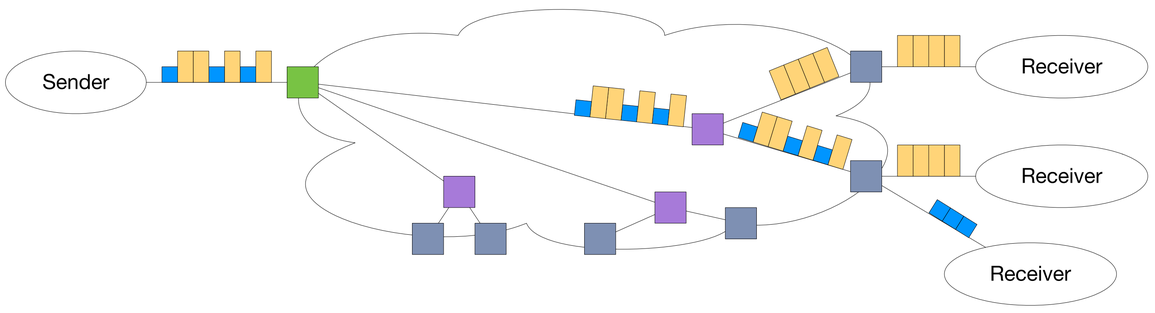
Consolidating Streams to Improve DASH Cache Utilisation

University of Glasgow
School of Computing Science

Stephen McQuistin and Colin Perkins
University of Glasgow, Scotland, UK

Introduction

A growing number of video services use MPEG Dynamic Adaptive Streaming over HTTP (DASH) to deliver content. These flows are delivered using a Content Delivery Network (CDN), with a number of caches on the path.



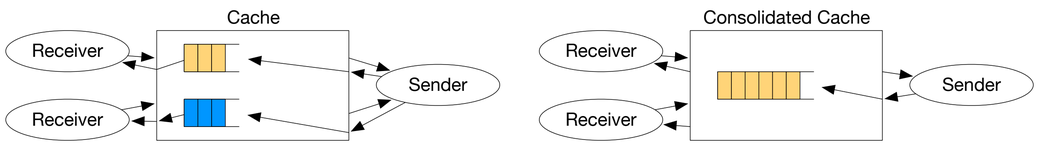
HTTP caches interact poorly with multiple flows of the same content: time and quality differences reduce cache hit-ratios, negatively impacting performance. Our techniques consolidate near-simultaneous flows based on time or quality, reducing the overall number of flows, and increasing the cache capacity for remaining flows. In large-scale streaming platforms, we estimate that there are a sufficient number of near-simultaneous flows for the same content for a non-trivial improvement in performance, and these improvements are only likely to increase as the use of DASH grows.

Stream Consolidation

Our consolidating cache identifies near-simultaneous flows for the same content at potentially different rates, and selects (based on the methods below) one flow that could serve the clients of the other flows. The remaining flows are then eliminated from the cache, freeing up space for the surviving flow. HTTP redirects are used to serve clients with chunks from the surviving flow.
To allow this, the cache intercepts and interprets [4] the media presentation description (MPD) for all DASH flows being requested. The MPD describes the URL structure of the content, and the different representations offered.

Rate-based

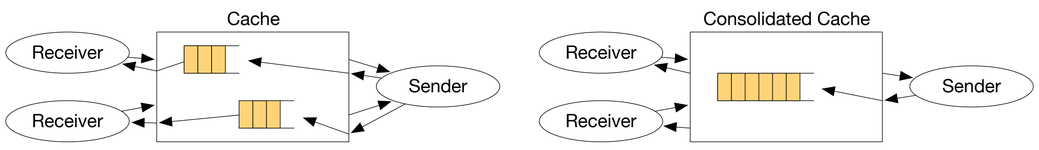
Simultaneous streams of the same content, but at different representations, can be consolidated. We select the best common representation available, based on the bandwidth capacity between the cache and each client, and the cache and the server.



Gouta et al. show that offering multiple bitrate representations decreases cache hit-ratios by 15% [3], indicating the performance improvements that this technique may yield.
CF-DASH [2] allows clients and caches to agree on a representation. This agreed representation is not enforced by the cache, allowing a given clients to provide a better quality of experience by requesting a higher representation than that agreed upon. This reduces the likelihood of deployment.

Time-based

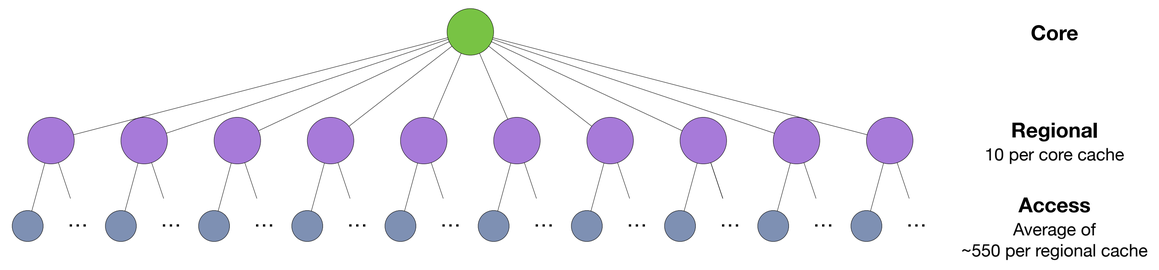
Near-simultaneous streams of the same content, at the same bitrate, but at different time offsets, can be consolidated.



Near-simultaneous is defined as a time offset that, if the flows are consolidated, would not significantly impact the user. To reduce the likelihood of clients skipping to offset the impact of the technique, we prefer to repeat content that a client has seen, rather than skip ahead. Therefore, the surviving flow is that which is requesting content from the earliest time instant.

Preliminary Analysis

We use the UK's BBC iPlayer platform as an example of a large-scale streaming service; in July 2015, it saw an average of 6.3 million requests for TV content per day. Of these, 483,000 (7%) were made at the peak time, with an average of 28,890 (0.6%) for the most requested content.



Assuming the cache hierarchy shown above, and using UK population estimates, we estimate that each access cache will see around five near-simultaneous streams. The benefits of our techniques ripple up the cache hierarchy: fewer representations being cached in the access caches leads to fewer representations at regional caches, and fewer representations at core caches. Core caches will see thousands of simultaneous streams, allowing our techniques to deliver a significant performance improvement.
Current iPlayer usage is dwarfed by traditional TV consumptions. As this mix shifts towards IP delivery, the techniques we propose will become increasingly useful. They will remain so for as long as live or scheduled TV is delivered by content providers.

Implementation Challenges

Degraded quality of experience Manipulating DASH flows in the way we propose will result in either in introducing skips or reducing the bitrate representation provided to users. This does not necessarily lead to a net reduction in quality of experience – our techniques allow for other metrics, such as start-up time, to be improved.

Widespread end-to-end encryption The techniques proposed would not work with DASH flows that have been encrypted end-to-end. This is a wider challenge for performance-enhancing middleboxes in general, but technical solutions exist [5] if the performance improvements prove worthwhile. Existing systems often only encrypt the payload, leaving the HTTP headers in the clear, and enabling caching.

Conclusion

Our techniques are designed to improve the performance of DASH applications by improving cache hit ratios. It remains for these techniques to be implemented and evaluated, including investigating interactions with other caching policies, such as prefetching algorithms. These techniques, and the performance improvements that they offer, will become increasingly beneficial as the use of DASH grows.

References

[1] iPlayer - Monthly Performance Pack, June and July 2015. BBC iStats, July 2015.

[2] Z. Aouini, M. T. Diallo, A. Gouta, A.-M. Kermarrec, and Y. Lelouedec. Improving Caching Efficiency and Quality of Experience with CF-Dash. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, NOSSDAV '14, pages 61:61–61:66, Singapore, 2014. ACM.

[3] A. Gouta, D. Hong, A.-M. Kermarrec, and Y. Lelouedec. HTTP Adaptive Streaming in Mobile Networks: Characteristics and Caching Opportunities. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*, pages 90–100, Aug 2013.

[4] D. H. Lee, C. Dovrolis, and A. C. Begen. Caching in HTTP adaptive streaming: Friend or foe? In *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Singapore, March 2014. ACM.

[5] D. Naylor, K. Schomp, M. Varvello, I. Leontiadis, J. Blackburn, D. R. López, K. Papagiannaki, P. Rodriguez Rodriguez, and P. Steenkiste. Multi-Context TLS (mcTLS): Enabling Secure In-Network Functionality in TLS. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 199–212, London, United Kingdom, 2015. ACM.