Download Manager

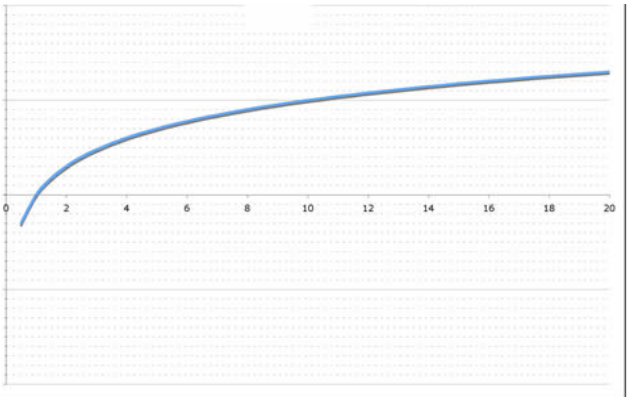
The download manager program uses an algorithm to meet the real time constraints for the play out of the received video segments. Without adjusting the quality and by this the amount of data that needs to be transmitted in relation to the number of currently buffered segments and the bit rate the video stalls during the play out. This is caused because the data segments are not available in time. When downloading the video at the highest quality, it can be derived from the measurement.csv report containing the MES values and playout.csv report that the video play out stalls, because the next segment was not received yet and the buffer is empty. Furthermore it is seen that the bit rate is highly fluctuating.

Bit rate adaption algorithm

As a rate adaption algorithm, an algorithm with a logarithmic characteristic was designed. The algorithm was designed to operate with highly fluctuating data rates. Therefore an integrative, as well as a differential function part was integrated to match the system requirements, resulting into an ID-modulator. The integrative element of the function integrates the measured bit rate over a certain number of measurements and calculates the average out of these. In the program the user can adjust the number of measurements taken as basis for the average value.

The differential part comes into consideration, when looking at the buffer. It is necessary to avoid fast fluctuation in relation to the quality of the video, while it is vital to meet real time constraints to avoid a stalling of the play out. Therefore the implemented algorithm uses a logarithmic characteristic to compute the representation of the next segment, according to the filling degree of the buffer. If the buffer is very low, while the average bit rate is much higher than the bit rate of the lowest representation the program only takes a certain percentage of the average bandwidth as guaranteed and decides to download a segment with a worse quality. This is done, because the buffer ran empty in the past due to a low transmission rate. So the program cannot be sure if the data rate will be very low again in the next moment. In addition a fast change in the quality of the video would be stronger recognized by the user. When the buffer runs full more of the average bit rate is trusted, so segments with higher quality are downloaded. To avoid breakdowns, when having a high quality stream because the buffer is full, the integration is done over several previous measurements, which decreases the effect of bit rate changes and the fluctuation of the bit rate. Furthermore the variation is smaller, so a higher percentage of the bit rate is trusted.

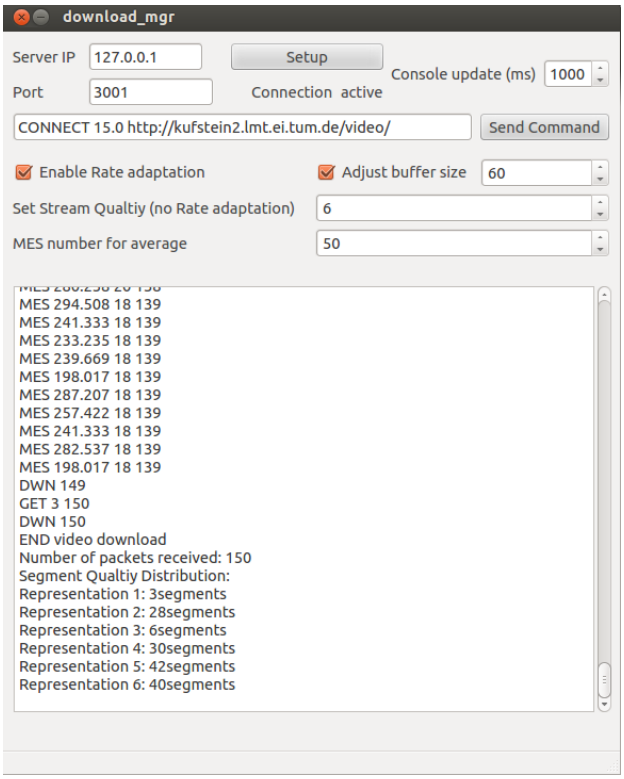
The consistency of the representation improves, when the size of the buffer is increased, because this increases the number of quantization steps, so the differential part of the function becomes less aggressive. In addition the quantity of measurements used for the computation of the average bit rate can be increased. This increases the consistency of the representation further, but may degrades the swing characteristic of the system for sudden changes, which last for a longer time. The result would be a shift of the operating point/area, which results from the equal distribution of integration and differentiation function parts.



To improve the code further and to avoid the differentiation effect of the function at start up, the buffer is preloaded by a linear function, which only relies on the average bit rate until the play out starts or the limit of the buffer is reached. The buffer limit for play out may be much smaller than the buffer used in the function. The buffer is only loaded linearly to avoid a transient oscillation of the system for small buffer sizes. In case the play out buffer size is picked very small, the system will swing in over a certain time. The system will be stable if the size of the system buffer is sufficient (min(size of buffer) = number of quality representations in size.

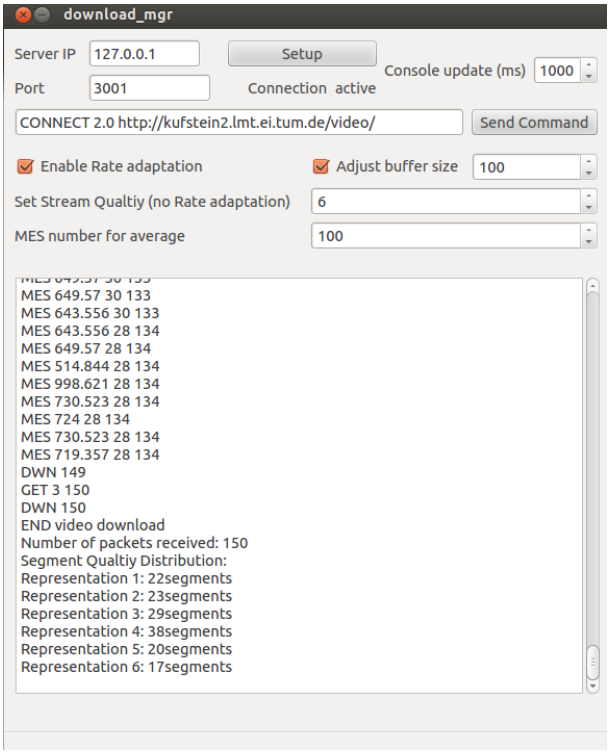
The Algorithm was developed with strong focus on transient changes in bitrates , that last for a longer time window, like the shift from 3G to Edge network link speed, when traveling inside a underground. However it is also well suited to cope with fast changing bitrates that show a purely heuristic behaviour.
To increase the performance of the system the average bit rate is only calculated, when a new packet needs to be requested from the server. Furthermore the update rate of the console can be adjusted to improve the system performance of very weak machines, but it should not be necessary for normal devices like smartphones and computers. The mathematical overhead of the algorithms is very small so it can be used on systems that lack performance. At start up the average is calculated on basis of fewer measurements until the defined number of measurement is available in the buffer. This results in a weaker effect of the integration value when more measurements are required than available at the play out start time. This effect will vanish, when the determined numbers of measurements were acquired.
The algorithm is working well after the transient oscillation of the system finished, providing a relatively consistent output representation to the user.

Buffer playout size: 15.0 sec;
Aspired maximum buffer size: 60 segments;
Number of MES measurements used to calculate average value: 50;



Record 1: PSNR average:38.58 min:29.85 max:inf

Buffer playout size: 2.0;
Aspired maximum buffer size: 100;
Number of measurements used to calculate average value: 100;



Record 2: PSNR average:36.66 min:29.86 max:inf

Other setup measurements:
PSNR average:36.49 min:30.44 max:inf
PSNR average:37.83 min:29.80 max:inf
PSNR average:36.52 min:29.34 max:inf

The PSNR of the video download at highest quality, without rate adaptation is:
PSNR average:44.61 min:41.32 max:inf

The peak signal to noise ratio of the rate adaption algorithm is 8 PSNR worse than the download of only the segments at with best representation.

When downloading a stream, all setup values, except the play out buffer initial size, can be changed dynamically. This was implemented to give the user more possibilities to analyse the system bitrate and to optimize the Integrator Differentiator (ID) actuator for the desired Network. The behaviour of the transmission is displayed in the Debug window, by displaying the measurements. If the buffer size is not adjusted, the size of the buffer will be equal to the number of segments necessary to store all segments until play out time.
After receiving all segments a histogram is given out at the Debug window, determining how many segments were downloaded with which quality representation.

No pipelining is applied when downloading the segments. The download is occurring totally sequential. This means, that the next packet is requested only after the previous was received by the dash_deamon.