by prioritizing the packets that can be directly decoded without any HARQ retransmissions. Scheduling mechanisms studied are presented in Section III.

HARQ also exists on data transmission for mobile phones such as LTE or future 5G networks [15], where transmission delays are lower. In a satellite communication context, long delay links make difficult to transpose terrestrial solutions [16] aiming at optimizing transmissions with HARQ.

C. Transport protocol and traffic

We use UDP in our simulations to carry VoIP traffic. We measure the most useful metrics obtained in different scenarios, which are the throughput, as well as the jitter,

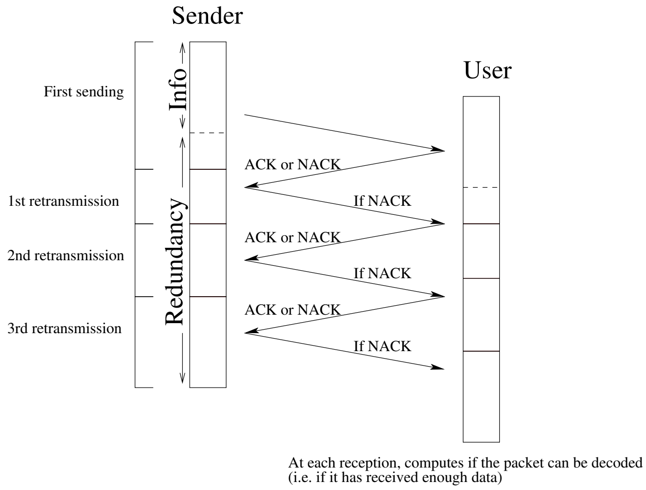

Fig. 2: Description of type II HARQ

Algorithm	Traffic	Metric	Computation complexity
PF [20]	Best-effort	Throughput	Very low
M-LWDF [6]	Real-time	Throughput and time in buffer	Low
EXP-PF [21]	Real-time and BE	Throughput and time in buffer	Low
UBMT [7]	All	Throughput and time in buffer	Medium
BBS [8]	Depends on metric	Throughput or time in buffer	High
BPS	Depends on metric	Throughput or time in buffer	High

TABLE I: Comparison of different schedulers

the delay, the losses, and the performance of HARQ, i.e. the average number of retransmission needed to decode a packet.

Each flow is being sent at a rate of 64 kb/s, following a Pareto distribution with burst time of 500 ms and idle time of 50 ms. The packets' size is 210 Bytes. This configuration allows to simulate VoIP traffic [17]. However, our version of HARQ uses packets with a size of 1115 bytes. Thus every VoIP packet entering the HARQ module is padded with zeros to get the correct size. The maximum load of useful coded bits on the LMS link is 9.4 Mb/s.

This kind of traffic is very sensitive to the latency and the jitter. Following ITU-T G114 [18], the maximum acceptable delay for toll quality satellite links is 400 ms (note this value is a maximum, a fair compromise is to be around 200 ms). However, a high jitter cannot be tolerated and must be minimized [18]. Similarly, losses cannot be higher than 3 % without decreasing the transmission quality and the user satisfaction. So the challenge is to find a solution to minimize the losses and the jitter, while allowing to maximize the throughput for each user.

To characterize the transmission quality, we compute the Mean Opinion Score (MOS), giving information on the user satisfaction. Basically, the MOS is computed as follows [19]:

$$R = R_0 - I_s - I_d - I_e + A \quad (1)$$

where:

- R_0 is the basic SNR;
- I_s is the simultaneous impairment factor;
- I_d is the delay impairment factor;
- I_e is the equipment impairment factor;
- A is the advantage factor. It is linked to users who can accept a lower quality taking into account the context of the transmission. With the use of satellite communications, this factor is set at 20 [19].

Then the MOS is computed from the R-factor following (2):

$$MOS = 1 + 0.035R + 7.10^{-6}R(R-60)(100-R) \quad (2)$$

III. Proposition of Schedulers

Different kind of schedulers have been introduced to optimize channel capacity and transmission delay in several communication contexts. They can use different metrics as the throughput of the link or the waiting time in the queue to schedule packets transmission. Some studies have already compared main existing algorithms [9], [10] in a satellite context. The main results are summarized in Table I. This table classifies these mechanisms as a function of the metric they seek to optimize. Among them, we choose Proportional Fairness (PF) against M-LWDF and EXP-PF (although defined for real-time transmission) or generic algorithm such as BBS and BPS for several reasons:

- Some scheduling mechanisms such as M-LWDF or EXP-PF need to be tuned to perform efficiently (sometimes involving complex parameters). Particularly, we tested M-LWDF and failed to achieve good performance;
- PF is a well-known and efficient scheduling system for fairness, simple to implement and computational efficient;
- We measured the gain that can be obtained on the LMS channel in a first batch of analysis and observed that optimizing the LMS channel can lead to a throughput gain up to 60 % compared to no scheduling policy. We show in this paper that PF performs results close to this optimum, thus no other scheduler can bring a real improvement compared to PF;
- We demonstrate in the following that adding a simple queue management scheme to PF allows to efficiently handle real-time traffic.

Proportional Fairness, which uses the throughput of the channel at the moment of transmission to send the packet having the best conditions while minimizing the channel usage, is a good candidate to optimize the LMS channel capacity. Packets entering the scheduler are stored in different queues based on the user's destination. There are as many queues as there are flows using the link. Each sending round, the scheduler selects a packet to send for all queues, following (3):

$$f_i(r) = \frac{r_i(t)}{\overline{r_i}(t)} \tag{3}$$

where $r_i(t)$ is the throughput of the channel at time t and $\overline{r_i}(t)$ is the smoothed throughput computed using exponential moving average. All the flows having an independent LMS channel evolution, we can ensure that the throughputs are all independent over time, and thus all different. The flow i^* elected to be transmitting is the one having the biggest $f_i(r)$ value. This value is computed as follows:

$$i^* = argmax(f_i(r)) \tag{4}$$

The channels being independent, the flow with the best value changes over time, allowing all the flows to be sent during the simulation. Furthermore, the division by $\overline{r_i}(t)$ implies that a flow cannot be sent during a too long time even if its channel quality is still good, to ensure short-term fairness between the flows and prevent large latency and jitter variations to other flows.

In real systems, the estimation of the LMS channel quality is obviously not instantaneous. The scheduler should get this estimation within a small delay. We neglect to consider this additional and fixed delay in our simulations. Indeed, this delay cannot be higher than the transmission time over the LMS link ($\approx 10\,ms$). So we can safely consider that the channel should not greatly evolve within this short time period.

We study PF performance in Section IV, then we propose in Section VI an improvement of PF to take into account the transmission delay and make it suitable for VoIP traffic while optimizing the LMS channel capacity.

We compute each queue size using the well-known formula given in [22]. The total buffer capacity is $B = \overline{RTT} * C/\sqrt{n}$, which is a more accurate extension of the common rule $B = \overline{RTT} * C$, and where $\overline{RTT}$ is the average Round-Trip Time of the flows, C is the bandwidth of the LMS link, and n is the number of flows sharing this link. Thus with PF, each queue has a size of B/n in order to have a total capacity of B. The number of users n can easily be computed by the scheduler by identifying the number of destinations.

We compare in this paper PF scheduler performance with other schedulers and queuing policies:

- Round Robin (RR): this scheduler does not take into account the quality of the channel when a packet is dequeued, and thus is not optimizing it. Each queue is served alternatively sending only one packet, then letting the other flows being transmitted;
- DropTail Small buffer (DT_S): only one queue with FIFO policy, the size of the queue is the same than in one queue of PF or RR, i.e. B/n. Thus the global storage capacity is different from PF or RR;
- DropTail High buffer (DT_H): only one queue with FIFO policy, the size of the queue is B, meaning the same total capacity than PF or RR.

IV. Results

We simulated different loads by varying the number of simultaneous flows. The number of flows is ranging from 25, where the global throughput is low and the buffers are never full, to 200 where the throughput entering the LMS channel is higher than its capacity, resulting in several tail drops.

We first observe in Figure 3 that when the system has to manage several flows, PF achieves a better throughput than the other policies. This is due to the better performance of HARQ, a packet being sent only when its channel has a low attenuation, as seen in Figure 4. Thus, more packets are decoded at the first HARQ transmission, leading to less retransmissions and a better LMS channel capacity usage.

When the system is not at saturation, we observe that DT policies allow more often to decode packets at first try than RR or PF. However, the number of packets decoded with 1, 2 or 3 retransmissions is lower with DT, and finally the total number of packets decoded is lower as we can see in Table II. The higher number of retransmissions with PF and RR does not negatively impact on the transmission quality because the LMS channel is not saturated, and can handle more retransmissions.

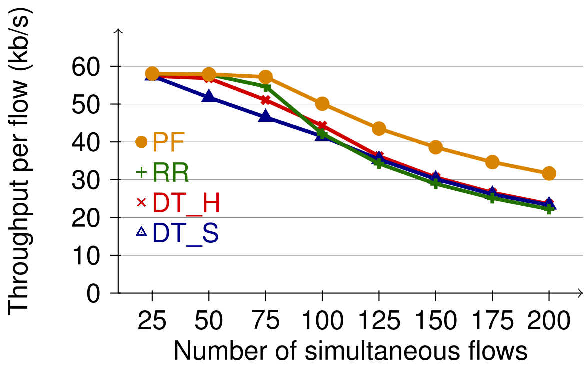

Fig. 3: Throughput obtained with different policies

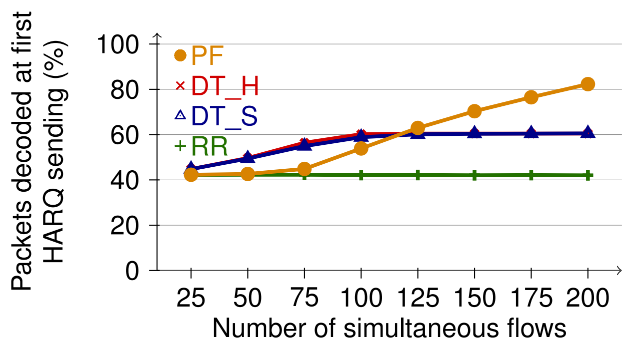

Fig. 4: Percentage of packets decoded at first HARQ sending with different policies

Policy	Number of retransmissions (%)				
	0	1	2	3	Drop
DT_S	50.11	23.16	14.65	12.07	1.21
DT_H	51.87	22.93	13.97	11.23	1.10
RR	42.36	24.28	24.71	8.65	0.61
PF	44.89	26.26	22.57	6.27	0.39

TABLE II: HARQ performance comparison with 75 users

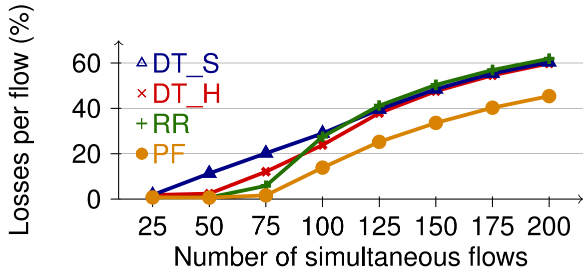

Fig. 5: Losses obtained with different policies

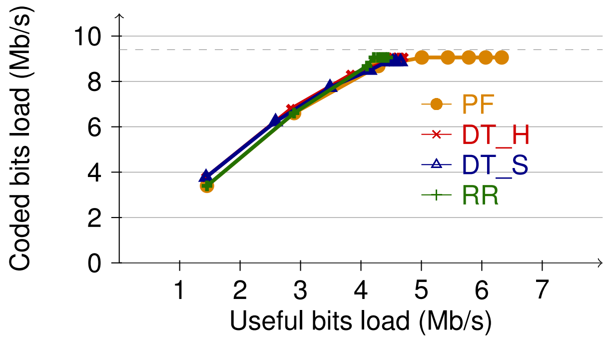

Fig. 6: LMS channel spectral efficiency

As a consequence, and as we can see in Figure 5, PF experiences less losses than other policies. As each packet uses less capacity of the LMS link, the scheduler can send more packets on this link decreasing the queue backlog. This allows more packets to be transmitted and to reduce drop due to buffer overflow.

Figure 6 has been generated by plotting the coded bits load of the LMS channel, depending on the useful bits load. Each point of a curve represents a different number of parallel flows. It shows that PF performs a better usage of the LMS channel than the other policies. Thus, for a same number of coded bits, PF can transmit more useful bits (meaning less redundancy bits) and the LMS channel is optimized with this scheduling policy.

Concerning the other metrics, we observe in Figure 7 that packets have a very long transmission time when using DT_H and RR. Packets sent with PF need more time than DT_S, but in acceptable ranges that are taking into account the context of satellite communications.

Finally the PF scheduling policy drastically increases the jitter. This increase is due to packets staying in the buffer as long as they need to have a good channel. This duration severely varies and is unpredictable, resulting in high jitter. As a matter of fact, high jitter strongly penalizes the quality of experience in particular for VoIP users [19], and needs to be lowered if we want to use PF in our context.

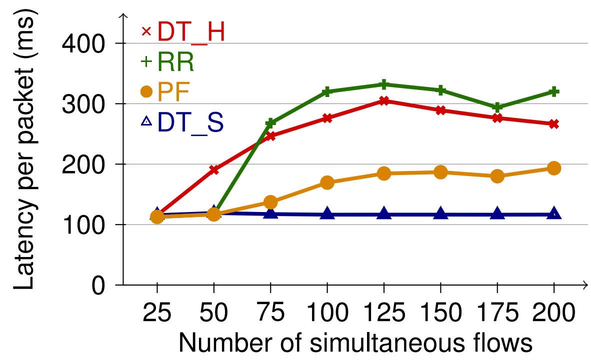

Fig. 7: Latency obtained with different policies

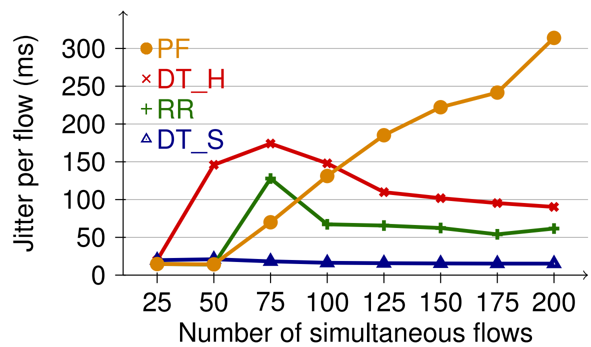

Fig. 8: Jitter obtained with different policies

V. Analysis

We drive our analysis by comparing each policy in terms of queue size, latency, and considering the resulting performance. We also compute the Mean Opinion Score as a QoE metric to assess user satisfaction.

Before comparing performance of each policy, we observed that the maximum capacity of the LMS channel, which is the bottleneck of our network, is reached for a number of users between 75 and 100. For higher number of users, the number of packets dropped, either by HARQ or queue overflow, both decreases communication performance and user satisfaction. Following these experiments, we estimated the saturation point at 100 users.

We can first observe that some policies provide very bad performance: RR and DT_H. For them, the high buffer capacity increases the latency. Thus, when the queues are full, the packet waiting time is B times the transmission time of a packet, where B is the total buffer capacity. This latency cannot be lowered. In addition to the fact that these policies do not optimize the LMS channel capacity by reducing the number of HARQ retransmissions, the global performance of

RR and DT_H makes them bad candidates in our scenario, and can already be discarded.

At first sight, DT_S may appear a good candidate due to its low jitter and latency, but the high error rate, even when there are a low number of flows, highly penalizes its performance. This high number of losses is caused by the low buffer capacity. This low capacity prevents a lot of users to use the scheduler simultaneously: with 75 users, we observe already more than 20 % of losses, implying that the number of simultaneous users has to be lower than 75. On the other hand, for the same number of users, PF loses only 3 % of the packets, which remains acceptable. Moreover, DT_S does not implement a scheduling policy taking into account the channel quality, limiting performance of HARQ, as presented in Table II.

We can see in Figure 10 that the MOS is higher for PF when a low number of users are transmitting compared to DT_S, which is penalized by the high error rate. PF achieves a good score up to the LMS channel saturation, with a MOS value higher than 3. We also have to keep in mind that in the context of satellite communications, the high delay always negatively impacts on the MOS, compared to terrestrial connection.

Unlike DT_S, PF is considerably improving the throughput but at a price of a very high jitter. Because the packets are sent on the LMS channel only when the attenuation is low, the packets may wait a long time before getting good conditions. This waiting time varies significantly, cannot be predicted, and is the main cause of this high jitter, which is highly decreasing the communication quality and the user satisfaction. This high jitter needs to be lowered if we want to use PF as a scheduling mechanism. We present in Section VI a solution answering this problem.

VI. Controlling the queuing delay to improve VoIP performance with CoDeS

We have seen in Section V that PF optimizes the LMS channel capacity, but at a price of a high delay and jitter, which decreases the user QoE. To take into account the sojourn time of the packets in the queues, we propose to add a queue management policy to PF. We call this new mechanism Controlled Delay Scheduler (CoDeS). Basically, CoDeS sets a timeout threshold value beyond which packets are dropped. Indeed, in VoIP transmissions, packets have a temporal deadline beyond which they will be discarded and not played by the VoIP receiver. Keeping outdated packets in the queues would uselessly consume buffer capacity and then LMS channel capacity during the transmission to the ground receiver. Thus, the global aim of this scheduler is to decrease both latency and jitter by dropping packets thanks to this timeout threshold preventing buffer overflow.

Improvements brought out by CoDeS compared to PF are given in Figure 9, where the timeout value has been set to 100 ms. We can observe as expected a decrease of the latency and the jitter, without impacting the other metrics. The number of losses is approximately the same than with PF, with acceptable values up to 100 users. However the cause of the drops is different, as shown in Figure 11. With CoDeS, packets can be dropped due to timeout, freeing space in the buffer for new packets, that would have been dropped with PF (*i.e.* without queuing management) due to buffer overflow. The total amount of drops is finally the same with the two policies, but the mean sojourn time of the packets is lower with CoDeS.

Thus, CoDeS improves the transmission quality by lowering both delay and jitter, but does not improve LMS channel optimization compared to PF. Indeed, we only added to PF a queuing management, which has no effect on the LMS link scheduling. The value of 100 users is still the limit in our simulations beyond which any scheduling policy looses too many packets to achieve a good user QoE.

We tested CoDeS with different queue timeout values to find the best configuration. The lower the value is, the lower jitter and latency are, without impacting the other metrics, up to a minimal limit reached with a timeout around 50 ms. In Figure 12, compared to a timeout value of 100 ms, we observe as we could expect that the jitter is higher with a timeout of 200 ms. On the other hand, with lower values such as 50 ms, the number of losses slightly tends to increase, as seen in Figure 13. This increase is due to the packet drops occurring too early, totally emptying the buffers instead of keeping them full. Thus, a correct value of timeout should be set between 50 and 100 ms, to have the best compromise between jitter and losses.

In Figure 14, we can see an increase of the MOS with CoDeS, with the timeout value of 100 ms. For the same number of flows, CoDeS slightly increases the MOS compared to PF, when the system becomes saturated. The MOS values with CoDeS are also higher than DT_S, due to the low number of losses, latency and jitter. Thus, by combining scheduling policy on the LMS channel and queue management, CoDeS achieves a good QoE, while optimizing the LMS channel capacity.

VII. Conclusion

Measurements show that the joint use of queue management and scheduling policy may not guarantee performance improvements. That being said, an adequate selection of algorithms can result in higher QoE for the VoIP users.

When it comes to congestion-friendly transport protocols, other solutions mixing scheduling policies and queue management may be relevant and solutions such as FQ-CoDel [23] would be an interesting algorithm to compare CoDeS with. Both FQ-CoDel and CoDeS have been designed with dedicated goals in mind: while FQ-CoDel focuses on prioritizing new flows without having too much impact on the old flows, CoDeS has been designed to optimize the usage of a scarce radio resource to allow more users in the system and provide a better QoE.

In a future work, we plan to assess the performance of CoDeS when some traffic is carried out by TCP. Indeed, TCP-based traffic may be impacted by our solutions, since it may induce (1) a highly variable delay and/or losses but (2) a

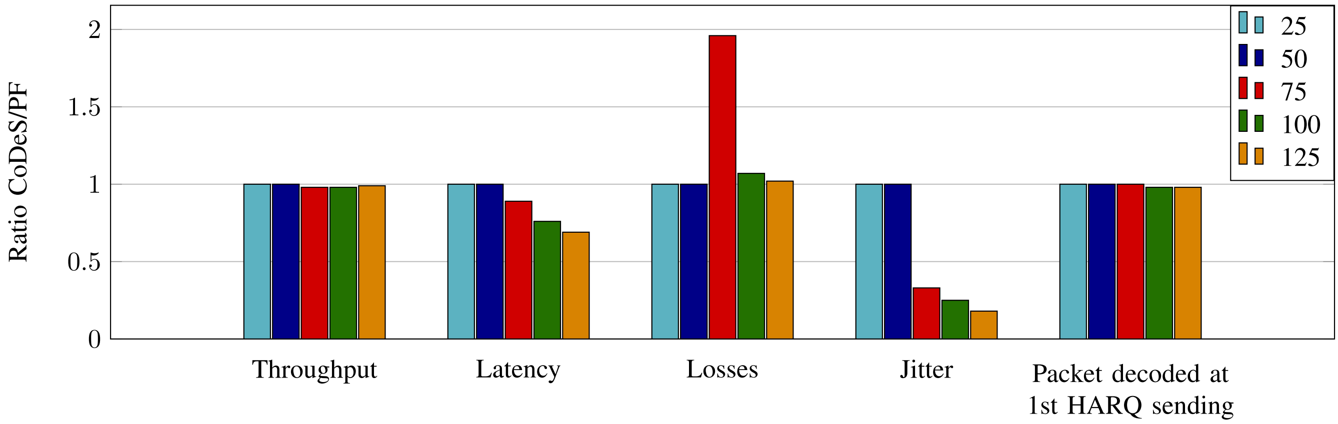

Fig. 9: Comparison of the ratio of performance between CoDeS and PF, with a timeout of 100 ms

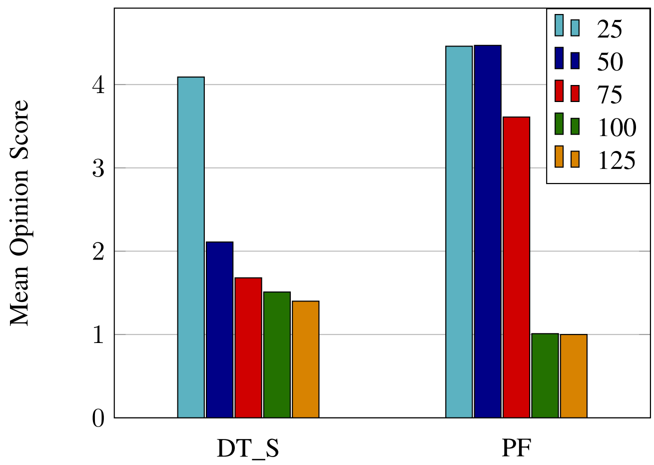

Fig. 10: Mean Opinion Score of DT_S and PF and as a function of the number of flows

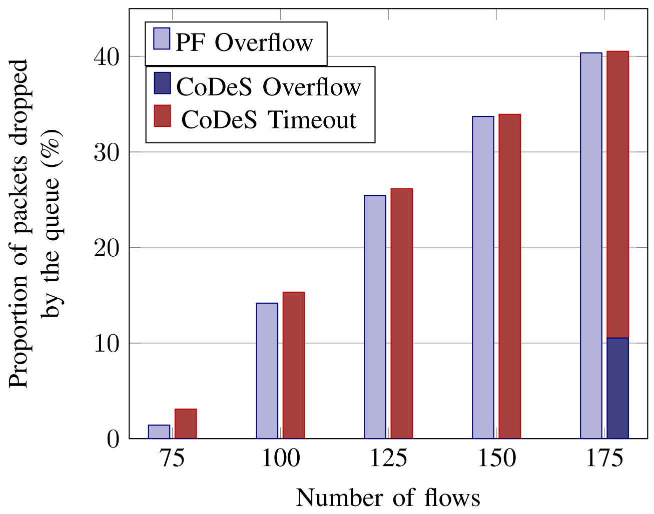

Fig. 11: Comparison of the cause of drops with PF and CoDeS, with a timeout of 100 ms

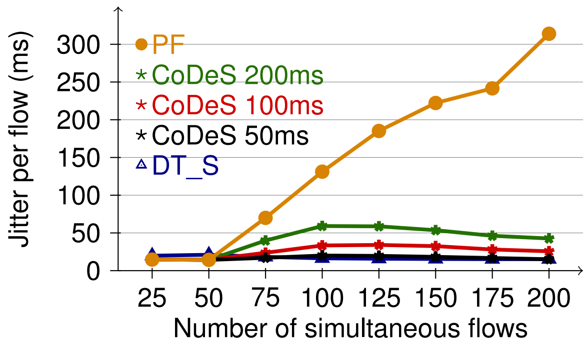

Fig. 12: Jitter obtained with different policies

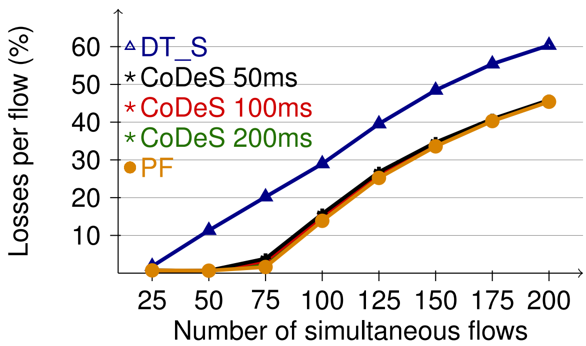

Fig. 13: Losses obtained with different policies

momentarily higher throughput. This would let us have fair comparisons with AQM-based solutions.

Acknowledgment

The authors would like to thank CNES and Thales Alenia Space for funding support.

References

[1] N. Blaunstein, Y. Cohen, and M. Hayakawa. Prediction of fading phenomena in land-satellite communication links. *Radio Science*, 45(6), 2010.

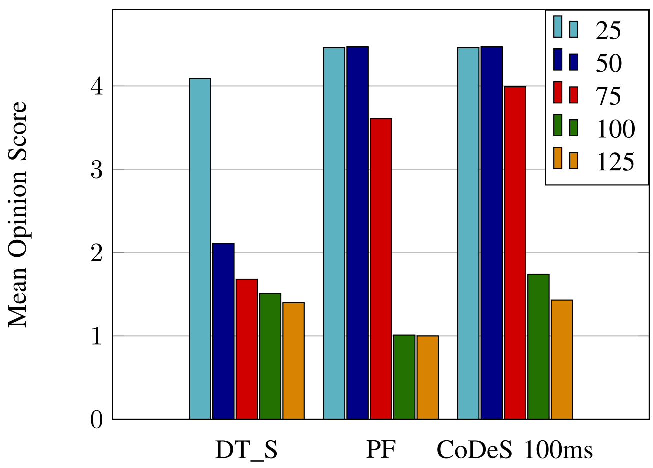

Fig. 14: Mean Opinion Score of DT_S, PF and CoDeS 100ms as a function of the number of flows

[2] F. Perez-Fontan, M. A. Vazquez-Castro, S. Buonomo, J. P. Poiares-Baptista, and B. Arbesser-Rastburg. S-band LMS propagation channel behaviour for different environments, degrees of shadowing and elevation angles. *Institute of Electrical and Electronics Engineers Transactions on Broadcasting*, 44(1):40–76, Mar 1998.

[3] A. Sastry. Performance of Hybrid Error Control Schemes of Satellite Channels. *Institute of Electrical and Electronics Engineers Transactions on Communications*, 23(7):689–694, Jul 1975.

[4] Tomaso De Cola, Harald Ernst, and Mario Marchese. *Application of Long Erasure Codes and ARQ Schemes for Achieving High Data Transfer Performance Over Long Delay Networks*, pages 643–656. Springer US, Boston, MA, 2008.

[5] Nicolas Kuhn, Emmanuel Lochin, Jérôme Lacan, Roksana Boreli, and Laurence Clarac. On the impact of link layer retransmission schemes on TCP over 4G satellite links. *International Journal of Satellite Communications and Networking*, vol. 33(1):pp. 19–42, 2015.

[6] Sanjay Shakkottai and Alexander L. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in HDR. In Jorge Moreira de Souza, Nelson L.S. da Fonseca, and Edmundo A. de Souza e Silva, editors, *Teletraffic Engineering in the Internet Era*, volume 4 of *Teletraffic Science and Engineering*, pages 793 – 804. Elsevier, 2001.

[7] G. Fairhurst, G. Giambene, S. Giannetti, C. P. Niebla, and A. Sali. Cross-layer study for resource management in DVB-S2 with mobile users. In *2009 International Workshop on Satellite and Space Communications*, pages 257–261, Sept 2009.

[8] E. Chaput, M. Verloop, A. L. Beylot, and C. Baudoin. Utility function based packet scheduling over DVB-S2. In *2013 IEEE International Conference on Communications (ICC)*, pages 4288–4292, June 2013.

[9] J. B. Dupé, E. Chaput, C. Baudoin, C. Bès, A. Deramecourt, and A. L. Beylot. Rule-based packet scheduling for dvb-s2 through generic stream encapsulation. In *2014 IEEE International Conference on Communications (ICC)*, pages 3576–3581, June 2014.

[10] J. B. Dupé, E. Chaput, C. Baudoin, C. Bès, A. Deramecourt, and A. L. Beylot. Optimized gse packet scheduling over dvb-s2. In *2014 IEEE Global Communications Conference*, pages 2856–2861, Dec 2014.

[11] G. Ash and D. McDysan. Generic Connection Admission Control (GCAC) Algorithm Specification for IP/MPLS Networks. RFC 6601 (Experimental), April 2012.

[12] Lloyd Wood. SaVi: satellite constellation visualization. In *First Annual Centre for Communication Systems Research Research Symposium*, 2011.

[13] Rami Ali Ahmad, Jérôme Lacan, Fabrice Arnal, Mathieu Gineste, and Laurence Clarac. Enhanced HARQ for Delay Tolerant Services in Mobile Satellite Communications. In *The Seventh International Conference on Advances in Satellite and Space Communications 2015*, pages pp. 1–6, Barcelona, ES, April 2015.

[14] Bastien Tauran, Emmanuel Lochin, Jérôme Lacan, Fabrice Arnal, Mathieu Gineste, Laurence Clarac, and Nicolas Kuhn. Making H-ARQ suitable for a mobile TCP receiver over LEO satellite constellations. 2017.

[15] M. G. Sarret, D. Catania, F. Frederiksen, A. F. Cattoni, G. Berardinelli, and P. Mogensen. Improving link robustness in 5G ultra-dense small cells by hybrid ARQ. In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pages 491–495, Aug 2014.

[16] S. R. Khosravirad, G. Berardinelli, K. I. Pedersen, and F. Frederiksen. Enhanced HARQ Design for 5G Wide Area Technology. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–5, May.

[17] Golam Sarwar, Roksana Boreli, and Emmanuel Lochin. On the quality of VoIP with DCCP for satellite communications. *International Journal of Satellite Communications and Networking*, vol. 30:pp. 163–180, 2012.

[18] ITU T-REC-G.114 Series G: Transmission systems and media, digital systems and networks, International telephone connections and circuits, General Recommendations on the transmission quality for an entire international telephone connection, one-way transmission time, May 2003.

[19] ITU T-REC-G.1020, Performance parameter definitions for quality of speech and other voiceband applications utilizing IP networks.

[20] F P Kelly, A K Maulloo, and D K H Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, Mar 1998.

[21] Sanjay Shakkottai and Alexander L Stolyar. Scheduling for multiple flows sharing a time-varying channel: The exponential rule. *Translations of the American Mathematical Society-Series 2*, 207:185–202, 2002.

[22] Guido Appenzeller, Isaac Keslassy, and Nick McKeown. Sizing Router Buffers. *SIGCOMM Comput. Commun. Rev.*, 34(4), August 2004.

[23] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm. RFC 8290 (Experimental), January 2018.