Proxy Affinity

Christian Spielvogel, László Böszörményi

UNIVERSITÄT KLAGENFURT

Institute of Information Technology
University Klagenfurt
Technical Report No TR/ITEC/07/2.06
July 17, 2007

ABSTRACT An approach for combining Peer-to-Peer systems and Content Delivery Networks is presented. The combined system is called Proxy-to-Proxy and used to enable quality-of-service based multimedia delivery. The Proxy-to-Proxy network is characterized by consisting of multiple proxy groups. The groups are formed using a metrics called proxy affinity. The metrics combines three utility values, namely (1) Network Closeness, (2) Semantical Closeness and (3) Load Closeness. So the groups are (1) located close to end-clients, (2) homogenous concerning the type of content and (3) balanced concerning the amount of required resources. Network Closeness, Semantical Closeness and Load Closeness can be weighted against each other. Weighting forces either pure Peer-to-Peer behavior, pure Content Delivery Network behavior or a combination of both. We compare all three behaviors against each other and examine the influence on the quality of streamed multimedia content. In order to emulate the system behavior and evaluate it against existing Peer-to-Peer and CDN approaches the NS-2 [8] based gnutella-simulator (GnuSim) [2] has been combined with a tool called EvalVid [4]. EvalVid allows mapping packet delays and losses to the quality of real videos within NS-2. The quality of the received content is evaluated using an objective metrics called Mean Opinion Score (MOS).

KEY WORDS
Distributed Multimedia, Quality of Service (QoS), Peer-to-Peer (P2P) systems, Content Distribution Networks (CDN)

1 Introduction

When multimedia data are streamed over a best effort network it is challenging to provide the end-user with the expected quality. The expected quality is achieved by avoiding packet losses and high delays. Well known approaches to achieve this goal are Peer-to-Peer systems and Content Distribution Networks.
In this work we focus on the problems of Peer-to-Peer systems and Content Distribution Networks and propose a new solution called Proxy-to-Proxy. The main idea followed in the Proxy-to-Proxy approach is to compensate disadvantages of Peer-to-Peer systems and Content Distribution Networks by combining the advantageous characteristics. The advantage of Peer-to-Peer systems is that peers are located close to end-users which is not necessarily true for Content Delivery Networks. The advantages of Content Delivery Network nodes are that the nodes are always on, have good network connectivity and high storage space which is often not the case for Peer-to-Peer nodes.
Combining Peer-to-Peer and Content Delivery Network characteristics is based on a metrics called proxy affinity. Proxy affinity includes three utility values called Network Closeness, Semantical Closeness and Load Closeness. Network Closeness is used for building groups "close" to future client requests. This resembles peer-to-peer behavior. Load Closeness is used for balancing the amount of required and available resources between alternative groups. Resource balancing is a typical feature of Content Delivery Networks. Semantical Closeness is required for making the groups homogenous concerning the type

of shared content. This characteristics can be found in Peer-to-Peer systems as well as in Content Delivery Networks.
In order to compare Peer-to-Peer, Content Delivery Network and Proxy-to-Proxy behavior against each other in the same situation the system characteristics have to be changed. Changing the characteristics can be done dynamically by setting different weights to Network Closeness, Semantical Closeness and Load Closeness.
The system has been fully implemented and can be used to stream MPEG-1,2,4 multimedia content over IP based networks. For the evaluation the system has been combined with network simulator NS-2 [8]. Using this combination real media stream can be transmitted under different network conditions. The result from the emulation is presented in section 5.1.

2 Related work

Content Distribution Networks are operated by third party providers. The main components are surrogate servers in the backbone areas of network providers. On the server side they provide reliability and scalability. On the client side they provide high throughput with low latencies. Content Delivery Networks direct each client request to the most appropriate surrogate server using DNS or HTTP redirection. The original application area has been the distribution of web objects. The main optimization criterion is to minimize the replication costs between the origin- and the surrogate servers. Nowadays, the systems are also used to distribute multimedia content [?] [?]. The second approach are Peer-to-Peer systems where workstations of individual users collaborate to build a distributed system. Content that is downloaded by one peer is usually made available for all other peers. So popular content is highly available. Peer-to-Peer systems mainly differentiate in the architecture they provide for content retrieval. The architecture can either be fully centralized, fully decentralized or hybrid [?].

2.1 Problems Content Distribution Networks

Figure 1 shows a simplified scenario for a typical CDN [7] containing one origin server, a set of M high performance surrogate servers with high speed network connections and a set of N clients. One would assume that once there are enough surrogate server resources to serve the requests, performance bottlenecks can be avoided. The problem in this example is not the server performance but the geographical location of the surrogate servers relative to the clients with a bottleneck in-between.

As a result the quality of the content streamed from the surrogates would be worse than from the original server. One solution for this kind of problem is CDN peering [?], where providers temporally rent surrogates from other CDN providers that are closer to actual client requests. In the example scenario CDN peering is not applicable because we assume that there are no surrogates with free capacity from other CDN providers. Redirecting the requests to the origin server would not scale and render the Content

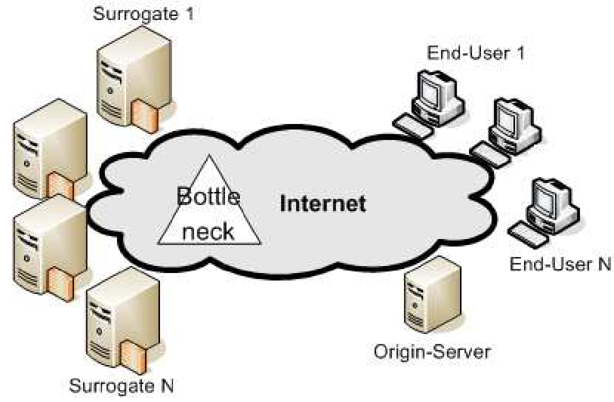

Figure 1: Content Delivery Network Szenario

Distribution Network unnecessary.

2.2 Problems of Peer-to-Peer systems

Peer-to-Peer systems are based on the assumption that content is downloaded from a "nearby" peer and not from the origin server. The downloaded content is again shared for other peers which increases scalability and download performance. Popular content is usually available by multiple peers and can be served by them in cooperation [3]. The main drawbacks of peer-to-peer networks are the fragility and the usually low network capacity of the individual peers. Fragility means that the up-time of a peer completely depends on the end-user. Upload capacity is usually low because according to [3] most home users have ADSL or cable modem connections. In scenario visualized in figure 2 we focus on the uptime of the peers combined with the content distribution among them. Take for example a video having a bit rate of 600 Kbit/s and a playback time of one hour.

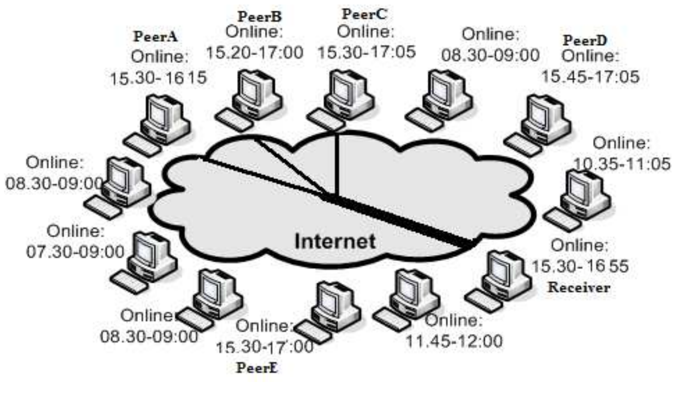

Figure 2: Peer-to-Peer System Szenario

The video is provided by the peers A,B,C. Each of the peers has an upload capacity of 250 Kbit/s, the aggregated bandwidth is 750 Kbit/s. The playback starts at 15.45 and ends at 16.45 o'clock. Peer A is switched off after 30 minutes (at 16.15 o'clock). So from 16.15 o'clock onwards the aggregated bandwidth from Peers B and C is 500 kbit/sec but the video still requires 600 Kbit/s. There are other peers with sufficient resources, like peer D or peer E. The peers D and E don't share the required and cannot be forced to. As a result the receiver peer is only able to view the content with the original quality for the first 30 minutes.

3 The ProXy-to-ProXy Network

In this section we describe how to combine the Peer-to-Peer with the Content Delivery Network approach. The new approach is called Proxy-to-Proxy. Proxy-to-Proxy is aimed to get rid of the problems Peer-to-Peer systems and Content Delivery Networks have (see section 2.1 and 2.2).
The core components are workstations (proxies) dedicated for storing, manipulating and streaming MPEG-1,2,4 video and audio content. The workstations are located in local area networks or connected to the Internet using dial-up lines. Each proxy learns about other proxies by connecting to a distributed Domain Name System (DNS) [5]. Proxies cooperate with other proxies by forming groups. Groups are characterized by an unique leader and a number of proxies. The group formation process (section 3.4) is based on combining Network Closeness (section 3.1), Semantical Closeness (section 3.2) and Load Closeness (section 3.3). The weighted sum of the three measures gives the so-called Proxy Affinity (see section 3.4)

3.1 Network Closeness (NC)

Network Closeness is a metrics to maximize the throughput between the proxies and future clients. The difficulty is that future clients are not known by the time a group is created. What is known instead is that each client is connected to one proxy (entrance-proxy). The entrance-proxy and the end-client have to be in the same local network. By knowing the absolute position of the proxy-gateway, the positions of the "future" clients are also known. A logical view of a proxy group can be found in figure 3. The groups leaders

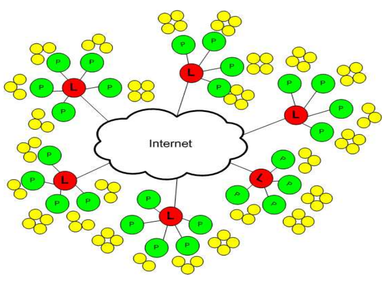

Figure 3: Logical Proxy-Client View

are labeled red, the proxy-gateways are labeled green and the end-clients are labeled yellow. It is assumed that bottlenecks occur on the paths between the proxies and not between the gateway and the end-client. Making this assumption, it is only necessary to avoid the bottlenecks between the proxies to serve requests without packet loss and jitter. Therefore each new proxy is added to the group where it has the highest average available bandwidth (Network Closeness) to all N group members. $NetworkCloseness$ to one group is calculated as:

$$NetworkCloseness = \frac{\frac{1}{N}\sum_{i=1}^{N} AvailBW(i)}{AvailUploadBW}$$

N is the number of current group members, AvailBW is the measured available bandwidth between the new proxy and group member $i, 1 \leq i \leq N$. The upload bandwidth of the new proxy is *AvailUploadBW*. The higher the available bandwidth to all group members the better is the Network Closeness value. In case that the AvailBW is equal to AvailUploadBW, Network Closeness takes the value 1. Otherwise it is between 0 and 1.

3.2 Semantical Closeness (SC)

Semantical Closeness is used to make proxy groups homogenous. In a homogenous group all proxies share the same type of content. In the current system the type of content is expressed by combining video genres and playback times. Genres are required to distinguish between types of movies. So it is possible to differentiate for example between a scientific documentation and an entertainment movie. Playback times are required to distinguish between the trailer and the full version of the same movie. For calculating *Semantical Closeness*, all movies need to be mapped to disjoint categories. The average playback duration for all movies in category c is calculated as:

$$AvgPb_c = \frac{1}{S} * \sum_{s=1}^{S} playbackTime_s^c \tag{1}$$

where S is the number of movies belonging to the category. In the next step for each category c that is available on the new proxy and within the examined group semantical closeness SC_c is calculated.

$$SC_c = \begin{cases} \frac{avgPb(G)}{avgPb(P)} & if avgPb(G) < avgPb(P) \\ \frac{avgPb(P)}{avgPb(G)} & if avgPb(G) \geq avgPb(P) \\ 0 & if\ avgPlayback(P||G) == 0 \end{cases}$$

For comparing alternative groups against each other it is necessary to calculate the average semantical closeness SC for all categories CA on a per group basis:

$$SC = \frac{1}{CA} \sum_{c=1}^{CA} SC_c \tag{2}$$

3.3 Load Closeness (LC)

Load Closeness represents the relationship between the currently available and required resources for a group. In the actual state of the system the only resource considered is the available upload bandwidth of the proxies. Load Closeness LC_p for proxy p is calculated as:

$$LC_p = 100 * max(1, \frac{UploadBW_{Required}}{UploadBW_{Available}}) \tag{3}$$

The *required* upload-bandwidth $UploadBW_{Required}$ is the accumulated bit rate of all N currently streamed data flows:

$$UploadBW_{Required} = \sum_{i=1}^{N} bitrate_i$$

The LoadCloseness for a group is calculated by averaging the Load Closeness values for all members M:

$$LoadCloseness = \frac{1}{M} \sum_{i=1}^{M} LoadCloseness_i \tag{4}$$

LoadCloseness can take values between 0 and 100%. The higher the value the more loaded is the group and the more additional ressources are required.

3.4 Proxy Group Formation

Proxy groups are built driven by the notion of *ProxyAffinity.* Proxy Affinity is the weighted sum of *Load Closeness (LC), Semantical Closeness(SC) and Network Closeness(NC)*:

$$ProxyAffinity = \alpha * LC + \beta * SC + \gamma * NC \tag{5}$$

If a new proxy enters the system, it joins that group to which it has the highest affinity value. In case that all factors are weighted equally the system behaves like a combination of a Peer-to-Peer system and Content Delivery Network. The experimental results using this system behavior can be found in section 5.3 In order to enforce pure Peer-to-Peer behavior α and β are set to 0. In this case groups are built close to future clients but the content is not homogenous and the request load is not balanced. The experimental results using this system behavior can be found in section 5.1
In order to enforce pure Content Delivery Network behavior β is set to zero. Using this parameter setting the group is homogenous and the load is balanced but the content is not stored close to the client locations. The experimental results using this system behavior can be found in section 5.2.

4 The Simulation Model

Our simulation model consists of three generic layers. The first (bottom) layer contains the packet-level network simulator NS-2 [8] and Brite [6]. The goal of Brite is to generate

accurate synthetic structures for NS-2 that reflect real Internet behavior.
The second layer is the Peer-to-Peer protocol layer based on the gnutella simulator [2].For the Peer-to-Peer simulation we have used the gnutella protocol [1] which is one of the most popular protocols for distributed peer-to-peer file sharing applications like LimeWire, Gnucleus or BearShare. For the CDN and the Proxy-to-Proxy simulation we have added a central content management instance that decides about request redirection.
The third layer embodies the application behavior. Nodes can be online or offline, do search or download content. This layer we have enhanced by merging EvalVid [4] which is a tool-set for evaluating the quality of videos transmitted over synthetic network connections.
EvalVid enables to measure QoS parameters of the underlying (simulated) network, providing methods for determining frame loss and delay. Lost or delayed frames are substituted by the last frame that has been decoded properly. According to [4], this resembles a real world video player behavior. The quality of a transmitted video file is calculated using the Mean Opinion Score (MOS). The Mean Opinion Score is based on calculating the quality difference between the original and the received video using the PSNR metrics.The advantage of MOS is that it is a more representative metrics than pure PSNR. More information about the conversion from PSNR to MOS can be found in [4].

5 Simulation

The generated topology for all simulations contains 1000 nodes, distributed over 100 networks in 20 autonomous systems. The access speed for the nodes varies between 256 Kbit/s and 100 Mbit/s. The number of video files shared by each host is on average 500 (the typical number of files shared by a Gnutella host [3]). 40% of the files have a playback time of 120 minutes (typical length of a Hollywood movie), 30% have a playback time of 60 minutes and 30% of 30 minutes. The data-rate of the movies varies between 100 and 400 Kbit/s.

5.1 Peer-to-Peer Simulation

In order to quantize the effect Peer-to-Peers systems problems (section 2.2) on the resulting media quality we have simulated the behavior of such systems concerning online time and network loss in a 24 hours trace. For example an online time of 50% means that each peer has subsequent (normally distributed) online and offline times, resulting in a total up-time of 12 hours. The network loss values have been varied between 10% and 100% loss for each online level, yielding the following min, mean and max MOS results:

Online level in %	min MOS	mean MOS	max MOS
10	2.0014	2.06631	2.1115
20	2.0019	2.89885	3.3290
30	2.0019	3.00497	3.3174
40	2.0019	3.41421	4.2381
50	2.0019	3.41876	4.2570
60	2.0019	3.57989	4.6695
70	2.0019	3.67477	4.9593
80	2.0019	3.61779	4.7875
90	2.0019	3.68281	4.9995
100	2.0019	3.68281	4.9995

Considering that the online time of gnutella peers is on average between one and two hours every day (corresponding to at most 10% online time in our trace) it can be seen that the quality of the received streams in the scenario could be improved on average by 78% (2,06 vs. 3,68 MOS) and at most by 136% (2,11 vs. 4,99 MOS) simply by enforcing an online time of 100 %.

5.2 CDN Simulation

For Content Distribution Networks we have simulated the resulting media quality from varying the distance between the client and the closest surrogate server as well as the percentage of available content in an 24 hours trace. For example having an content availability of 50% means that 5 out of 10 requests are served by the surrogate server, for the others the content has to be replicated just in time and forwarded to the receiver. The min, mean and max MOS values from this variations in correspondence to the distance in networks elements are listed as follows:

Distance in NW elements	min MOS	mean MOS	max MOS
1	3.194000	3.678255	4.022800
2	3.194000	3.678255	4.022800
3	3.159125	3.740797	4.134100
4	3.058950	3.758410	4.193000
5	2.922800	3.758410	4.134100
6	2.723175	3.614535	4.020850
7	2.519875	3.306957	3.625225
8	2.309350	2.909415	3.146850
9	2.103275	2.471552	2.595375
10	1.869050	2.000920	2.040250

Assuming that the average distance between the nearest surrogate server and the client can be reduced from the average distance of 10 hops to 5 hops, a quality improvement of at least 56% (1,86 vs. 2,92 MOS), on average 87% (2,00 vs. 3,75 MOS) and at most 102% (2,04 vs. 4,13 MOS) can be achieved.

5.3 X2X Simulation

In order to evaluate our Proxy-to-Proxy approach we have made the assumption that the proxies have an availability of 100%, ignoring failure times. Content Distribution Network behavior has been simulated using Load Closeness (LC) and Semantical Closeness (SC) according to the dynamic parameter setting in equation 1 (section 3.3):

$$ProxyAffinity = \frac{LC}{SC} * LC + \frac{SC}{LC} * SC \qquad (6)$$

For simulating Peer-to-Peer behavior we have only used Network Closeness (NC), (see section 3.3):

$$ProxyAffinity = NC \qquad (7)$$

And Proxy-to-Proxy behavior (merged P2P and CDN) has been simulated using Load Closeness (LC), Semantical Closeness (SC) and Network Closeness (NC), (see section 3.3): $ProxyAffinity =$

$$\frac{LC}{SC + NC} * LC + \frac{SC}{LC + NC} * SC + \frac{NC}{LC + SC} * NC \qquad (8)$$

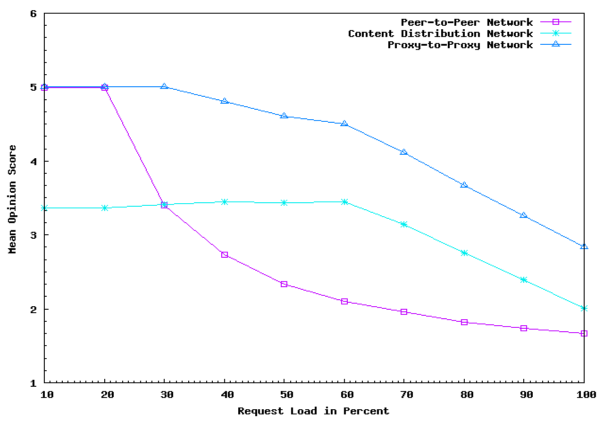

Figure 4: Proxy-to-Proxy Network

All three approaches have been compared against each other (Figure 4), the results yielding the worst, average and best quality are listed below:

Simulated behavior	min MOS	mean MOS	max MOS
P2P (equation 2)	1,668770	2,775338	5,000000
CDN (equation 3)	2,007025	3,075795	3,444350
X2X (equation 4)	2,841410	4,279561	5,000000

Building groups based on X2X or P2P behavior has been compared in the first experiment series. It can be seen that the min and mean MOS values of the X2X approach are 71% and 54% better than using the P2P approach. The highest MOS value achieved by both approaches is equal. Comparing X2X to CDN behavior it can be seen that the min MOS value of the X2X approach is 40%, the average value is 36% and the best value is 45% better than using the CDN approach.

5.4 Conclusion

We have analyzed the problems of Peer-to-Peer systems and Content Distribution Networks. Peer-to-Peer systems suffer from low availability (online time of individual peers) and have a content management problem concerning the relationship between shared content and required content. Content Distribution Networks are static and bound to surrogate server locations, sometimes having no possibility to avoid long network paths between the surrogate servers and clients. As a solution for these problems we have presented a system, called Proxy-to-Proxy being able to combine the highly dynamic but fragile Peer-to-Peer approach with the more robust but inflexible Content Distribution Network approach. The P2P,CDN and X2X approaches have been compared against each other concerning the quality of streamed multimedia content, using the Mean Opinion Score metrics. The emulation results show that the quality, using the X2X approach, is 71% and 45% better than quality achieved using the P2P or CDN approach.

References

[1] E. Adar and B. Huberman. Free riding on gnutella, October 2000.

[2] Qi He, Mostafa Ammar, George Riley, Himanshu Raj, and Richard Fujimoto. Mapping peer behavior to packet-level details: A framework for packet-level simulation of peer-to-peer systems. In *Mapping peer behavior to packet-level details: A framework for packet-level simulation of peer-to-peer systems*, pages 71–78, 2003.

[3] Shoaib Khan, Rüdiger Schollmeier, and Eckehard Steinbach. A performance comparison of multiple description video streaming in peer-to-peer and content delivery networks, presented at ieee 2004 international conference on multimedia and expo (icme2004), 2004.

[4] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid - a framework for video transmission and quality evaluation. In *Computer Performance Evaluation/Tools*, pages 255–272, 2003.

[5] J. Klensin. Rfc 3467 - role of the domain name system (dns), 2003. Available at: ftp://ftp.is.co.za/rfc/rfc3467.txt.

[6] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. Brite:an approach to universal topology generation. In *9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001*, pages 346–353, August 2001.

[7] B. Molina Moreno, C.E. Palau Salvador, M. Esteve Domingo, I. Alonso Pena, and V. Ruiz Extremera. On content delivery network implementation. In *Elsevier Science, Computer Communications, Volume 29, Issue 12*, pages 2396–2412, 2006.

[8] The Network Simulator ns-2 (v2.1b8a). http://www.isi.edu/nsnam/ns/, October 2001.

[9] Peter Schojer, Laszlo Böszörmenyi, and Hermann Hellwagner. Qbix-g – a transcoding multimedia proxy. In *Multimedia Computing and Networking*, pages 60710C–1 – 60710C–16, January 2006.

[10] Anthony Vetro and Christian Timmerer. N5845 - Text of ISO/IEC 21000-7 FCD - Part 7: Digital Item Adaptation. July 2003. http://www.chiariglione.org/mpeg/working_documents/ mpeg-21/dia/dia_fcd.zip.

Institute of Information Technology
University Klagenfurt
Universitaetsstr. 65-67
A-9020 Klagenfurt
Austria

http://www-itec.uni-klu.ac.at

UNIVERSITÄT KLAGENFURT

University Klagenfurt