Summary Review Documentation for

“Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard”
Authors: T. Huang, N. Handigol, B. Heller, N. McKeown, R. Johari

Reviewer #1

Summary: This paper presents an in-depth analysis to understand challenges in good bitrate adaptation in HTTP based video streaming in the presence of competing traffic. They identify several interesting artifacts of overlaying video on top of HTTP on top of TCP; noting that using the bandwidth estimate to pick a bitrate is likely to be broken because the bandwidth estimate is biased.

Strengths: This is timely problem given the growth in video, and that other measurements have pointed out some deficiencies in video players without providing any deep insights. This paper provides a more in-depth analysis and interesting insights.

Weaknesses: Not sure I buy into the recommendations as being valid or robust – feels that the authors are trying to be too "sensationalist" here.

Comments to authors: At a high-level, I found myself in agreement with many of the insights and observations made in the paper and saw many nicely written quotes that crisply summarized the crux of the problem and intuition here. That said, it did seem to be a somewhat belabored and repetitive effort to make these points – this would have been a no-brainer for me as a short paper but the narrowness of the observations makes me a bit less excited in giving a higher rating.

Also, the recommendations you make seemed a bit out of place and somewhat counter to the whole idea of being a "good citizen" – e.g., be aggressive, use higher bitrate always, eliminate bandwidth estimation etc. Your results suggest a more nuanced result (e.g., what you are really saying is that use the buffer size and not the bandwidth) but as written in the conclusion it makes the recommendations sound more overselling than they need to be.

Reviewer #2

Summary: The authors presents measurements of the "downward spiral" effect in video clients where the streaming rate plummets in the presence of competing traffic. The paper explains why this effect appears and proposes a few possible mitigation strategies.

Strengths: The authors presents measurements of the "downward spiral" effect in video clients, in which the streaming rate plummets in the presence of competing traffic. This effect is crippling video experience and seems to appear in practice, thus trying to measure and explain it is a very good idea. The paper tries to nail the root cause of this effect and finds many clues on the way.

Weaknesses: The main explanation given for the downward spiral effect is a bit confused - a large part of the blame is placed on slow start, when it’s not clear that’s the case.

Comments to authors: This is an interesting paper. I am convinced this effect you measure exists, and I would like to understand why it happens; your paper is very close to understanding it, but I don’t think it’s quite there yet.

You mention in a few places that slow-start creates problems - I am not sure that is true. To avoid slow start you could of course disable the idle timer in the stack to avoid cwnd expiring, but I think you will find the effect will be worse - because the video flow will get tons of losses in the first round-trip time.

Out of the reasons you enumerate, my favorite explanation is a problem of convergence to a fair share because there is not enough time to converge. It would be great if you could measure (in simulations, or with real stacks, it should not make a big difference) what is the perceived throughput of one short flow downloading the equivalent number of bytes and competing with a long flow. You would vary the short flow’s size, the link parameters, the number of long flows, etc. I think this may give you a good handle on the problem - your figure 8 is a good starting point, but not conclusive enough.

I am not sure the following sentence is fully justified: "with a smaller segment, the video flow becomes even more susceptible to the loss induced by the competing flow" - loss rate should be the same for both flows; thus the number of losses should be proportional to the sending rate; thus the competing flow should see more loss - as it does, in your experiment fig.8. Are you suggesting there are some sort of phase effects where the short flow sees a higher loss rate? Perhaps the video flows sends a burst of packets in its initial window and that causes a series of packet losses at the bottleneck?

The options you detail in the beginning of section 6, that could be implemented at the server - why do you make the point they can’t they be implemented at the client? It would seem to me that any history based decision would fail to adapt when there is competition for the bandwidth.

Mitigation strategies: i) why not try to promote information from larger chunks, instead of just killing outliers? ii) being less conservative seems like bad solution - how aggressive should you be? iii) I think the only real solution from this section is to increase the
chunk size - but then the provider has a cost if the user skips ahead, etc. Would this extra cost matter?

I like the idea for a rate-less strategy mentioned in your conclusions.

**Reviewer #3**

**Summary:** This paper analysis pathologies in client-side video rate selection in the presence of competing bulk transfer traffic. The authors identify the underlying causes and present some recommendations to overcome the pathologies.

**Strengths:** This problem is important, and the analysis is thorough across the small space of providers chosen. The problems with video rate selection and underlying causes are explained very clearly through carefully designed experiments.

**Weaknesses:** It’s not clear how big the delta from prior work is. [3] has examined the video rate selection problem, although the focus there is on competing video flows. [15, 4] have also identified the problems. The measurements are limited to 3 providers. Also, key recommendations are not evaluated.

**Comments to authors:** While the paper tackles an important problem that has received quite a bit of attention in recent times, I wonder how (1) novel and (2) relevant the work is. To me, the delta wrt [3] is not totally clear. The authors say that [3] considers interaction between competing video flows. True, but the observations they make regarding the fact that pathological rate selection happens appear to be roughly the same. This paper delves deeper into showing that the root cause lies in how estimation of bandwidth above TCP interacts with HTTP, chunk size and video playback buffer, but to me, [3] steals some of the thunder away to the extent that I felt large parts of section 4, and to some extent 5, were redundant wrt [3]. [3] also compares different players which you don’t. At the very least, the paper should provide a much more detailed comparison with [3].

I wish the paper spend some more time in convincing me the problem is real and that a streaming-centered solution is necessary. A clearer description of scenario being considered would be nice. E.g., I was not sure if you were focusing on home networks. Perhaps your were, since you were consider very low levels of statistical multiplexing. (If you were to consider wide-area links where there are more flows contending, then I would be more curious to see the interaction between large scale statistical multiplexing and the performance of video flows.) In particular, what if you employed traffic prioritization in the home and demoted bulk transfers to lower priority? You would not require the changes you propose in section 6, then?

Related to this, one thing that would have made your work more compelling to me is a better evaluation of some of your recommendations, in particular, the suggestion of not estimating available bandwidth at all. I like this suggestion quite a bit: you are arguing for considering just the playback buffer and to “increase the rate when the buffer is high, and decrease when the buffer is low”. This makes sense because low buffer means client is playing back at a rate faster than network is delivering data → available bandwidth is low → pick lower rate. I wish this were more of a focus in the paper. A key issue that arises here is how to monitor buffer size and when to react. This will crucially determine the overall effectiveness of this suggestion.

I’m also not sure how generic the results are. Do the three providers you sample represent the cross section of available techniques for stream rate selection? Do they cover different player types that are in use today (as studied in [3])?

**Reviewer #4**

**Summary:** The paper looks into commercial video streaming services, and shows that many of these services select poor rates in presence of a competing background flow. The motivation for the work is weak, and many of the findings are not particularly novel.

**Strengths:** Use of commercial servers in implementation based evaluation

**Weaknesses:** The motivation for the work is weak, and many of the findings are not particularly novel.

**Comments to authors:** The problem of correct rate selection for video streaming is important, however the paper focuses on some very specific scenarios, that of “one” competing “video” based background flow. How realistic is this situation? Some motivation is needed in this context to strengthen the paper. Normally, when some one is viewing a video, why do they want to view another at the same time? The background traffic needs to be more realistically modeled. If there are other situations whether the above scenario works, that needs to be explicitly mentioned. In general, I would have also liked to see if similar problems arise if the background traffic was browsing or downloads from different servers.

There is considerable literature on video encoding, sophisticated rate selection etc. If these commercial services employed any of those techniques, would the problem disappear? Some insight into this also needs to be provided. Otherwise the problem is moot.

How TCP reacts in presence of competing flows is rather well known. So, some of the explanations for the behavior didn’t come across as particularly novel. The interesting aspect specific to the video streaming was the on-off behavior. However, it looks like this was also noted in some related work. So, in terms of novelty, the paper doesn’t come out as particularly strong.

**Reviewer #5**

**Summary:** The authors conducted a study of 3 popular video streaming services (Hulu, Netflix and Vudu). In these services the client carries out bandwidth estimation and uses this to pick the rate for video streaming. They showed that all 3 services underestimate the available bandwidth when competing flows are present and thus pick streaming rates far below what they could. Through a detailed and careful analysis, the authors show that this occurs because of interactions between the video playback buffer, HTTP and TCP.

**Strengths:** Overall, this paper is extremely well written, presents a very careful and systematic study, and reports on an important
and timely topic. This is a nice paper on root cause analysis of a problem that occurs due to lack of a cross-layer design approach and poor bandwidth estimation methods in practice.

**Weaknesses:** I suppose that one could argue that this is an exercise in reverse-engineering.

**Comments to authors:** This paper is really well written. The sequence of plots 4, 5, 6 and 8 are really helpful to illustrate what's happening. Nice job!

The authors identify the following issues in how the video playback rate is selected by the client. First, the client measures the instantaneous available bandwidth. If it measures this by the arrival rate of the last chunk, then it frequently underestimates by a long shot. Second, the client often goes into an ON-OFF behavior cycle (app layer) and during the OFF periods, the TCP congestion window times out. Also during the OFF period, the buffer fills and thus packet loss occurs. Third, the clients tend to pick very conservative playback rates given a bandwidth estimate. So overall the authors claim that the essential problem is that the client is measuring the available bandwidth ABOVE the TCP layer and is unaware that TCP itself is having trouble reaching its fair share of the bandwidth.

The study mainly focuses on an analysis and illustration of the problem for 3 real-world popular services. The authors also propose 3 fixes that are shown to work via a few examples. The fixes focus on (1) being less conservative; (2) using a more stable estimate of the bandwidth via averaging multiple samples; (3) using larger chunks of video download so that TCP can reach its fair share on its own.

Do you know or are you guessing that the client in Service A measures instantaneous bandwidth via the arrival rate of the last chunk? You show through half a dozen examples that the selected rate appears correlated with the perceived throughput (as you define it). Why just 6 examples? Why not run this over hundreds of samples from your data and report the frequency of this co-occurrence?

**Response from the Authors**

We would like to thank our reviewers for the thorough and thoughtful comments. We have fixed the text and incorporated all the minor and editorial comments. There are three main concerns raised by the reviewers and we will respond to each of them below.

First, the reviewers suggest a more precise explanation for the issue of why convergence to fair share does not happen. This phenomenon happens because of the following: Once the video clients playout buffer is full, the client enters a periodic ON-OFF sequence. We found that the OFF period is long enough to make TCP reset the cwnd, causing the video flow to ramp up again from slow start. During the OFF period, the competing TCP flow fills the router buffer, so the video flow sees high packet loss once it is back to the ON period. Worse still, the video could finish downloading its video chunk before its cwnd climbs up to its fair share and re-enters the OFF period. This process will repeat for every ON-OFF period and as a consequence, the video flow underestimates the available bandwidth. To help readers understand this phenomenon after seeing all the measurement results presented in Section 5.1 - 5.3, we add a summary paragraph at the end of Section 5.3.

Second, a number of reviewers were concerned about ramifications of the video flow being more aggressive. What we really mean is to make the video flow rely on TCP to get its fair share and be less conservative on selecting video rates at the HTTP layer; TCP prevents the single video flow from ever getting more than its fair share. We presented some preliminary results for a less conservative algorithm in Section 6.3. The results show being less conservative at the HTTP layer helps improve video quality while still maintaining reasonable playout buffer occupancy. To avoid the confusion, we changed the wording in the text and emphasized that we will still be a good citizen by having our algorithm operate on top of TCP.

Last, many reviewers were concerned about the novelty of this work and asked for an explanation of the difference between this work and related work [3]. In our work, the competing flow is a general HTTP file download, instead of another video streaming player. Thus, the scenario we consider is different from [3]. In [3], they focus on the unfairness problem between two video players. In our work, we focus on the unfairness between two TCP flows, of which one is a video streaming flow and the other is a general HTTP file download. [3] finds that ON-OFF behavior is the problem causing unfairness between two video players. In our work, we found that there are many other factors that can lead to undesirable behaviors in rate adaptation algorithms. In some cases, ON-OFF is just a trigger that leads the algorithm into a vicious cycle. In other cases, the algorithm is sensitive to TCP temporal behavior, resulting in rate fluctuation.