Video Caching in the Access Network

Hendrik Buyle,

Supervisor(s): Bart Dhoedt, Filip De Turck

Abstract—**In this document we talk about installing caches on the routers in the access network of a digital television provider. We talk about the effects those caches have on various components in the network. We research whether or not those proxy servers are at all useful, where in the network those servers should be placed and which caching algorithms should be used by those proxy servers in which circumstances. In a last part of this article we suggest a few means of cooperation between those proxy servers.**

Keywords— **cache, access network, video streaming, time-shifted TV, caching algorithm**

I. Introduction

THE first part of this paper talks about a few basic concepts concerning video streaming, time-shifted television and RTSP Proxy servers in general.

Time-shifted television is a service which allows customers to pause, rewind and fast forward live television. It also allows the customer to start watching a particular show with a delay anywhere between a second and a few hours.

One of the most important aspects of Time-shifted Television is the demand curve. The demand curve describes how the popularity of a certain show evolves over time starting from the moment it is first broadcasted. The curve shows how many viewers will request to watch the show after it has gone live. It is important for the digital television provider to predict the demand curves as their shape will influence the results of proxy servers.

Caches inside the network can be used in roughly three different ways. They can be used in combination with a client (a settopbox with caching capabilities), they can be used inside the core network in order to lessen the load of the streamer. They can also be used on the intermediate nodes inside the access network.

The last part of this section gives an overview of how proxy servers work from a conceptual point of view. We describe the architecture of the proxy server as the sum of just three components: the cache itself, an algorithm called the cache algo and an algorithm called the topo algo. The cache stores the captured streams. The cache algo is the part of the proxy which decides on all actions to be taken by the cache, such as: should a certain piece a stream be cached or should another piece which already is inside the cache be removed from it? The topo algo is the algorithm which handles all decisions of the cache concerning cooperation with other proxy servers. After a request for a certain part of a show is received which cannot be served from the cache, that request must be forwarded to another proxy server or to a streamer. The algorithm that decides which component to forward the request to is the topo algo.

II. The simulator

Because almost all results used in this document are generated by simulator, it is important to have a section on the working of that simulator.

We need to have a simulator because it just would not be possible to try out the algorithms in a real test lab. The amount of machines needed to simulate an access network large enough to be of interest to us would just be too high.

The most important thing to know about the simulator is how the simulations differ from the real world. There are a total of six ways in which a simplification made by the simulator or a technical issue inside the simulator can influence the simulation results. I feel the reader must know about these simplifications if he is to interpret a simulation result himself or understand the conclusions i drew from my simulation results [1].

III. Non-cooperative algorithms

One of the features most described caching algorithms have in common is that they work with a sliding interval. An interval can be seen as a window in time, a succession of smaller segments belonging to the same television channel. Each of these algorithms caches the segments inside the interval and removes other segments who have fallen out of the interval. There is one interval for each television channel. Another feature of these algorithms is to divide the cache into two parts, an S and an L part. The S part will be used for temporary storage of segments in between cache recalculations. The L part will be used to store the segments as decided by the algorithm.

The algorithms described in this document are “Prefix Algorithm”, “Sliding interval with fixed interval length”, “Sliding interval with variable interval length” and “Survival of the fittest algorithm”.

- The prefix algorithm stores the first few minutes of every show with the intention to lessen the startup delay of a new RTSP stream.
- The Sliding Interval algorithm with fixed interval length uses one interval for each television channel and lets that interval slide, the size of the interval is fixed for each channel.
- The Sliding Interval algorithm with variable interval length also uses one interval for each channel. The main difference between both algorithms is that this one allows its intervals to grow and shrink to respond to a change in popularity.
- The Survival of the Fittest algorithm is like the sliding interval algorithm with the variable interval lengths but allows multiple intervals for each channel and considers intervals with just one segment in it to be intervals just as well.

IV. Means of cooperation

In this section we talk about how Proxy server can cooperate and communicate with each other in order to improve results. Examples of information proxy servers wish to share are: the amount of free space in their caches, their level of performance, the current amount of streams passing through the links con-

E-mail: hendrik.buyle@UGent.be

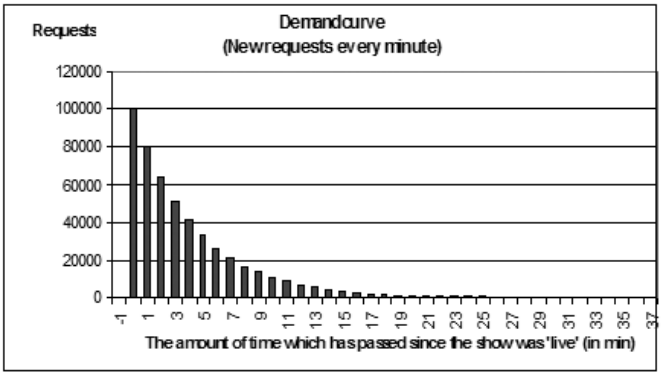

Fig. 1. Demandcurve

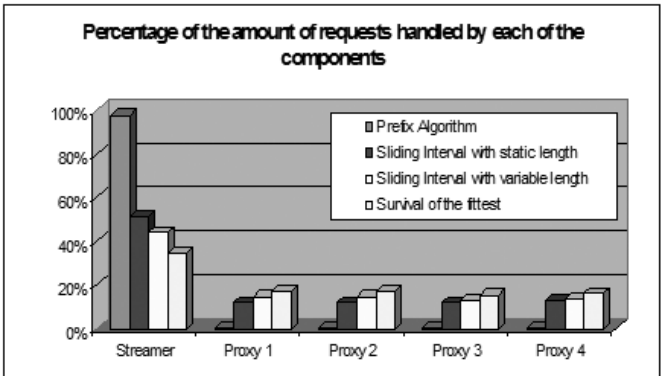

Fig. 2. Different caching algorithms

nected to the proxies, ... We suggest a centralized architecture is used to exchange this information with other proxies, one of the advantages of using a centralized server is that it can be used as a discovery service.

There are 4 different ways in which proxy servers can put shared information into good use:

1. In the calculation of the popularity of the segments
2. Enabling the promiscuous mode of the interfaces of the proxy server: this allows a proxy to capture and cache a stream which is passing through the router it is installed on.
3. Intelligent forwarding of requests: We suggest 4 different algorithms for the forwarding of requests:
   (a) Forwarding to the central server.
   (b) Normal Topo Algo: the proxy looks for another proxy which does have the segment to forward the request to
   (c) Shared Topo Algo: the proxy looks for another proxy which does have the segment to forward the request to, if it can't find one, it will forward the request to the proxy which has the most free space in its cache. This makes that second cache request the segment from the central server, allowing it to cache the segment.
   (d) Complex Topo Algo: the proxy looks for another proxy which does have the segment to forward the request to, if it can't find one, if will forward the request to the proxy which it has chosen based on a formula taking free space on that cache, performance of that proxy, weight of the path to that proxy, weight of the path from the streamer to that proxy, into account.
4. Intelligent caching: We suggest an algorithm which allows the provider to limit the number of copies of each segment on all proxy servers in the entire network to a specified amount.

V. Simulation results

Using the results from several simulations we can come the the following results:

The more flexible a caching algorithms is the better its results. Survival of the Fittest Algorithm gives us the best results, Sliding interval algorithm with fixed intervallength gives us the worst.

When there is no cooperation between caches the used topology has no effect on the performance of the caches.

The more flexible a caching algorithm is, the more its performance will improve if the caching algorithm would use a larger cache.

In most cases algorithms that need a value for ΔT will perform best with ΔT as small as possible.

Using promiscuous mode in combination with a simple forwarding strategy will result in a serious improvement of results.

The forwarding strategy with the best results is the Shared Topo Algo

If one wishes one could relieve the load on the streamer by using the Shared Cache Algo, the price to pay is a serious increase of the load of most links.

VI. Conclusions

In this paper several caching algorithms and several request forwarding strategies have been presented. We should remember that the Survival of the Fittest algorithm is the best Caching algorithm and that Shared Topo Algo is the forwarding strategy with the best results. Caches are very useful inside an access network and even inside the core network of a provider. Cooperation between those caches is also useful but it comes with a price. That price is an increase of the load of most of the links.

Acknowledgments

The author would like to acknowledge, next to the (head)supervisors, the daily supervisors: Wim Van de Meerssche and Tim Wauters. My final paper would not have been possible without their help.

References

[1] Hendrik Buyle, *Video caching in het access netwerk* final paper University Ghent, 2006.