Improving the Performance of Quality-Adaptive Streaming over Multiple Heterogeneous Access Networks

Kristian Evensen (kristrev@simula.no), MPG@Simula, 23/2-2011
Co-authors: Dominik Kaspar, Carsten Griwodz, Pål Halvorsen, Audun Hansen

Using multiple links simultaneously introduces several challenges

Bandwidth heterogeneity

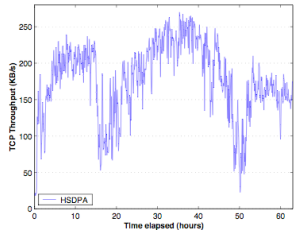
Vs.

Latency heterogeneity

Vs.

Challenges

Link dynamics



Reliability

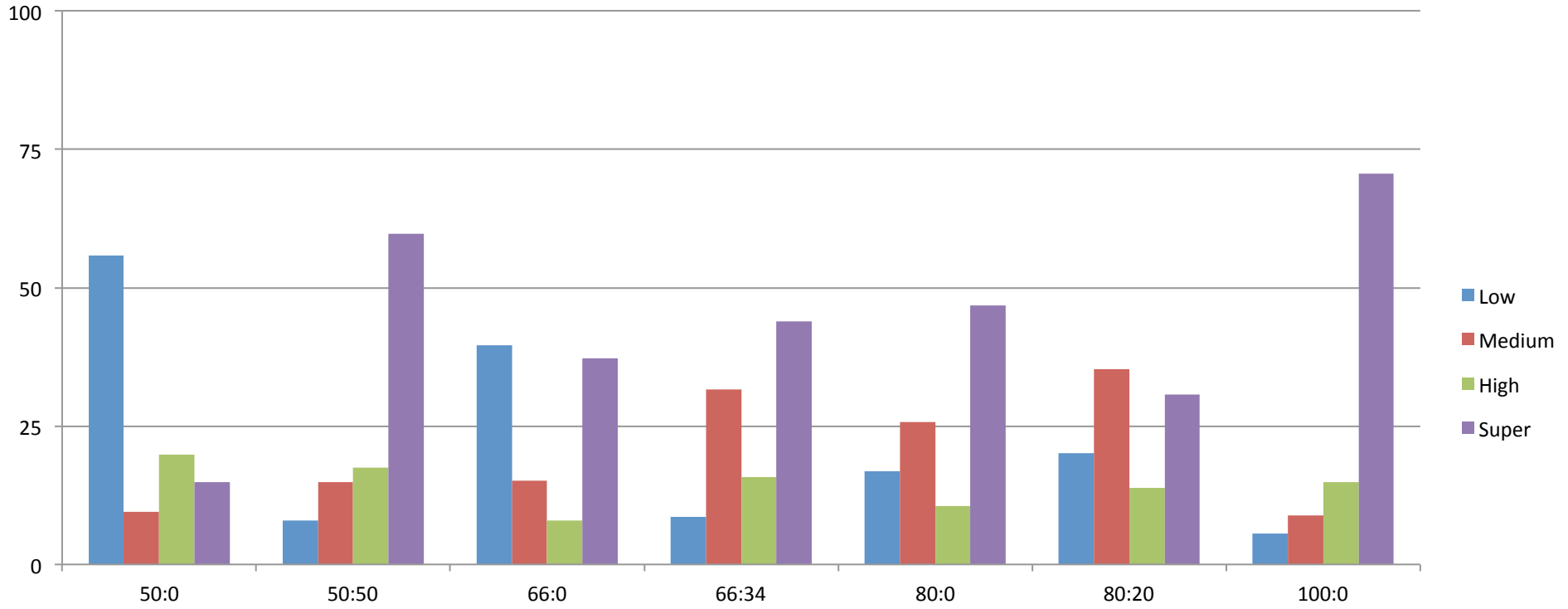
We have focused on bandwidth and latency heterogeneity, as well as emulated dynamics and real world links.

Single wireless links are often unable to support streaming of high-quality video.

Previously, we presented a novel approach for increasing the quality of HTTP-based streaming by requesting fixed size (logical) subsegments over independent links.

However, there is a problem …

Static subsegment approach, on-demand streaming and buffersize/startup delay of two segments



Bandwidth ratio. X:Y means that one link was allocated X% of the bandwidth, while the other Y%.

Unless the buffer is large enough to compensate for the bandwidth heterogeneity, the fixed size subsegment approach will never reach maximum performance.

In many scenarios (for example live streaming), increasing the buffer/startup delay is not desirable.

A more dynamic subsegment approach that takes the capacity of the different links into consideration is needed.

We wanted to evaluate the performance of our new approach for different types of streaming.

On-demand streaming

Live streaming with buffering

Live streaming without buffering

Earlier, we only focused on live streaming with buffering, however, different types of streaming introduces different challenges.

First step in reducing the effect of bandwidth heterogeneity: Improving the quality adaption mechanism

How long will it take to receive a segment in the specified quality:

transfer_time[quality_level] = segment_size[quality_level] / aggregated_throughput

How much content is buffered (in time):

duration_buffered_data = time_left_playout + (buffered_segments * duration)

How long will it take to receive the data that is already requested:

pipeline_delay = requested_data / aggregated_throughput

More parameters are needed to accurately calculate how much time can be spent receiving a segment without causing a deadline miss.

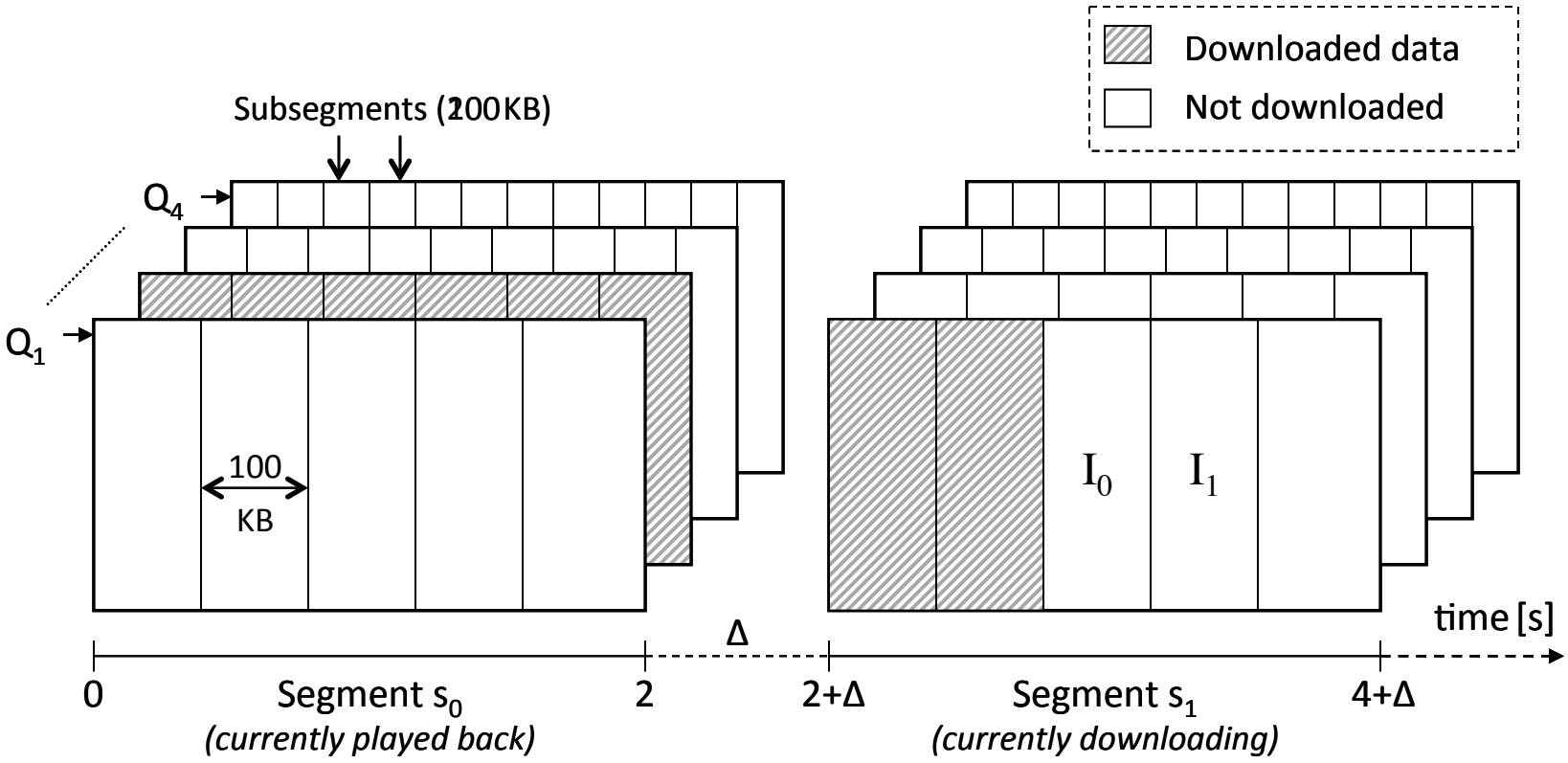
First step in reducing the effect of bandwidth heterogeneity: Improving the quality adaption mechanism

Quality adaption loop:

```
for quality level = ”super” to ”low” do
        if transfer_time[quality_level] < (duration_buffered_data – pipeline_delay)
                return quality_level
```

The new quality adaption mechanism considers all relevant delays and selects the most suited quality at the time of the request.

Step number two is to replacing the static subsegment approach:



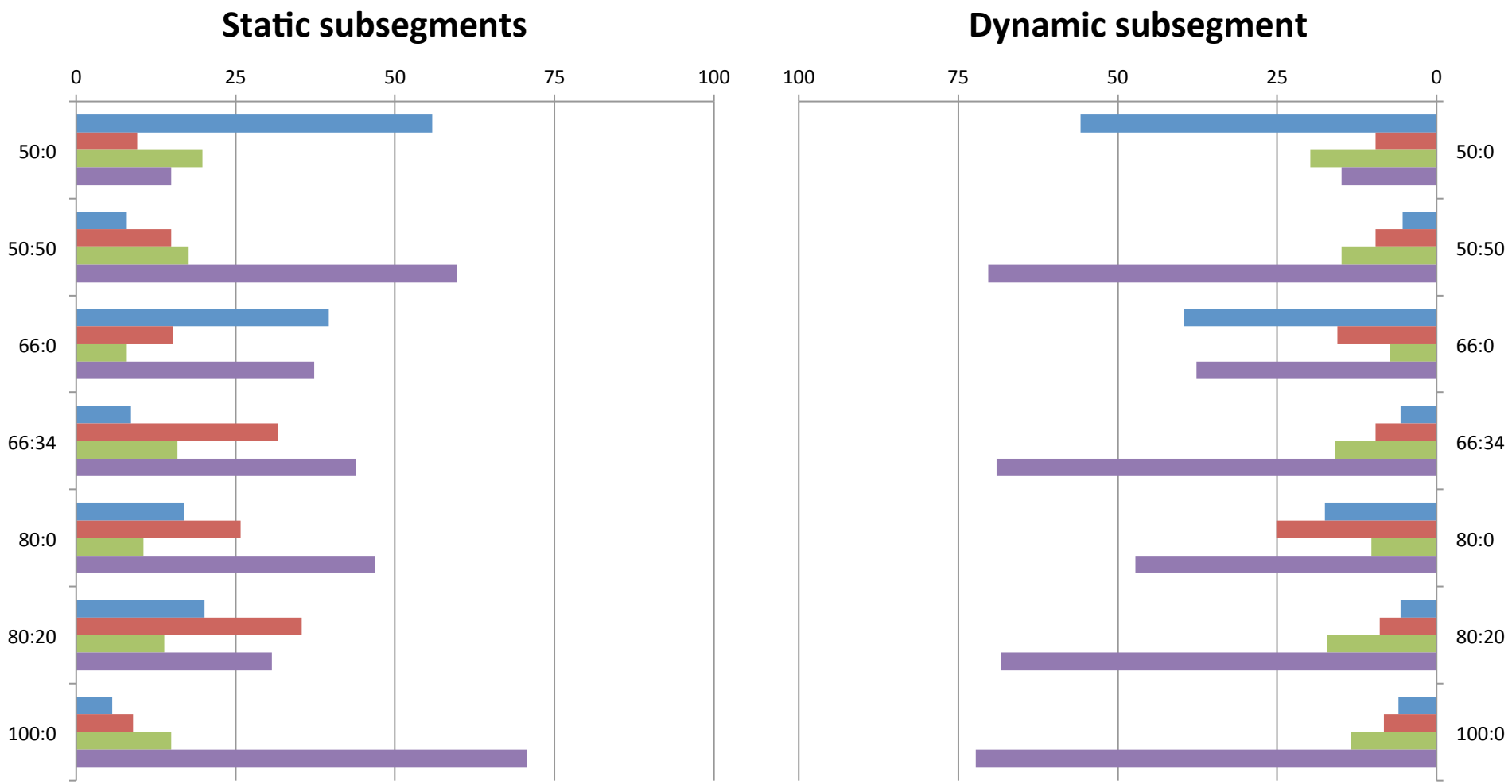
Dynamically dividing subsegments based on measured throughput reduces the share of data allocated to the slow link (the ratio between I_0 an I_1 is 3:2)

For testing purposes, 100 two second video segments were used. They have the following characteristics:

Quality level	Low	Medium	High	Super
Minimum bitrate per segment (Kbit/s)	524	866	1491	2212
Average bitrate per segment (Kbit/s)	746	1300	2142	3010
Maximum bitrate per segment (Kbit/s)	1057	1923	3293	4884

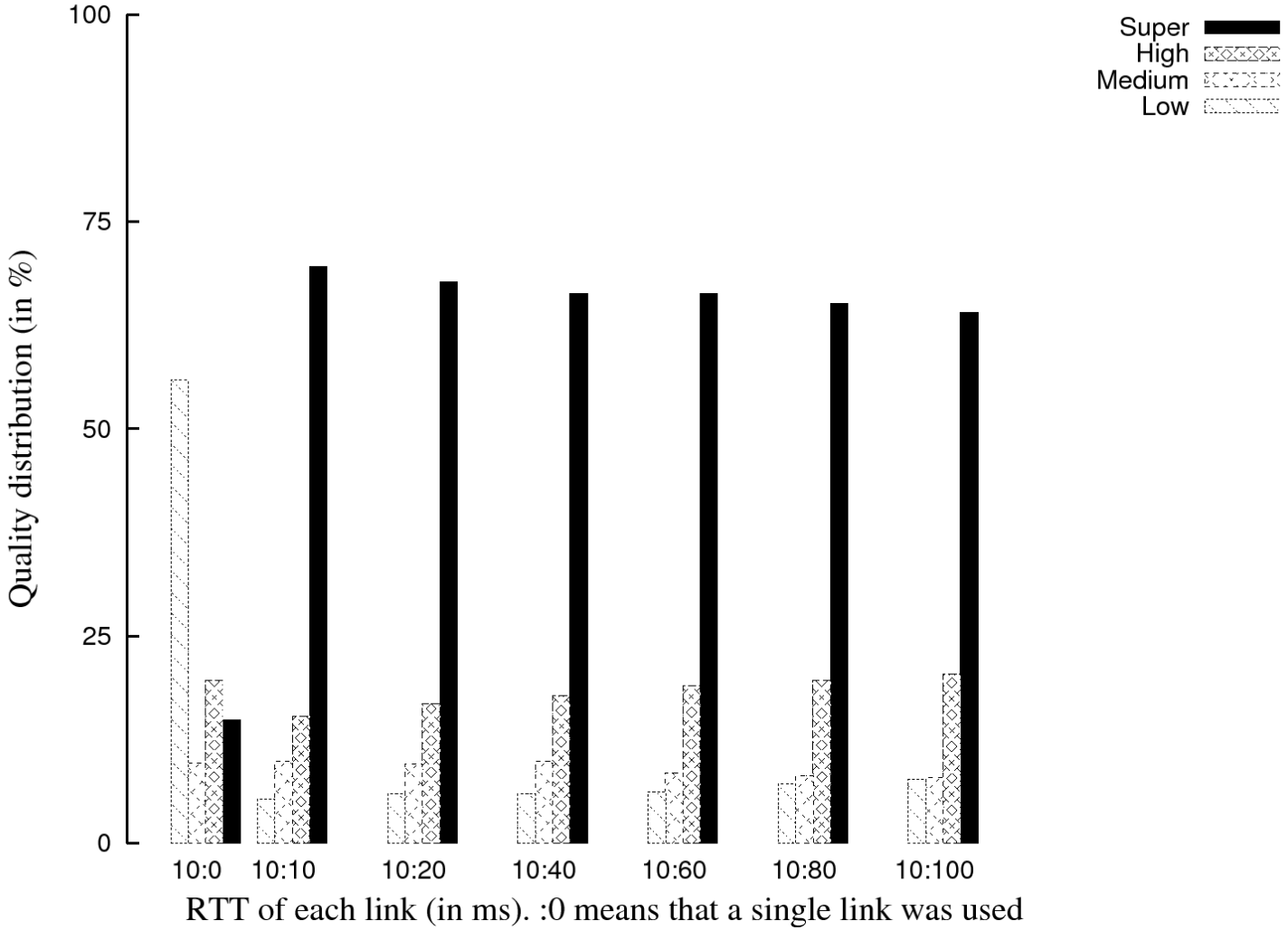
In all our controlled network environment tests, the combined bandwidth was 3 Mbit/s (i.e., the average requirement for super quality)

By dividing the subsegments dynamically, bandwidth heterogeneity should not have an effect.



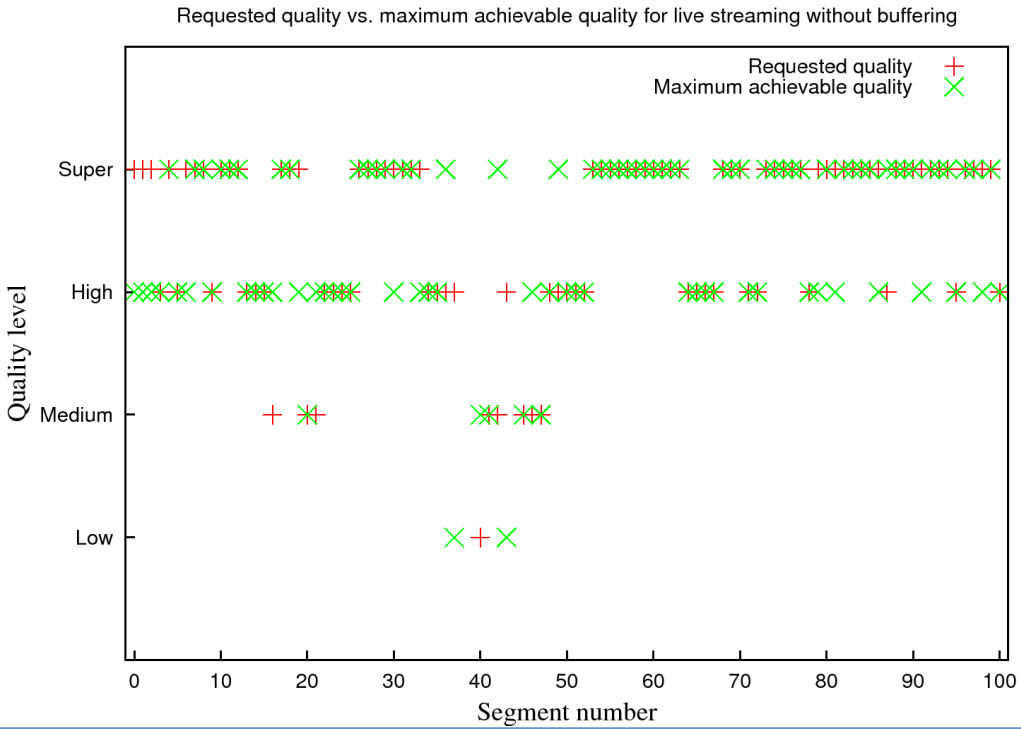
Considering link capacity enables allocating more accurate shares of data, and the video quality distribution is independent of heterogeneity.

Increased latency cause requests and retransmissions to arrive later, potentially affecting the performance



Smart use of HTTP pipelining ensures that data is requested early enough to avoid latency heterogeneity having a significant effect.

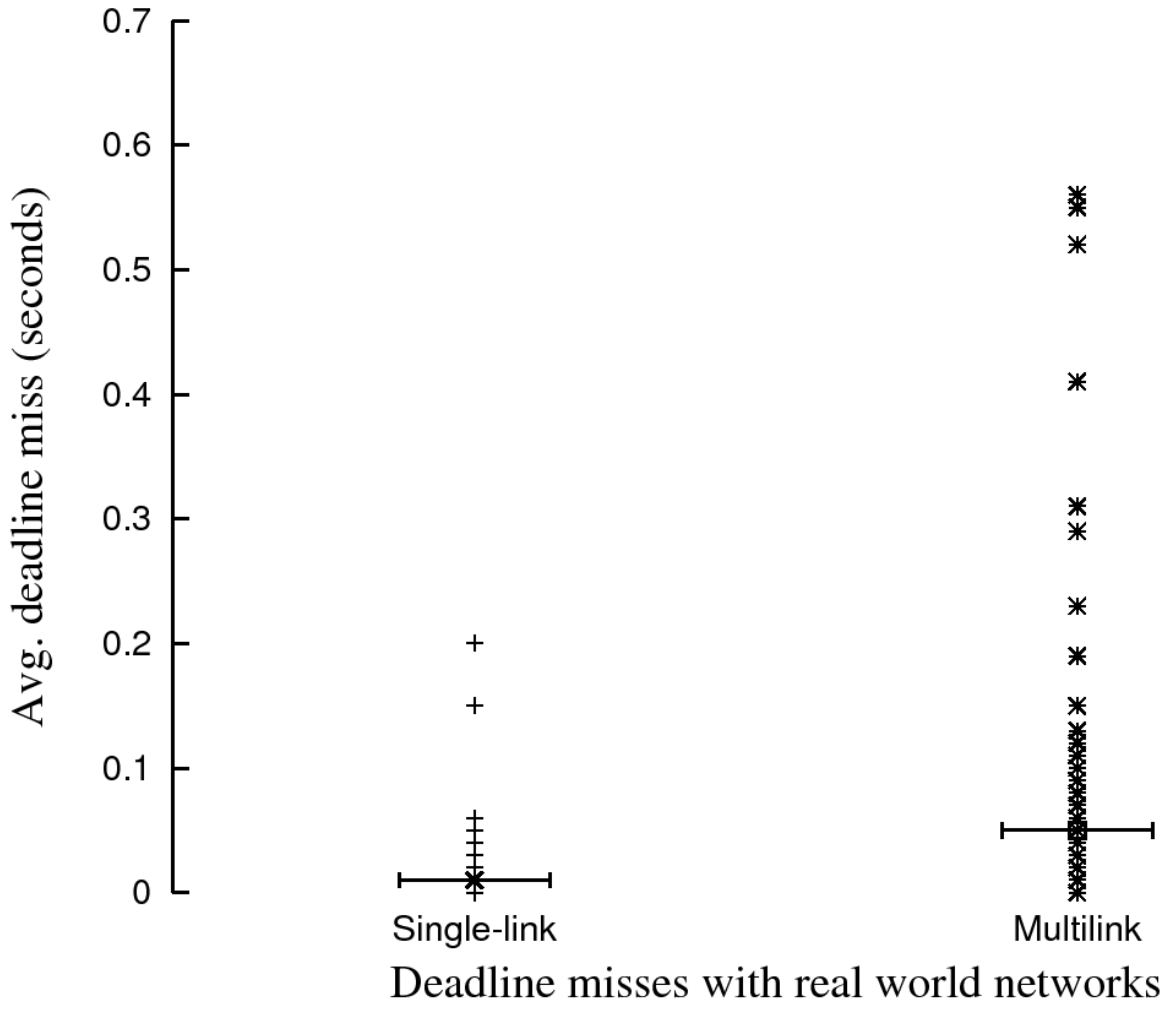
Performing experiments with real-world gives an impression of how the solution will actually perform.



WLAN (avg. throughput 287 KB/s) + HSDPA (avg. throughput 167 KB/s)				
Quality level	Low	Medium	High	Super
Single link	0%	27%	68%	5%
Multilink	1%	10%	35%	55%

A significant quality increase was seen for all streaming types when the second link was added.

For live streaming, deadline misses are especially important (who knows what you could miss?).

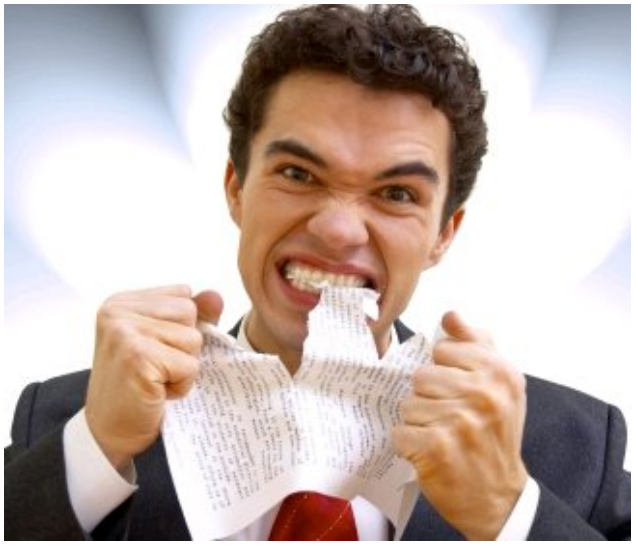


Deadline misses with real world networks

Using multiple links caused deadline misses, however, they were rarely severe. Also, for our tests, the number of skipped segments was acceptable.

In summary, the dynamic subsegment approach utilizes the links more efficiently.

Increased video quality for all three types of streaming



Quality is now independent of bandwidth heterogeneity

Questions?