Pull-Patching: A Combination of Multicast and Adaptive Segmented HTTP Streaming

Espen Jacoben, Carsten Griwodz and Pål Halvorsen

[simula . research laboratory]

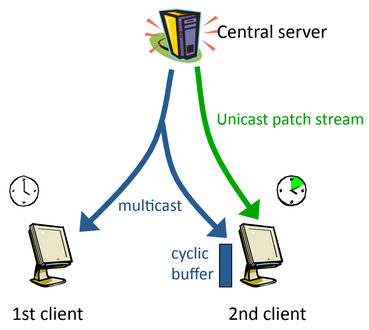
UNIVERSITY OF OSLO

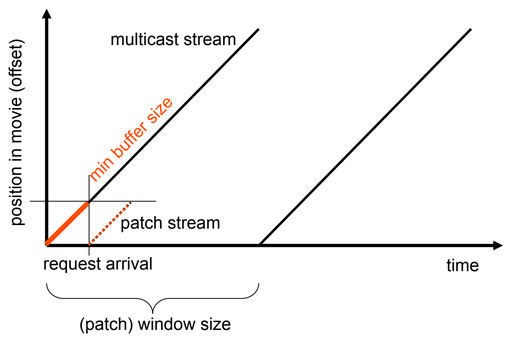
Multicast delivery for video streaming gains credibility with the introduction of commercial IPTV. We therefore revisit patching, a video-on-demand idea from the 1990s. We have built Pull-Patching, an approach that combines the patching ideas with adaptive segmented HTTP streaming, a unicast technique that is used by most commercial providers of large-scale, true video-on-demand in the Internet today. The prototype is tested in a combined Internet and lab environment where we show the influence of practical factors like packet loss, delay and limited resource availability, and identify several details that require further study.

Architecture

Patching

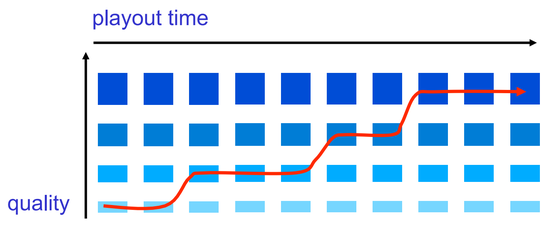
The server uses multicast to send a video periodically. Clients join the last started multicast stream upon arrival. They receive the missing first part of the video stream in an individual unicast (patch) stream. While streaming and playing out the patch stream, the multicast stream is buffered.

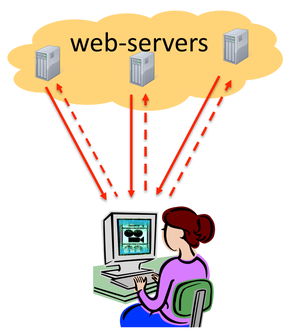




Segmented Adaptive HTTP Streaming

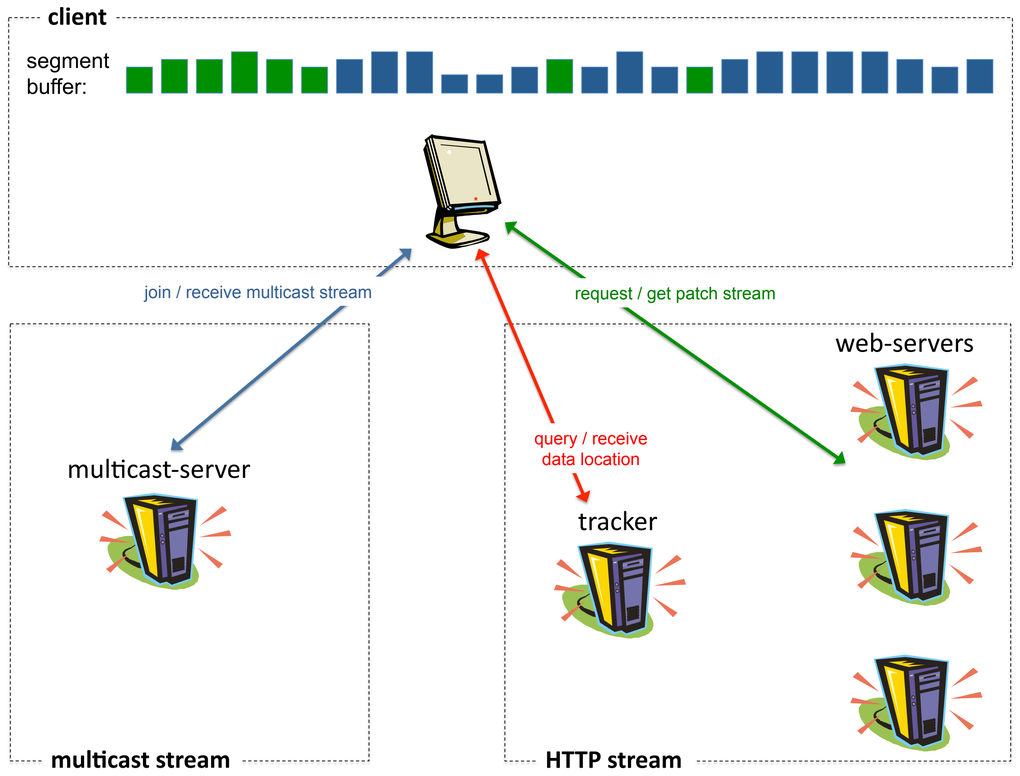
The video is divided into smaller segments where each segment is coded in several bit rates (qualities) for adaption. Each segment is retrieved using plain HTTP GET requests from traditional web-servers.





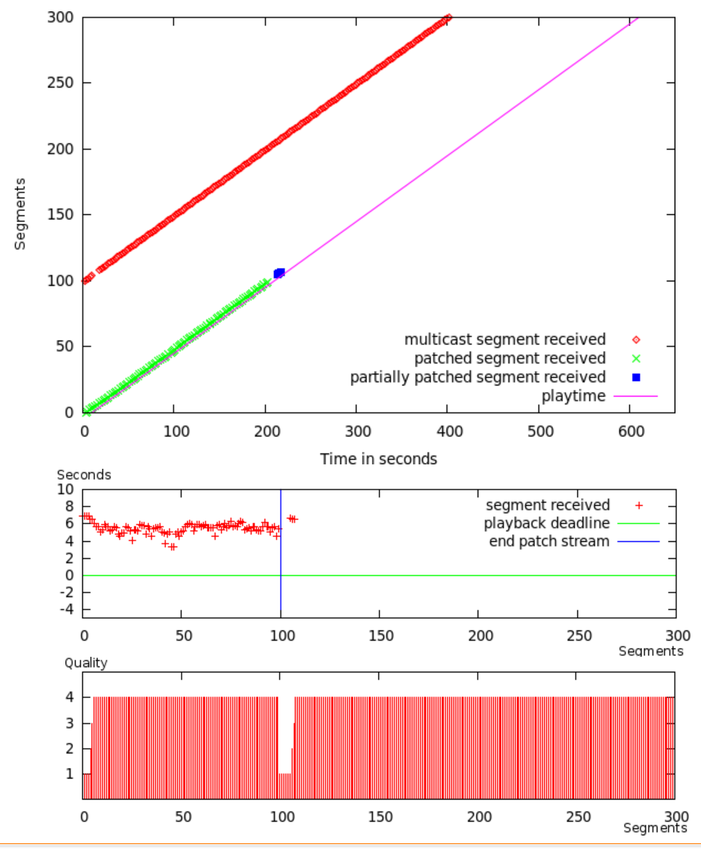
Pull-Patching

Pull-Patching combines the best of Patching and Adaptive Segmented HTTP streaming. The server multicasts the video stream as in traditional patching, but to enable adaptation, the video is segmented and coded in multiple bit rates, which are sent in separate multicast streams. A client then joins a multicast stream according to its available bandwidth and retrieves the patch and eventual repair streams using HTTP streaming.

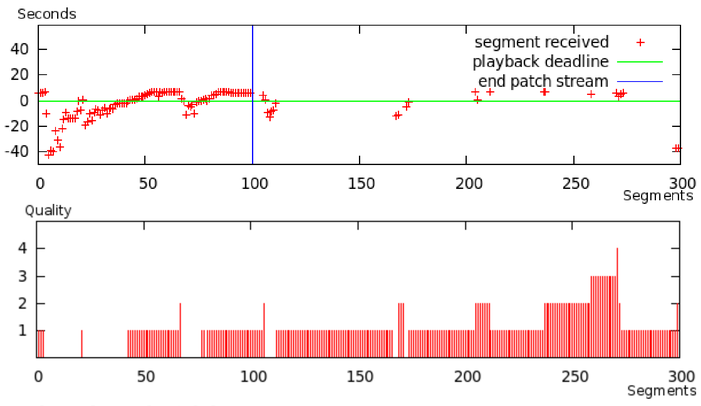


Experiments

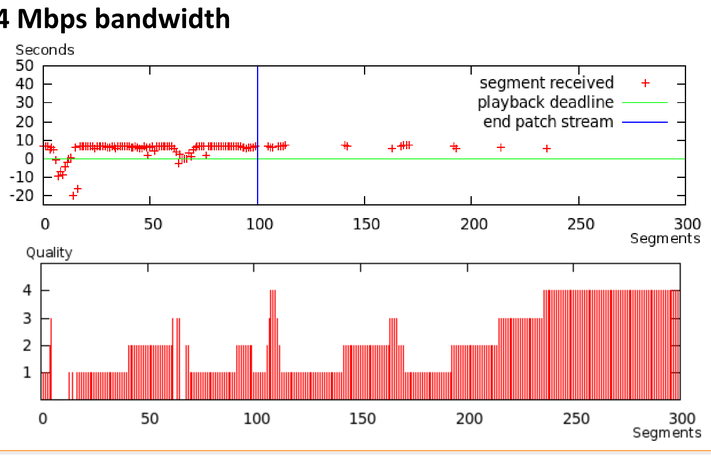
8 Mbps bandwidth



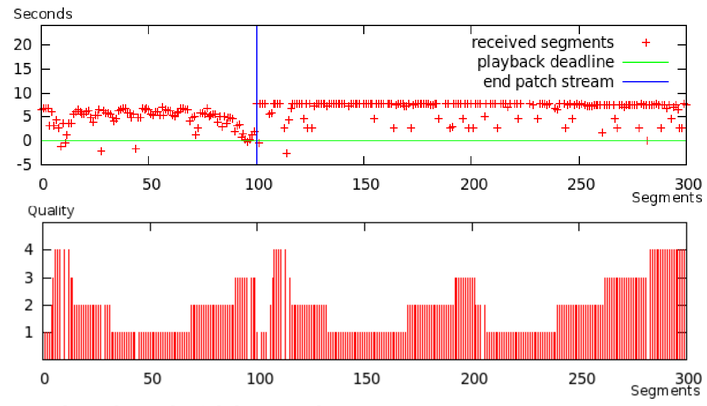
2 Mbps bandwidth



4 Mbps bandwidth



8 Mbps bandwidth, 1 % loss



8 Mbps bandwidth, 3 % loss

Conclusions

Pull-Patching overcomes the drawbacks of earlier patching approaches that make server-side decisions only and that suffer from bandwidth limitations, packet loss, delays and jitter, as well as patching server scalability. In contrast to previous schemes, Pull-Patching puts the client in control. It can adapt its video quality dynamically to locally experienced resource availability and handle lost and delayed packets by starting another patch stream.

Open Issues

- Multicast-patch-stream scheduling
- Adaptive (hierarchically layered) codecs
- Timing of repair-patch requests
- …