Live Streaming Content on a Peer-to Peer Network.

[1]M.Sri Lakshmi, [2]M.Sampath Kumar, [3]Krishna Veni

[1]*DEPT of MCA,*
G.Pullaiah college of Engineering and technology,
Kurnool, Andhra Pradesh, India.
marri_srilakshmi67@yahoo.co.in

[2,3]*DEPT of CSE,*
Kottam College of Engineering,Kurnool ,
Andhra Pradesh, India.
[2]sampath.kumar_7@live.com
[3]krishnavenijntu@gmail.com

Abstract:Many peer-to-peer systems have been deployed on a large scale to provide users with "live streaming", that is Internet delivered real-time multimedia content, much as the traditional television service (to name a few, CoolStreaming,TVants and PPLive, UUSee are examples in case).All such systems are organized in an unstructured manner, with epidemic-style information exchanges: users decide who to interact with in an adaptive manner, on the basis of past experience, and decide which data blocks to exchange with their "logical neighbors" by relying on simple, local rules. Despite their success, the performance of such unstructured systems is still poorly understood, especially in comparison of structured systems, such as Split Stream, the performance of which is relatively well understood in symmetric scenarios. In this paper, In this paper we describe how cool streaming are used and We emphasize three salient features of this data-driven design: 1) easy to implement, as it does not have to construct and maintain a complex global structure; 2) efficient, as data forwarding is dynamically determined according to data availability while not restricted by specific directions; and 3) robust and resilient, as the partnerships enable adaptive and quick switching among multi-suppliers.

Keywords— **Coolstreaming, peer-to-peer technology (p2p), Live streaming, Zattoo, single overlay.**

INTRODUCTION

There is an emerging market for IPTV. Numerous commercial systems now offer services over the Internet that are similar to traditional over-the-air, cable, or satellite TV. Live television, time-shifted programming, and content-on demand are all presently available over the Internet. Increased broadband speed, growth of broadband subscription base, and improved video compression technologies have contributed to the emergence of these IPTV services. We draw a distinction between three uses of peer-to-peer (P2P) networks: delay tolerant file download of archival material, delay sensitive progressive download (or streaming) of archival material, and real-time live streaming. In the first case, the completion of download is elastic, depending on available bandwidth in the P2P network. The application buffer receives data as it trickles in and informs the user upon the completion of download. The user can then start playing back the file for viewing in the case of a video file. Bittorrent and variants are example of delay-tolerant file download systems.

Individuals use the Internet to express themselves through daily web-site logs, audio broadcasting, and web cams. However, live video, the most expressive form, is not currently possible due to the large bandwidth requirement. Clear, smooth video requires approximately 100 kbps of upstream speed. Even with a high speed DSL or cable connection of several hundred kbps, the source can only send the video to a couple of people.

Drawing from these measurements, we report on the operational scalability of Zattoo's live streaming system along several key issues:

1) How does the system scale in terms of overlay size and its effectiveness in utilizing peers' uplink bandwidth?

2) How responsive is the system during channel switching, for example, when compared to the 3-second channel switch time of satellite TV?

3) How effective is the packet retransmission scheme in allowing a peer to recover from transient congestion?

4) How effective is the receiver-based peer-division multiplexing scheme in delivering synchronized sub-streams?

5) How effective is the global bandwidth subsidy system in provisioning for flash crowd scenarios?

SYSTEM ARCHITECTURE:

The Zattoo system rebroadcasts live TV, captured from satellites, onto the Internet. The system carries each TV channel on a separate peer-to-peer delivery network and is not limited in the number of TV channels it can carry. Although a peer can freely switch from one TV channel to another, and thereby departing and joining different peer-to-peer networks, it can only join one peer-to-peer network at any one time.We henceforth limit our description of the Zattoo delivery network as it pertains to carrying one TV channel. Fig. 1 shows a typical setup of a single TV channel carried on the Zattoo network.

The encoding server may be physically separated from the server delivering the encoded content onto the Zattoo network. For ease of exposition, we will consider the two as logically co-located on an *Encoding Server*. Users are required to register themselves at the Zattoo website to download a free copy of the Zattoo player application. To receive the signal of a channel, the user first authenticates itself to the Zattoo *Authentication Server*. Upon authentication, the user is granted a ticket with limited lifetime. The user then presents this ticket, along with the identity of the TV channel of interest, to the Zattoo *Rendezvous Server*. If the ticket specifies that the user is authorized to receive signal of the said TV channel, the Rendezvous Server returns to the user a list of peers currently joined to the P2P network carrying the channel, together with a signed channel ticket. If the user is the first peer to join a channel, the list of peers it receives contain only the Encoding Server. The user joins the channel by contacting the peers returned by the Rendezvous Server, presenting its channel ticket, and obtaining the live stream of the channel from them

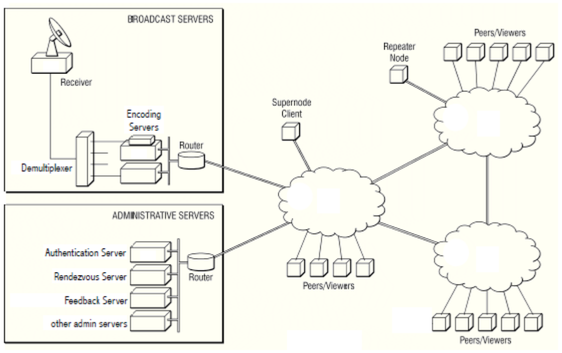

Fig. 1. Zattoo delivery network architecture.

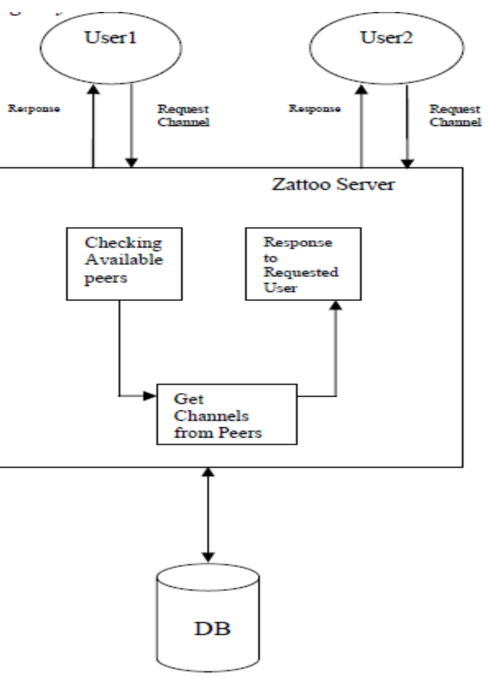

Fig 2. System Architecture

A. Why we use cool streaming Cool streaming is mainly used for:

- Easy to implement, as it does not have to construct and maintain a complex global structure.
- Efficient, as data forwarding is dynamically determined according to data availability while not restricted by specific directions.
- Robust and resilient, as the partnerships enable adaptive and quick switching among multi-suppliers.

B. Basic Components There are three basic modules in the system:

1) Membership manager, which maintains partial view of the overlay.

2) Partnership manager, which establishes and maintains TCP connections, or partnership, with other peer nodes. It also exchanges the availability of stream data in the buffer map (BM) with the peer nodes, which we will explain later.

3) Stream manager, which is the key component for data delivery. Besides providing stream data to the media player, it also makes decisions on where and how to retrieve stream data.The central design in this system is based on the datadriven notion, in which every peer node periodically exchanges its data availability information with a set of partners to retrieve unavailable data, while also supplying available data to others. Fig 3 shows how cool streaming is working

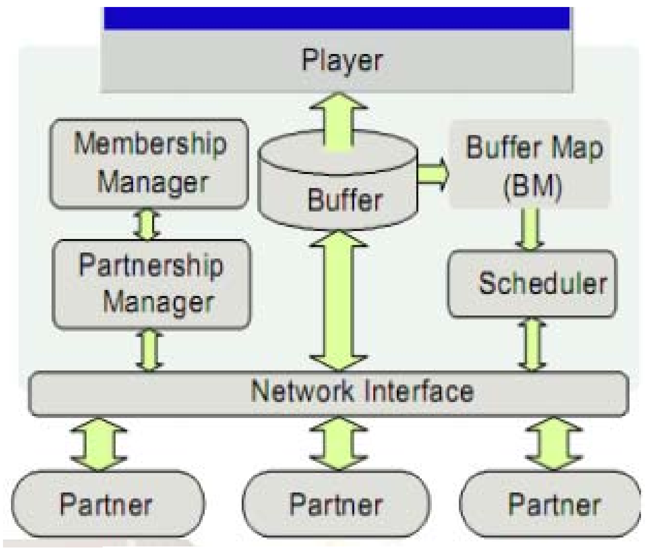

Fig. 3. Coolstreaming System Architecture

C. Multiple Sub streams

The video stream is divided into blocks with equal sizes, and each block is assigned a sequence number to represent its playback order in the stream. Since it is a live video streaming and we use TCP for transmissions, the sequence number also serves as a timestamp, which can be used to combine and reorder the blocks after reception. One of the key factors contributing to the success in P2P file sharing applications is the adoption of the gossip concept, in which a node can request different small chunks of file content from different nodes. This achieves significantly higher efficiency compared to other traditional systems. This is also adopted in cool streaming. Specifically, a video stream is divided into multiple sub streams; nodes could subscribe to different sub streams from different partners.

1) End-to-end bandwidth is a key problem in large scale streaming:

In a client/server system,We can support a large number of users in a local area by adding more servers. For a global scaled service, simply adding servers is not enough however. The end-to-end bandwidth may limit the geographical distribution of the users. Hence, CDN is a possible solution, but it remains very expensive and is not readily deployable. On the other hand, P2P enables intelligent path selections that may avoid the above problem. Nonetheless, we also find that, comparing with client-server solution, the overlay solution may introduce additional delay for a user to smoothly playback the video.

2) Upload bandwidth is a physical limitation: While P2P streaming is more flexible than client/server, its does have certain limitations. The most significant is the demands on the upload bandwidth. For a successful P2P media streaming system, at least we should have average upload bandwidth larger than the streaming rate. Hence, ADSL and other asymmetrical Internet accesses become obstacles.

3) ISP issues and traffic engineering effects: Currently, a large portion of the available bandwidth at network edges and backbone links are occupied by such P2P applications as BitTorrent. Many ISPs thus limit this kind of traffic. For file sharing the limitation may only be annoying (just slow down the download),but for media streaming it can be fatal. Other filtering mechanisms may also create problem to p2p streaming systems. We have already observed such problems in CoolStreaming.

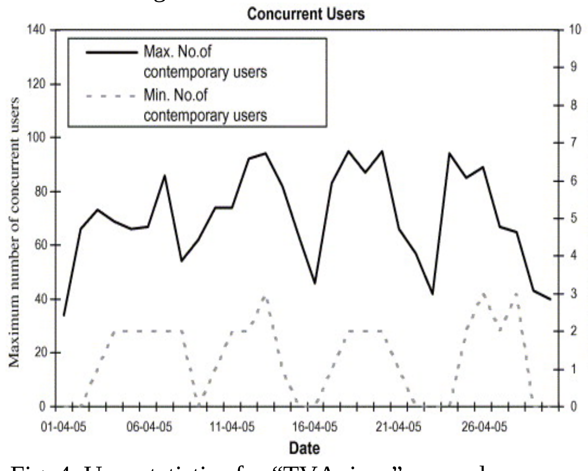

Fig. 4. User statistics for “TVAvisen” a popular news on demand service in Denmark.

D. *Tree-based Protocols and Extensions*

As mentioned previously, many overlay streaming systems employ a tree structure, stemmed from IP multicast. Constructing and maintaining an efficient distribution tree among the overlay nodes is a key issue to these systems. In Coop Net [3], the video source, as the root of the tree, collects the information of all the nodes for tree construction and maintenance. Such a centralized algorithm can be very efficient, but relies on a powerful and dedicated root node. To the contrary, distributed algorithms, such as Spread It, NICE [12], and ZIGZAG [11], perform the constructing and routing functions across a series of nodes. For a largescale network, these algorithms adopt hierarchical clustering to achieve minimized transmission delay (in terms of tree height) as well as bounded node workload. Still, an internal node in a tree has a higher load and its leave or crash often causes buffer underflow in a large population of descendants. Several tree repairing algorithms have been devised to accommodate node dynamics yet the tree tructure may still experience frequent breaks in the highly dynamic Internet environment.

E. Buffering

A buffer map or BM is introduced to represent the availability of the latest blocks of different sub streams in the buffer. This information also has to be exchanged periodically among partners in order to determine which sub stream to subscribe to. The detailed structure of the buffer map is as follows: BM is represented by a series 2K of -tuples, where K is the number of sub streams.

CONCLUSIONS

We proposed new system for live streaming using cool streaming. Cool streaming focused on the efficiency associated with multitree construction and also explored the simplicity and scalability adopted from unstructured networks. There are two points that we like to emphasize from this study: 1) A working system is essential in providing basic understanding; 2) There is a large number of practical problems that have to be dealt with in real engineering.

REFERENCES

1. S. Xie, B. Li, G. Y. Keung, and X. Zhang, “CoolStreaming: Design, Theory, and Practice,” IEEE Trans. on Multimedia, vol. 9, no. 8, December 2007.
2. Hyunseok chang, “live Streaming with receiver base peer division multiplexing”, in proc.IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 19, NO. 1, FEB 2011
3. S. Q. Zhuang, B. Y. Zhao, and A. D. Joseph, “Bayeux: An architecturefor scalable and faulttolerant wide-area data dissemination,” in Proc.NOSSDAV’01, New York, Jun. 2001.
4. D. Tran, K. Hua, and T. Do, “ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming,” in Proc. IEEE INFOCOM, 2003.
5. L. Guo, S. Chen, S. Ren, X. Chen, and S. Jiang, “PROP: a scalable and reliable P2P assisted proxy streaming system,” in Proc. ICDCS’04, Tokyo, Japan, Mar. 2004.
6. V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, “Distributing streaming media content using cooperative networking,” in Proc. NOSSDAV’02, USA, May 2002.
7. S. Jin and A. Bestavros, “Cache-and-relay streaming media delivery for asynchronous clients,” in Proc. International Workshop on Networked Group Communication (NGC’02),Boston, MA, USA, Oct. 2002.
8. W. Wang and B. Li, “Market-based self optimization for autonomic service overlay G.S.PAVANENDRA, V. Sneha latha / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp. 390-394 networks,” IEEE J. Sel. Areas Commun., vol. 23, no. 12, Dec. 2005.
9. M. Wang and B. Li, “Lava: A reality check of network coding in peer-to-peer live streaming,” in Proc. IEEE Infocom, May 2007.

10. Y. Guo, K. Suh, J. Kurose, and D. Towsley, “P2Cast: peer-to-peer patching scheme for VoD service,” in Proc. WWW’03, Budapest, Hungary, May 2003.
11. D. A. Tran, K. A. Hua, and T. T. Do, “A peer-topeer architecture for media streaming,” in IEEE J. Select. Areas in Comm., vol. 22, Jan. 2004.