AUTOMATIC TRAFFIC SMOOTHING FOR LOW LATENCY (ATSLL)

INVENTOR:

STEPHEN GLENNON

2/8/21

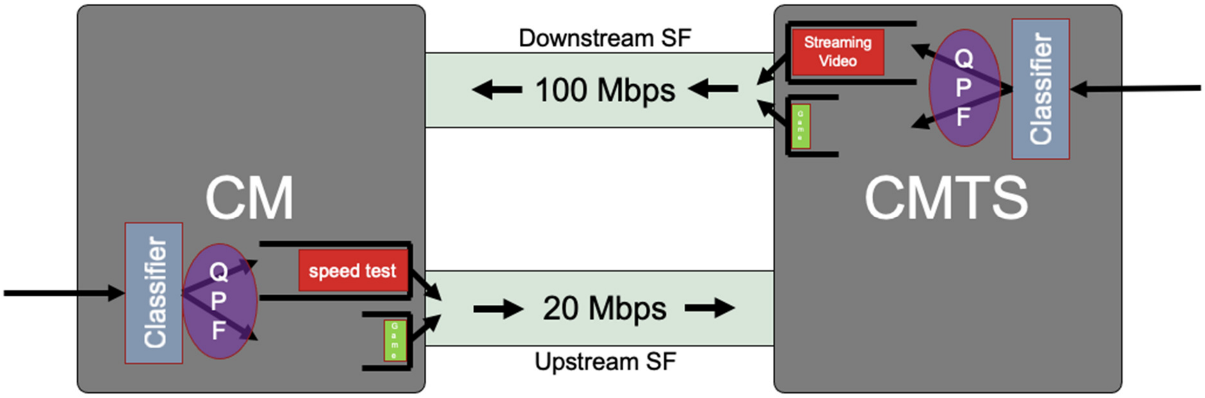
Automatic Traffic Smoothing for Low Latency

Inventors: Stephen G Glennon

Background of the invention:

One of the problems for low latency applications, with regards to Low Latency DOCSIS, is when an application is not capacity seeking (it has relatively fixed bandwidth, like 1-3 3Mbps for video), but the traffic is bursty (like frames of video).

At present, the simple, low effort approach of applying differentiated services marking on the video will get the traffic into the low latency queue, but the bursty nature of the video (with perhaps 5-10 packets arriving at the cable modem back to back) causes Low Latency DOCSIS to identify the traffic as queue-building/latency inducing. This can result in the packets being re-directed to the classic queue, and suffering from the variable latency that queue exposes.



QPF: Queue Protection Function

Figure 1 - Low Latency DOCSIS with Dual Queues and Queue Protection

Low Latency DOCSIS has dual queues feeding dual service flows for both upstream and downstream paths. It relies on two forms of traffic identification to direct latency-sensitive traffic into the low latency queue:

- Packets marked with Differentiated Services Code Points (DSCP), for un-responsive, non-queue-building traffic

- Packets marked with ECT-1 for responsive, L4S-compliant traffic.

For both these types of traffic the low latency queue is short, typically holding a maximum of 1ms of data. When the CMTS or cable modem detect that the low latency queue is filling, it invokes a queue protection function (QPF) that checks each flow to see how much it is contributing to the latency. If that goes above 1ms then the packets in that flow get re-directed to the classic queue.

For non-responsive traffic, this relies on the client application evenly spacing the packets to ensure that they do not cause contribution to the latency. For responsive traffic (L4S-compliant), there is a more complex feedback mechanism to speed up and slow down the packet rate to remain between 0.5 and 1ms of latency contribution.

The difficulty is that implementing L4S is complicated and intrusive. Participating as a non-responsive application is easy, simply setting the differentiated services field on the socket that is sending the traffic.

The even-spacing requirement may be easy (for example for voice, which sends a single packet every 1-3ms), but is more complicated for some applications. Video communication in particular tends to compress an entire frame of video, then send the completed frame at line rates (1Gbps). Resulting video frames might contain between 7,000 and 21,000 bytes, which correspond to between 5 and 14 1500-byte packets back to back. Typically for a 12Mbps upstream more than two packets back to back will cause the queue protection function to sanction (or redirect) packets, placing them in the classic queue.

Figure 2 shows the timing and size of video packets in FaceTime over a 5 second period. It shows that video packets are send as one chunk, and also shows that approximately every 2 seconds a much larger than usual frame is presented. This corresponds to an Intra-encoded reference frame which allows for random access within the video stream. Packets vary in size from around 7,000 bytes to around 21,000 bytes for reference frames. These cause data to be sent at instantaneous rates (with back to back packets) that cause queue protection to be activated, and the packets to be redirected to the classic queue. This results in out-of-order packet delivery, and much larger latency and latency variation than is intended.

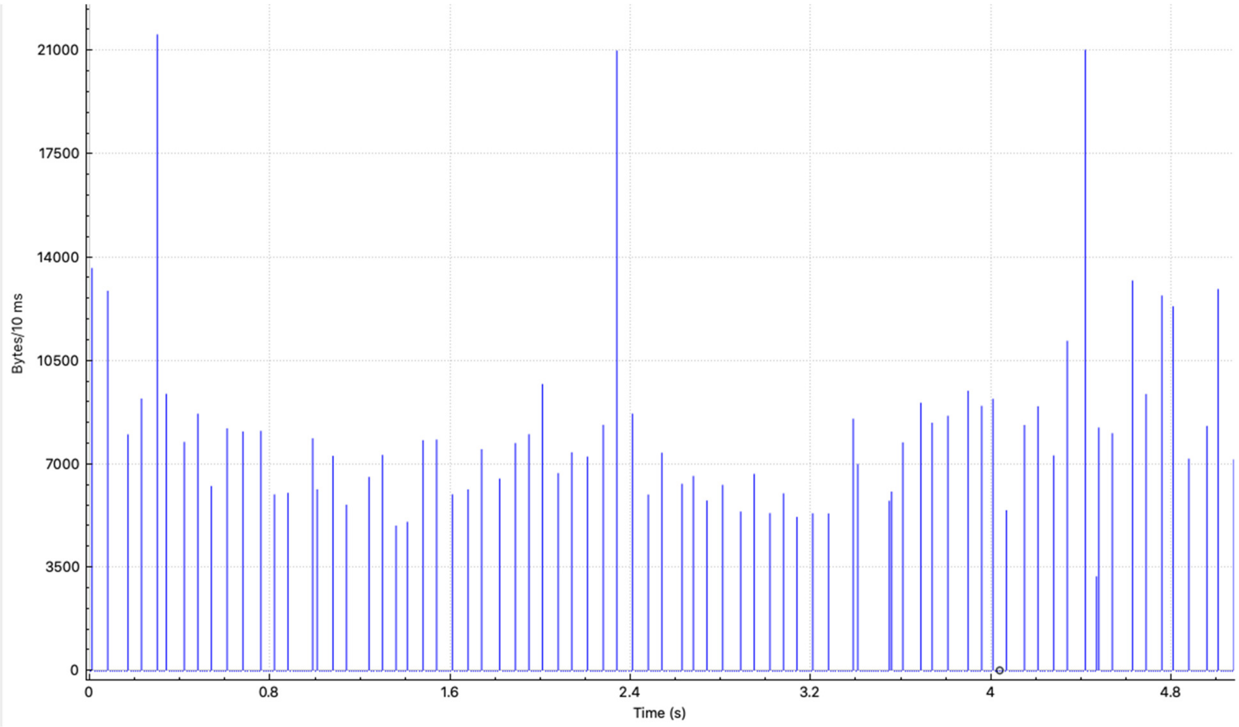


Figure 2 - Packet sizes and timing for video packets in FaceTime

The Invention

The invention here is to automatically identify this type of traffic flow and apply smoothing to the rate of packets before they hit the cable modem. Since the overall data rate might only be 1Mbps, far below the allowable 5Mbps default congestion rate of Low Latency DOCSIS, this video data should be able to be delivered via the low latency queue without any of the packets being redirected to the classic queue.

This might most naturally be implemented in the AP/Router that is before the cable modem on the video upstream (for example with a FaceTime or Zoom video stream). However, it might also be possible to implement this in the source device, for example the iPhone or the laptop that is sourcing the video for a zoom call.

The process anticipated is that an element is monitoring the creation of new network flows (as identified by source and destination IP address, and port number). When a new flow is created that has differentiated services marking to identify it as latency sensitive, an element in the router/AP keeps track of the average data rate of the stream. It then spaces out the arriving packets before forwarding them to the cable modem for transmission in a manner that ensures the packets do not trigger the queue protection function to re-direct the packet into the classic queue.

The simplest manner for the element to space out the packets is to delay the transmission of any packet sufficient that the average rate is never exceeded by two packets being sent too close together. While each packet is sent at line rate, it is assumed the element would leave enough gap that the combination of gap since last packet and line-rate transmission of the current packet in aggregate would not exceed the average rate my more than some margin such as 5%. This approach needs no knowledge of the specific parameters of the queue protection function, as it simply smooths out the packet transmission. The downside of this approach is that packets may be delayed more than is necessary to void triggering the queue protection function. Certain packets may be delayed more than necessary, which somewhat defeats the motivation of low latency.

A more sophisticated approach might involve knowledge of the parameters of the queue protection function, such as the congestion rate (e.g., 5.0Mbps) and the thresholds for detecting congestion, (e.g., 0.5ms and 1.0ms). Such an approach can perform a parallel calculation to the queue protection function and can delay packets only as much as is required to avoid triggering the queue protection function in the cable modem.

Now this approach works in the upstream direction for sessions initiating in the consumer home. It will avoid triggering queue protection in the cable modem upstream. It should also have the effect of avoiding triggering queue protection in the CMTS for the destination location, as the packets will have been spaced out. There is some potential that other network links cause some “bunching up” of these upstream packets before they arrive at a destination downstream link, but in general should maintain approximately the same spacing on the end-to-end journey. It may be appropriate to space the packets slightly more than the minimum required in the upstream to allow some margin for the timing of the packets to be changed on their journey.

It is also possible to implement this spacing in a network element in the downstream path prior to the CMTS downstream queue protection function.