Re: BUFFERBLOAT FREE AND CONGESTION RESILIENT SATCOM NETWORKS (ARTES AT 6B.129)
https://esastar-publication-ext.sso.esa.int/ESATenderActions/details/75587

Dave Täht
CSO, LibreQoE

To whom it may concern: (Contact Person Florence Glandieres)

As the co-founder of the Bufferbloat Project, and contributor to multiple Fair Queuing (FQ) and Active Queue Management (AQM) methods and their associated test tools, I applaud this effort. I believe my company, LibreQoE, is in a unique position to be able to help implement this project.

Clarification Request: December 23, 2024

The request, as it stands, has a few points that I would like to address and clarify before submitting a proposal.

Aim of the Tender

"The objective of this activity is to design, implement[1] and test active queuing management and packet scheduling techniques at the user terminals and ground equipment, so that the SATCOM network continues to provide seamlessly an acceptable throughput and latency to users under any congestion situation."

A SATCOM system consists of at least the following components:

- Packet Scheduler
- Per Station Scheduler
- Mechanisms in place to avoid a DOS

By far the most complex thing is the "Per station" scheduler, which your request seemingly conflates with packet scheduling and AQM as if they were in a pure broadcast all-always connected' sort of environment. The underlying capabilities of the underlying media access control (MAC), and its "per station scheduler", are critically important in a good design.

Per Station Scheduler

"*Targeted Improvements:- Enabling a responsive and fair SATCOM network in any traffic and resources availability situation.- Keeping the additional queuing latency below 20ms (an order of a magnitude vs state of the art) per flow in any situation for both user terminals and gateways"*

[1] It is difficult to design new AQMs. We suggest testing and improving on existing, deployed AQM technology, such as CAKE, fq_codel, and/or fq_pie. We do not believe a non-fq'd AQM such as PIE or codel alone can meet your requirements.

The per station scheduler needs to provide an optimal solution to

- supplying time-slots and the right beams for transmission,
- estimating future demands for bandwidth, and
- supplying a bounded minimum service interval to each active terminal, ideally less than 10ms, and preferably under a ms.

Assuming those requirements hold true our existing FQ and AQM technologies can run on top of it. To some extent you have to optimize for cost/bit but providing service on sub 10ms intervals is mandatory. Ideally the underlying MAC<->AQM interface provides some clue as to pending backlogs to that scheduler. The minimum possible queuing latency is 2x the maximum time to service.

Regrettably for MEO orbits, 20ms of queuing is insufficient to initially scale to the proper delay/bandwidth product.

The terrestrial AQM “state of the art” is actually closer to 5ms, but it takes a while to stabilize there.

Discussion of Queueing Issues

“Description: In SATCOM networks, tail drop queuing is commonly used,however it is not the best option since only two out of the following three qualities can be achieved: (1) high throughput, (2) small and bounded latency and jitter range, and (3) no service outage.”

This description of the flaws of tail drop FIFO queuing is flawed. Service outages have nothing to do with it. FIFOs induce problems with tail loss timeouts, “TCP global synchronization”, and an inability to right-size themselves to the available bandwidth and RTT. All AQMs attempt to right-size themselves to the available bandwidth, and deal with TCP global synchronization. The “drop head” AQMs, codel[2] and especially cobalt (part of CAKE), essentially eliminate tail loss, and in the case of real-time communications throw out the most stale data.

Details of the Solution Set

“Quality of service could be improved, and all three elements achieved with Active Queue Management (AQM) adapted for radio resource management of satellite communication networks. This will particularly benefit voice and video conferencing over satellite.”

AQM controls queue length only, which in order to get decent initial throughput needs to have a queue of bdp/sqrt(flows)[3], that is gradually reduced to an optimal point. A queue is a shock absorber.

---

2 https://www.rfc-editor.org/rfc/rfc8289
3 http://yuba.stanford.edu/~sarslan/files/Updating_the_Theory_of_Buffer_Sizing.pdf

The “packet scheduling” of Flow Queueing[4], an improvement over Fair Queueing, particularly benefits voice and videoconferencing, by doing packet multiplexing that essentially bypasses capacity-seeking flows. The bandwidth requirements of these flows are minimal and sharing a queue with a capacity seeking flow increases jitter and latency.

Any form of FQ is better than any form of AQM for these protocols.

The combination of FQ and AQM, as in fq_pie, fq_codel, and CAKE[5], are the current state of the art.

“This activity will adapt the design and test active queuing management and packet scheduling algorithms, fulfilling the above mentioned three qualities. “

The third quality is bogus, as mentioned already.

“The algorithms shall ensure seamless acceptable throughput and latency among the users and achieve additional queuing latency below 20ms per flow in any situation. “

The best we have been able to accomplish in WiFi in ten years of trying has been a worst case of about 33ms. Part of the problem there is too-deep hardware buffering (in excess of 11ms), and interactions with the underlying MAC. The satellite radio encrypt/encode path in your design should be kept to the minimum number of packets possible to leave as much room for FQ and AQM techniques to manage packets in front of it. For example, a full size packet with a 10Mbit uplink consumes 1.3ms of airtime. 16 full-sized packets wedged in there blows your proposed 20ms latency budget entirely.

Orbital Mechanics

“They shall be agnostic of orbits.”

Regrettably there are three key parameters that an AQM must know in order to do its job properly: Bandwidth, RTT, and scheduled interruptions of service. These parameters are indeed “agnostic of orbits” but the current physical path latency and bandwidth need to be supplied to the AQM system on a regular basis, and especially in advance of a pending large change. In lieu of knowing the bandwidth directly, backpressure can be applied from the radio.

“The targeted solution will time bound the packet latency by dynamically adjusting the bandwidth distribution among all users, considering the type of traffic in order to minimize degradation of all user quality of service.”

This is a really broad requirement and is more targeted at the base station controller implementing the MAC (which has to manage hundreds or thousands of terminals), than the user terminal itself. It

[4] https://datatracker.ietf.org/doc/rfc8290/ The “flow queuing” mechanism has also been applied in CAKE and fq_pie. Benefits best described here: https://ieeexplore.ieee.org/document/8469111
[5] https://ieeexplore.ieee.org/document/8475045

is the role of the "station scheduler" to make this possible. "Flow Queuing" techniques are actually applicable to the station scheduling task[6], but occur in that layer. Care needs to be taken that the flows of packets sent via multiple beams do not arrive out of order.

Testing Discussion

"A test bed will be developed, and promising algorithms will be implemented and tested. Testing of the most promising one shall also be conducted over LEO and MEO satellite emulated channels."

Actually, emulating the behaviors of the satellite channels is the hard part. Evaluating the performance of the AQMs over that is fairly straightforward but congruent. I would suggest that the standardized testbeds include developing a variety of satellite MAC emulations, and that the most promising AQMs be tested and evolved concurrently.

A MEO orbit is ill-defined, being somewhere in the range of LEO to GEO[7], with a total physical latency ranging from 20 ms to over 600 ms. All the AQMs that we know of today are sensitive to the observed RTT and available bandwidth, and need at least an approximation of those in order to function properly. Some unpublished work of ours points to the advice in the codel RFC not scaling linearly to GEO, where a target setting of 30 and an interval of 300ms seemed robust and sufficient, but it was very sensitive to the underlying buffering and scheduling in the wireless encoder. Assuming 8000 km is what is meant by MEO here (150ms directly overhead) in this request, the rightest answer seems to be … testing.

Lastly, at bandwidths greater than 50Mbit, the bufferbloat starts to move to the WiFi. We highly recommend fq_codel'd WiFi[8]. The latest work is available in OpenWrt for many common chipsets.

SUMMARY

- A hard requirement for no more than 20 ms of queuing, at "MEO", will hurt throughput.
- Flow Queuing replacing the FIFO is the best option for good voip and videoconferencing support.
- The radio encrypt/encode path must be minimally buffered.
- A per station scheduler integrated with Active Queue Management is needed. I hope you have already decided on an underlying MAC or have that out for a much larger, separate bid.

If I may have the temerity and gumption to rewrite your request, something like the following would be more achievable.

[6] https://www.cs.kau.se/tohojo/airtime-fairness/
[7] https://earthobservatory.nasa.gov/features/OrbitsCatalog/page1.php
[8] https://api.starlink.com/public-files/StarlinkLatency.pdf We have had more than a bit of influence improving Starlink to date. It took three years to get their attention, but in under nine months of work all over their stack - especially including instrumentation - they did well. It helps to have a gamer CEO!

DRAFT REVISED TENDER OFFER

The objective of this activity is to design, implement and test active queuing management and packet scheduling techniques at the user terminals and ground equipment, so that the SATCOM network continues to seamlessly provide an acceptable throughput and latency to users under the most common congestion situations.

Targeted Improvements:

- Enabling a responsive and fair SATCOM network in any traffic and resources availability situation.
- Optimizing for the needs of voice and video-conferencing traffic while still providing acceptable throughput to data transfer applications.

Description: In SATCOM networks, tail drop queuing is commonly used, however it is not the best option. Quality of service could be improved, with Flow Queueing (FQ) and Active Queue Management (AQM) techniques adapted for radio resource management of satellite communication networks. This will particularly benefit voice and video conferencing over satellite.

This activity will adapt the design and test active queuing management and packet scheduling algorithms. The algorithms shall attempt seamless acceptable throughput and latency per user terminal. They shall be agnostic of orbits, but regular updates of available bandwidth and latency will be supplied by the station scheduler, with a per station service interval not exceeding 10ms.

This project’s AQM team will coordinate with another MAC design team to further time bound the packet latency by dynamically adjusting the bandwidth distribution among all terminals, considering the type of traffic, in order to minimize degradation of all user quality of service.

A test bed will be developed, and promising algorithms will be implemented and tested against the latency and bandwidth characteristics of LEO and 8000 km orbits.

…

Our punt here is that there must somewhere already be a group designing the MAC. We would also like very much to be involved in the design of such beyond mere packet scheduling.

CONCLUSION

We would propose leveraging LibreQoS & CAKE as a base to integrate with the "per station scheduler", in part because it is very fast and keeps great statistics. We would leverage Linux (and ultimately OpenWrt) to develop the client terminal, and router. CAKE is on-the-fly reconfigurable[9] with simple DOS protection and also has useful features like ECN support and ACK decimation that can be put in play. Diffserv is also supported and can be used for critically important packets, such as routing packets.

Presuming this might lead to a bid we also seek clarification on locations the work can be carried out at, either due to coverage of test satellites, locations they are willing to ship test ground stations to, or ESA organisational requirements.

What equipment will be required or made available?

In the hope that these comments are useful,

Dave Täht
Co-Founder, Bufferbloat.net
CSO, LibreQoS
dave.taht@gmail.com
support@libreqos.io (please cc)

[9] https://github.com/lynxthecat/CAKE-autorate uses this reconfigureability today for LTE and 5g networks.