IlmStream: Efficient Multimedia Streaming in decentralised distributed systems

Thorsten Strufe
Telematics-Dept., Technische Universität Ilmenau

Abstract

Using the P2P-Model to implement multimedia streaming has been in the focus of research for some time. Many works have lead to very interesting results in the aspect of load balancing, caching and request scheduling. Only a few of the projects consider the decoupling of the neighbourhood relationship of the overlay network and the topology of the underlying infrastructure. In order to reduce the overall network load and to achieve point-to-point connections with better quality-of-service parameters we introduce IlmStream, a Peer-to-Peer Streaming System that forms geographical clusters of collaboratively streaming nodes.

Keywords: multimedia streaming, Peer-to-Peer, geographical clusters

INTRODUCTION

In a client/server-like setup, multimedia streaming leads to very high bandwidth loads on the server link. Especially if content of temporarily high interest is served, a request-overload at the server is observable, commonly known as flash crowds or the "slashdot-effect".

Many possible load balancing solutions, like multicast-communication, server-farms and content distribution networks[1] (CDN) have been proposed and implemented to solve this problem and to relieve central servers.

Anyway, in many circumstances these technologies can not be used. IP-Multicast on the one hand is only a solution in a homogeneous environment, where all members of the group have similar link capabilities. In real-world streaming domains this presumption can not be made, as very heterogeneous hosts with different up and downlink speed are requesting the same content. CDN and server farms on the other hand are very expensive as well as difficult to set up and to manage.

Another very promising idea is the use of decentralised distributed systems, also known as Peer-to-Peer-Networks, offering what is sometimes called end-host-based- or application-layer-multicast. In this approach every client downloading content in turn offers the retrieved data for other interested hosts to download at the same time. This leads to growing and self organising load balanced systems of cooperating peers.

[1] http://www.akamai.com

In the research of Peer-to-Peer-Streaming, four major tasks have to be solved. 1) Content and sources have to be located; hence a location service is needed. 2) After finding sources, a multicast-tree has to be built. 3) A scheduling-mechanism is needed to choose useful fragments of the datastream which as a result have to be requested. As a subtask, a flowcontrol has to be provided. 4) To insure quality for dynamically joining peers, a caching strategy has to be implemented.

Quite a few research groups [1, 2, 5, 6] have come up with Peer-to-Peer-Streaming-Solutions, each optimising either caching, scheduling or reorganisation of the resulting multicast tree.

In most of these approaches one problem is left to be solved: the fact that the topology of the overlay network is completely decoupled from the topology of the underlying real transmission network. The solution to this problem would lead to three advantages:

The ability to:

1) choose reliable short-distance connections with good QOS-parameters over longer distance connections, which are more susceptible to delays and jitter.
2) stream non-redundantly into intra-organizational broadband LAN's which are connected to the internet through *narrowband* links and which usually are *more expensive* to use.
3) keep traffic inside the autonomous systems of network carriers because it is
    3.1) quicker to find alternative routes on link errors of any sort and
    3.2) cheaper for the carrier.

As a solution, we propose the approach of locality aware peers, which form local clusters and which will result in better bandwidth efficiency. With this approach, we want to show that geographical clustering of streaming hosts, while leading to more efficient bandwidth use, does not reduce the QOS, does not lead to quality loss due to additional signalling or a lack of data sources.

PEER-TO-PEER MULTIMEDIA STREAMING

Classifying the domain leads to three different classes of multimedia streaming in Peer-to-Peer-Systems which have to be taken into consideration. The first class is live streaming. This setup is characterised through an unimportant history and through having a one-to-many communication-relationship: the stream has a central source and potentially very many sinks. Examples are the streaming of a news show or share prices from a stock-market. A second class is specified by having a central source and a relevant history. A good example of this class is the stream of a sports event, where the viewer might want to watch the last highlight again. A third class is Video-on-Demand, a setup with potentially many sources and many sinks, while the history might not be of interest, but the point in time for a new sink to join is not determined and hence the stream has to be available in the system at all times.

When designing a Peer-to-Peer multimedia streaming service, the target class should be taken into account, as it has a major impact on the needed infrastructure and features. While for the first class, a buffering mechanism, with an input queue is sufficient, for the other two classes, a stable caching function has to be introduced. Again, while for the first two classes, a very basic lookup service, which merely registers participating hosts, is needed, Video-on-Demand relies on a scalable distributed location service, in order to be able to locate the multimedia streams of interest. We focus on the second and third class in our work. In effect, a stable location service and caching functions have to be introduced.

RESOURCE LOCATION

Location has to be divided into two different tasks. At first, a multimedia stream of interest has to be found, and secondly, potential sources for this stream have to be located.

For the first two streaming classes, with only one main original streaming source, the first step will usually be done via context information. The service provider will have a webpage with a link or reference to the lookup service. In the third class, however, such information will usually not be available, for there is no single or dedicated source of the stream and a different location service is needed.

For systems of collaboratively streaming hosts, the subtasks of source location and organisation of the evolving overlay network (which forms the multicast-tree), are usually implemented rather heuristically.

Most solutions keep the whole management information, in a way similar to the 'bittorrent'[2], at the peer which is the first source in the multicast tree (the seeding peer), or on one dedicated host. This host then acts as a 'lookup service', merely collecting, managing and serving the addresses of participating hosts. Every host wanting to subscribe to the stream contacts this lookup service, requests information about other participating hosts and contacts these as potential sources. In this setup the lookup service can be used to organise the multicast-tree, as it can submit only addresses of a subset of the participating peers. For example a subset of peers with the highest offset in the stream, to the newly joined peer.

This introduction of a single dedicated service has a number of drawbacks. In very dynamic contexts, with frequently joining and leaving peers for example, the central location service can be a bottleneck leading to scalability problems and is always inherently a single-point-of-failure. Another problem arises if there is more than one seeding peer, as the one peer offering the lookup service has to be elected and a loadbalancing mechanism needs to be introduced. On failure of this service, all management information is lost.

The focus of our work is bandwidth-efficient and stable resource-location. Streaming-functions, such as scheduling, caching and presentation are not in the focus of this paper.

As we know stable and scalable decentralised location services from other domains, we introduce a Peer-to-Peer-System. This approach forms geographical clusters of hosts, supplying information which can be used to build efficient multicast-trees, to circumvent the problems mentioned above. With the decentralised distributed location service used, it is possible not only to find a multimedia-stream of interest, but participating hosts and additional information about them.

[2] http://www.bitconjurer.org

LOCATION SERVICE

The location service offers lookup of resources and as a result always returns the closest sources. The design, as shown in Figure 1, follows earlier research [4, 5], where better efficiency of bandwidth consumption could be shown for simple Peer-to-Peer-systems.

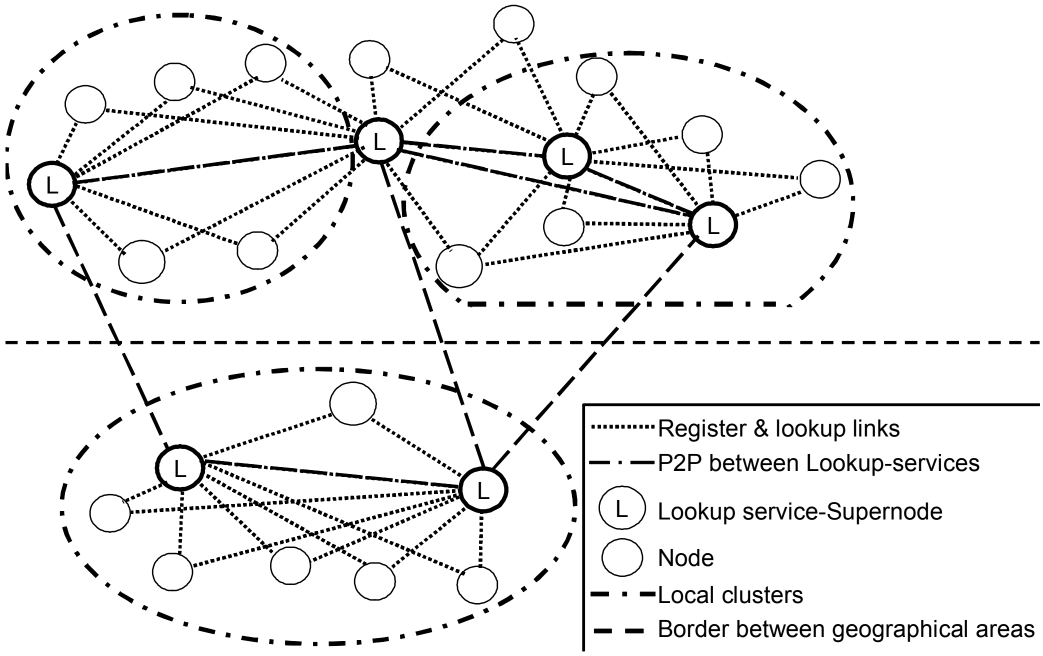

Figure 1: Locality-aware Peer-to-Peer location service

Roles

To circumvent the known scalability problems of flat and unstructured Peer-to-Peer approaches, the system consists of dynamically reconfiguring peers with two different roles: nodes and supernodes.

On joining the system, a host acts as a normal node and registers itself at a nearby supernode, before it submits its request for close sources of the desired stream. Every node can dynamically reconfigure itself to become a supernode, offering additional services.

The supernodes form a Peer-to-Peer-Network and provide a registry and lookup service for their nodes. They act as search gateways – any messaging dealing with lookup of sources, streams or other supernodes is done via the supernode network – and they keep track of streams and stream-offsets of the registered nodes.

Communication and Reorganisation

On request, every host, which is not connected already, bootstraps to the system, locates nearby stream-sources and starts streaming, following the Algorithm shown in Figure 2.

Algorithm ClosestSources

1 *Bootstrap to first supernode*
2 *request k supernodes* → sn^k
3 *choose set of closest supernodes* → SN_c
4 *set closest sn as primary supernode* sn_p
5 sn_p*::locate(streamsources)* → ss^n
6 *choose close sources and request stream fragments*
7 *for each new connection swap* SN_c
8 *if* $SN_c \neq SN_{c\_received}$ *goto 3*

Figure 2: Algorithm to choose closest sources

This algorithm leads to dynamically reorganised and efficiently structured multicast trees in grown and stable systems.

To evolve a new system of collaboratively streaming peers or to enlarge a system to new local areas, new clusters have to be created. The condition leading to the insertion of a new cluster is shown in Figure 3.

C_K:=subset of nodes, n:=node, sn:=supernode, RTT:=roundtriptime
SN_{ic}:=set of close supernodes known by node i, N:=all nodes

$$\exists C_K : |C_K| > 4 \wedge n_i \in C_K \Leftrightarrow \forall n_i \in N \; \forall n_j \in C_K \; \forall sn \in SN_{ic} \; RTT(n_i, n_j) < 2RTT(n_i, sn_l) \mid i,j,l \in N: i \neq j$$

Figure 3: Condition for cluster insertion

In each new cluster, one or more nodes have to be reconfigured to supernodes following an election algorithm with link-bandwidths as the deciding parameter. Registry-overflow, if the number of registered nodes at a supernode exceeds a dynamic upper border as shown in Figure 4, is a second condition for the election of a new supernode, leading to a new cluster.

sn_{ir}= nodes registered at supernode i
SN = all supernodes, N = all nodes

$$|sn_{ir}| > 2 \frac{|N|}{|SN|}$$

Figure 4: Registry-overflow

As it is obviously impossible to gather the exact set of all participating nodes and supernodes, these numbers are being estimated.

CONCLUSION AND FURTHER WORK

So far a lot of work has been done on the research of different approaches to load balancing through using Peer-to-Peer-Systems to stream multimedia-data.

The described system supports the second and third streaming classes, one-to-many streaming with caching and Video-on-Demand. It saves redundant bandwidth through the introduction of the signalling hierarchy and overall bandwidth through the introduction of locality awareness. Some work still remains to be done.

The first step of all has to be to implement the proposed system and to test, to see if the location awareness can improve the quality at all.

This step should clarify three questions:

1. Is the unstructured topology of overlay networks really a problem for multimedia streaming, or is the backbone-network able to cope with the inefficient and redundant data transfer)
2. Does the additional signalling needed, outweigh the advantage of streaming from nearby nodes?
3. Is the benefit of location awareness significant, or are there too few multimedia streams and too few viewers to establish a stable, clustered overlay network?

The first two questions can be answered through implementing and testing or simulation of the proposed system. The third question is more complex, as up to date information about the number of viewed multimedia streams and the distribution of the hosts is needed.

Another open question is, if the asymmetric links to end-users like ADSL, which are very popular in Europe, will lead to a source-drain. This problem however is inherent to every Peer-to-Peer Streaming System. We plan to explore this question through setting up a video-streaming service of amateur sports events in Europe.

We are confident that multimedia streams will become one of the main data-types transferred through the internet in the future and that collaborative streaming and location awareness will help to reduce overall network and server loads while leading to better quality of service.

REFERENCES

1. D. A. Tran, K. A. Hua and T. Do.”*ZIGZAG: An efficient Peer-to-Peer-scheme for media streaming.*” In *Proceedings of the* IEEE INFOCOM, 2003
2. Reza Rejaie, Antonio Ortega, “*PALS: Peer-to-Peer Adaptive Layered Streaming*”, `http://citeseer.nj.nec.com/576526.html`, 2003
3. Thorsten Strufe, “*Klassifikation von Namens- und Lokalisierungsdiensten in dezentralen verteilten Systemen*“ In *Tagungsband der 9. net.object days*, 2003
4. Thorsten Strufe, Dietrich Reschke. „*Efficient Content-Distribution in semi-decentralized Peer-to -Peer- Networks*“ In Proceedings to the 8th International Netties-Conference, 2002
5. T. P. Q. Nguyen, Avideh Zakhor, “*Distributed Video Streaming over Internet*” in Multimedia Computing and Networking, 2002
6. Mohamed M. Hefeeda, Bharat K. Bhargava “*On-Demand Media Streaming Over the Internet*”, `http://citeseer.nj.nec.com/570297.html`, 2003