VIDEO ON-DEMAND STREAMING ON THE INTERNET—A SURVEY

Xiangyang Zhang and Hossam Hassanein

Telecommunications Research Laboratory, School of Computing
Queen's University, Kingston, Ontario, Canada, K7L 3N6
{xiang,hossam}@cs.queensu.ca

ABSTRACT

Video-on-demand (VoD) streaming applications have evolved into one of the most popular types of applications on the Internet over the past decade. A great amount of research has been conducted to address the high bandwidth and stringent delay requirements of VoD applications on the video servers and the Internet. In this paper, we provide a broad survey of existing schemes and classify them into four categories. We also discuss the trade-offs and examine representative schemes in each category. Finally, we highlight future directions.

1. INTRODUCTION

Video on-demand (VoD) streaming applications have gained great popularity recently. Video news clips, full-length movies and TV shows, and videos made and shared by common people are watched by millions of people every day. Today, VoD streaming applications are generating a significant amount of traffic on the Internet. For example, during the inauguration ceremony of President Barack Obama on Jan 20, 2009, Akamai reported a peak of 7 million simultaneous streams destined to media companies such as NYTimes.com, Viacom, and WSJ.com [1].

A VoD streaming system consists of video servers, clients, and the network that connects the servers and clients. A client selects a video, usually by browsing a web site, and sends a request to a video server. The video server stores video files on its hard disk; it reads the video file into memory and delivers the requested video to the client via the network. The server may deliver at a rate different from the video's playback rate as long as packets arrive before their playback time. Each client plays the video at its own pace and may perform video cassette recorder (VCR) operations (e.g., seek new positions) during the playback.

VoD streaming applications exert enormous pressure on both the video servers and the network because of their high bandwidth and stringent delay requirements. The servers' throughput and the network traffic volume grow linearly with the number of clients. For example, to deliver a video encoded in MPEG4 with DVD-like quality to one million clients, the servers have to pump out 2 Tb/s of traffic. In comparison, video servers have throughput on the order of 100 Mb/s and Internet backbones have bandwidth on the order of 10 Gb/s.

2. APPROACHES TO VOD STREAMING

The primary objectives in designing a VoD system on the Internet is to minimize the network traffic and offload the video servers (i.e., reduce the upload requirement of the servers). To minimize network traffic, a video should be delivered to every requesting client using the shortest IP unicast path, and no link should carry more than one copy of the video. Since every requesting client will receive its own copy, to offload the video servers, other boxes are needed to replicate the video. There are four approaches to VoD streaming on the Internet: content delivery networks (CDNs), IP multicast-based schemes, application layer multicast-based peer-to-peer (P2P) networks, and swarming-based P2P networks. Table 1 summarizes some features of the four approaches.

Table 1. Approaches to VoD Streaming Systems

	Reducing load of	Caching video at	Services
CDNs	The Internet	Dedicated servers	True VoD, VCR
IP multicast-based schemes	Servers and the Internet	No caching	Near or true VoD, no VCR
Multicast-based P2P Networks	Servers and the Internet	Peers	True VoD, VCR
Swarming-based P2P Networks	Servers and the Internet	Peers	True VoD, VCR

CDNs extend the Client/Server (C/S) paradigm. Rather than one central server, CDNs have a large number of cache servers strategically placed across the Internet to serve nearby clients. CDNs do not reduce the number of the video servers needed, but they reduce the network traffic significantly.

IP multicast delivers the video to every client via the shortest path and the video traverses the links on the multicast tree only once. The video packets are replicated by the routers on the tree. However, IP multicast delivers the video in synchronicity to all the clients while VoD requires asynchronous delivery.

A P2P system is a distributed system in which a transient population of peers self-organizes into an *overlay network* on top of a *substrate network*, usually the Internet, to share computing power, storage, and communication bandwidth [2]. The focus of a P2P protocol is to organize peers in an efficient way suitable for the intended application. For VoD streaming applications, peers can be organized in two ways: an application layer multicast tree in which peers cache and forwards the video to their children, or a mesh network in which peers exchange video chunks with their neighbours. Both of the P2P approaches offload the video servers, and depending on the mapping of the overlay network to the substrate network, they can reduce the network traffic to some extent.

978-1-4244-5711-3/10/$26.00

3. CDNS

CDNs have been widely used to deliver web content and streaming media to a large population of users on the Internet since the late 1990s. CDNs use a large number of *cache servers* strategically placed across the Internet to cache popular data objects and serve nearby clients. When a client requests a specific data object, the request is redirected to a cache server close to the client. The cache server, if it has the data object, delivers the object to the client; otherwise, the cache server retrieves the data object from other servers for the client and keeps a copy locally. The placement of cache servers, the cache update policy, and servers and network monitoring are among the top issues in designing CDNs. Diley et al. [3] illustrate the architecture and design challenges of one of the leading CDNs, Akamai.

Since clients are served by nearby servers, and servers retrieve a video only once, CDNs significantly reduce the traffic on the Internet. When the servers are well placed and have sufficient bandwidth, the video playback quality is high and the latency is low. CDNs support a true VoD service and VCR operations because of their C/S nature. The biggest drawback of CDNs is the high cost. CDN operators not only invest heavily on servers, but also pay a large amount of money to Internet Service Providers (ISPs) for connecting their servers to the Internet.

4. IP MULTICAST-BASED SCHEMES

Much research has been conducted on using IP multicast for VoD streaming. Three categories of schemes have been proposed to bridge the gap between synchronous IP multicast and asynchronous VoD streaming: batching, periodic broadcasting, and patching. The main concern in these schemes is the server's output bandwidth, which also determines the network traffic volume. In all schemes, the servers' output bandwidth is divided into logical channels.

In the *batching* schemes such as [4], the server queues clients' requests for a video and uses an algorithm to admit a set of requests when a channel becomes available. Then it multicasts the entire video in the channel and informs the corresponding clients. Since clients' requests are not immediately granted, the batching schemes actually provide a near-VoD service, but not a true VoD service.

The *periodic broadcasting* schemes split a video into segments and broadcast each segment in a separate channel periodically. Clients wait for the beginning of the first segment, and download the data of the next segment while watching the current segment. Hu [5] classifies periodic broadcasting schemes into three kinds. Pyramid-like schemes such as [6,7] have increasing size segments and equal bandwidth channels. Harmonic-like schemes such as [8] have equal size segments and decreasing bandwidth channels. Hybrid schemes are a mixture of the above two kinds of schemes. For example, Pagoda [9] splits the video into fixed size segments and maps them into data streams of equal bandwidth, but broadcasts successive segments at decreasing frequencies. Although periodic broadcasting schemes also provide a near-VoD service, the wait time is significantly shorter than the batching schemes.

The *patching* schemes such as [10–13] provide a true VoD service. We take [11] as an example to illustrate how they work. The server's upload link is divided into logical channels with the bandwidth of the video's playback rate. A channel is either a regular channel in which the server multicasts the entire video or a patching channel in which the server multicasts only the leading part of the video. When a client requests a video from the server, the server instructs the client to download from a regular channel C_r and a patching channel C_p. Assume that channel C_r has proceeded to position t of the video at the time. The server multicasts the leading part of the video (i.e., from position 0 to position t) in the patching channel, which becomes free after the leading part has been delivered. The client downloads from the regular and the patching channels as instructed, playing the leading part of the video downloaded from the patching channel and buffering the content downloaded from the regular channel. The client exits the patching channel after it downloads the leading part of the video but remains in the regular channel until the end of the video. The *stream merging* scheme [14] share many similarities to the patching schemes. Each new client initiates a new channel. The download rate is higher than the video's playback rate, so a client can catch up to an earlier client. Then the two channels merge and one channel is freed.

In IP multicast-based schemes, the server's load and network traffic are determined by the number of channels, which is no longer linearly correlated to the number of users. However, the batching and periodic broadcasting schemes only support near-VoD services, and all the schemes do not support VCR operations. IP multicast-based schemes are also limited by the sparse deployment of IP multicast on the Internet.

5. MULTICAST-BASED P2P NETWORKS

All the IP multicast-schemes can be adapted to use application layer multicast. However, since peers can cache the video, true VoD services are easily achieved using application layer multicast, so the batching schemes and the periodic broadcasting schemes are no longer attractive. P2Cast [15] is a patching scheme using application layer multicast. Peers arriving within a certain length of interval and the server form a cluster. Peers in a cluster form an application layer multicast tree, called the *base tree*, rooted at the server. All the users download the video from the base tree. A late peer can fetch the missing parts from an early peer in the cluster. P2Cast does not support VCR operations.

All the other application layer multicast-based schemes can be classified as *asynchronous multicast* schemes in which peers form a multicast tree, and early peers cache the video and serve late peers. Some asynchronous multicast schemes do not support VCR operations, but this can be easily amended by adding a server that records all the peers' playback positions.

In Jin and Bestavros' *cache-and-relay* scheme [16], peers cache a moving window of the video they have watched. Peers form an application layer multicast tree according to their arrival time and positions on the network to minimize the total network distance of the tree using a heuristic greedy algorithm. A new peer queries a central server for nearby peers; the central server records all the peers' network positions. When a host fails, its children may contact the original server for a new parent.

DirectStream [17] uses a distributed directory, called AMDirectory, to facilitate tree-building and support VCR functions. AMDirectory's functionality is the same as a central directory server but is more scalable. The video servers and peers register with the AMDirectory; the entry for a peer includes its IP address, the time it joins, the position it currently plays, and its buffer size. All the peers watching a video cache a moving window of the most recent content. When a new peer u arrives, or seeks a new position, it queries the AMDirectory for a list of peers for potential parents. Early peers can be parent of late peers. Peer u then selects the peer from the list with the lowest $\frac{n^r}{x}$, where r is a constant, n is the number of hops, and x is the measured bandwidth between them.

P2VoD [18] organizes peers into generations according to their playback positions. A peer in generation $j+1$ has a parent in generation j. The video is split into blocks. At every peer in a generation, the oldest block cached is the same, and the newest block is the block the peer has just played. When a new peer arrives, it contacts the server for a list of peers in the previous generation as candidate parents. The new peer uses a certain algorithm such as round robin to select a parent from the list. When a peer v fails, its children first request peer v's siblings to be a parent; if unsuccessful, it resorts to the video server.

Several other schemes exist, using different tree-building algorithms or exploiting certain properties of VoD streaming. For example, oStream [19] first builds a directed graph according to peers' arrival time, then builds a multicast tree out of the graph using a distributed algorithm with the purpose of reducing the cost of the tree. dPAM [20] allows a peer to pre-fetch a portion of the video if its download rate is higher than the video's playback rate. CoopNet [21] uses the video server to build and manage the multicast tree; every peer queries the server to find a parent.

Multicast-based P2P networks are vulnerable to *peer churn* (i.e., peer turnover), especially the departure of peers close to the tree root. Because most peers' upload capacity can support a few peers, when the number of peers in the system is large, the application layer multicast tree may have such a large height that peers witness departures of ancestors every second.

6. SWARMING-BASED P2P NETWORKS

The swarming-based schemes are inspired by BitTorrent file sharing systems. Today, almost all P2P VoD streaming systems on the Internet use the swarming technique. In the swarming-based schemes as well as BitTorrent file sharing systems, the video file is split into fixed-size *piece*s. Every peer maintains a buffer storing video pieces and a *buffer map* describing the buffer. Each piece is represented by a bit in the buffer map, indicating whether the peer has the piece or not. A new peer contacts the *tracker* of the file for a list of peers to establish neighbourhood relationships. Each peer exchanges its buffer map with and pull missing pieces from its neighbours. Peers use a piece selection algorithm to decide which piece to request and from which neighbour to request, and an incentive policy to decide which request to honour. Usually the buffer is large enough to store several videos, and peers can continue uploading after they have finished watching the video.

However, swarming-based VoD systems are different from file sharing systems in three ways. (1) Once a peer started watching the video, every piece after the current playback position has a deadline, every piece before the current playback position is useless. (2) A peer may seek a new position during the playback. Once a peer seeks, pieces' deadlines and usefulness change. A side effect of seeking is that a peer does not necessarily need or store the whole file. (3) Usually the video server can be used as the last resort; it is always-on and has relatively large upload capacity.

Many swarming-based schemes have been proposed, such as [22–29]. Most schemes target the basic problems, while some schemes emphasize a specific aspect or address a specific situation. For example, [30] uses network coding to reduce network traffic, [29] attempts to discourage *free rider*s (peers that only download), and [27] uses always-on Set Top Boxes (STBs) with hard disks to cache videos. We restrict our introduction to three schemes due to limited space.

BASS [22] makes only two modifications to BitTorrent. First, BASS continues to use BitTorrent's *rarest-first* piece selection algorithm but ignores pieces prior to a peer's current playback position. Second, peers fetch from the server for pieces that are close to their playback deadline but cannot be obtained from other peers.

BiToS [23] modifies the piece selection algorithm for on-demand streaming. The state of a piece can be *downloaded*, *missed* (have missed or will miss the deadline so will not be downloaded), or *not-downloaded*. Not-downloaded pieces are classified into the *high priority set*, which has fixed size and contains pieces close to their playback deadline, and the *remaining set*. The piece selection algorithm selects a piece in the high priority set with probability p, selects a piece in the remaining set with probability $1-p$, and uses the rarest-first policy inside each set separately. The probability p can be adjusted by events such as the missing of a deadline. A not-downloaded piece can be *not-requested* or *currently-downloading*. A peer can have up to k currently-downloading pieces, where k is a system parameter. Peers continue to use the *tit-for-tat* incentive policy as in BitTorrent.

PPLive VoD [28] is a production system. As of January 2008, it has 150,000 simultaneously users and 500 movies online. The video playback rate is usually 381–450 kb/s. PPLive consists of a bootstrap server to help peers find a suitable tracker, a set of trackers to help peers to find other peers sharing a certain video, a set of video servers, log servers, etc. Each peer contributes 1 GB hard disk space to buffer videos; a peer may upload one video while downloading another or not downloading at all. A peer caches the video chunks it has played but does not pre-fetch video chunks because most peers stay online for less than an hour. When the buffer is full, a peer selects a video and removes all the chunks of that video. PPLive uses the *availability-to-demand* (ATD) ratio as the weight to the least recently used (LRU) algorithm to select which video to remove. If a video is viewed by n peers and cached (including partially cached) by c peers, then the ATD ratio is $max(\frac{c}{n}, 8)$, where 8 is chosen because it usually takes 8 uploading peers to satisfy a downloading peer. The ATD ratio is maintained by the tracker; a peer contacts the trackers for the ATD ratio of a certain video. Every peer reports to the tracker if it removes a video. PPLive's piece selection algorithm considers first the pieces close to their playback deadline, then the rarest pieces. A peer requests more from a neighbour if the neighbour has shorter response time.

Swarming-based P2P networks are robust against peer churn but have long start-up delays. Users' watching experience is highly related to the abundance of total upload capacity of peers. Some VoD systems adopt a hybrid structure: peers download from one another as in P2P networks; they can also download from dedicated servers that are organized as a CDN if some video pieces cannot be obtained from peers in time.

7. CONCLUSION

Many schemes have been proposed to minimize the network traffic generated by VoD applications and offload the video servers. IP multicast-based solutions achieve both objectives but are limited by the sparse deployment of IP multicast on the Internet. Multicast-based P2P networks utilize peers' upload capacity to offload the video servers but are vulnerable to peer churn. CDNs and swarming-based P2P networks are practical solutions. CDNs are capable of providing a high quality VoD service while incurring low network traffic, but require heavy investment. Swarming-based P2P networks have low cost, but more research are needed to achieve stable and high quality service. The combination of CDNs and swarming-based P2P networks have the potential to provide high quality service at a low cost.

8. REFERENCES

[1] "http://www.akamai.com," Accessed Nov. 2009.

[2] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.

[3] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Computing*, pp. 50–58, 2002.

[4] C. Aggarwal, J. Wolf, and P. Yu, "On optimal batching policies for video-on-demand storage servers," in *Proc. Int. Conf. on Multimedia Computing and Systems (ICMCS)*, 1996, pp. 253–258.

[5] A. Hu, "Video-on-demand broadcasting protocols: A comprehensive study," in *Proc. IEEE INFOCOM*, vol. 1, 2001, pp. 508–517.

[6] S. Viswanathan and T. Imielinski, "Pyramid broadcasting for video on demand service," in *Proc. Multimedia Computing and Networking (MMCN)*, 1995, pp. 66–77.

[7] K. A. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 4, pp. 89–100, 1997.

[8] L. S. Juhn and L. M. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 268–271, 1997.

[9] J. Paris, S. Carter, and D. Long, "A low bandwidth broadcasting protocol for video on demand," in *Proc. IEEE Int. Conf. on Computer Communications and Networks (ICCCN)*, 1998, pp. 690–697.

[10] S. Carter and D. Long, "Improving video-on-demand server efficiency through stream tapping," in *Proc. IEEE Int. Conf. on Computer Communications and Networks (ICCCN)*, 1997, pp. 200–207.

[11] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in *Proc. ACM Multimedia Int. Conf.*, 1998, pp. 191–200.

[12] Y. Cai, K. Hua, and K. Vu, "Optimizing patching performance," in *Proc. Multimedia Computing and Networking (MMCN)*, 1999.

[13] L. Gao and D. Towsley, "Supplying instantaneous video-on-demand services using controlled multicast," in *Proc. Int. Conf. on Multimedia Computing and Systems (ICMCS)*, vol. 130, 1999.

[14] D. Eager, M. Vernon, and J. Zahorjan, "Bandwidth skimming: A technique for cost-effective video-on-demand," in *Proc. Multimedia Computing and Networking (MMCN)*, 2000.

[15] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "P2Cast: peer-to-peer patching scheme for VoD service," in *Proc. Int. WWW Conf.*, 2003, pp. 301–309.

[16] S. Jin and A. Bestavros, "Cache-and-relay streaming media delivery for asynchronous clients," in *Proc. Int. Workshop on Networked Group Communication*, 2002.

[17] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "DirectStream: A directory-based peer-to-peer video streaming service," *Computer Communications*, vol. 31, no. 3, pp. 520–536, 2008.

[18] T. Do, K. Hua, and M. Tantaoui, "P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment," in *Proc. IEEE Int. Conf. on Communications*, vol. 3, 2004.

[19] Y. Cui, B. Li, and K. Nahrstedt, "oStream: Asynchronous streaming multicast in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 91–106, 2004.

[20] A. Sharma, A. Bestavros, and I. Matta, "dpam: a distributed prefetching protocol for scalable asynchronous multicast in P2P systems," in *Proc. IEEE INFOCOM*, vol. 2, 2005, pp. 1139–1150.

[21] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *Proc. Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2002, pp. 177–186.

[22] C. Dana, D. Li, D. Harrison, and C. N. Chuah, "BASS: BitTorrent assisted streaming system for video-on-demand," in *Proc. IEEE SPS Int. Workshop on Multimedia Signal Processing (MMSP)*, 2005, pp. 1–4.

[23] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing BitTorrent for supporting streaming applications," in *Proc. IEEE INFOCOM*, 2006, pp. 1–6.

[24] Y. Guo, S. Mathur, K. Ramaswamy, S. Yu, and B. Patel, "PONDER: Performance aware P2P video-on-demand service," in *Proc. IEEE GLOBECOM*, 2007, pp. 225–230.

[25] B. Cheng, X. Liu, Z. Zhang, and H. Jin, "A measurement study of a peer-to-peer video-on-demand system," in *Proc. Int. Workshop on Peer-to-Peer Systems*, 2007.

[26] J. Lv, X. Cheng, Q. Jiang, J. Ye, T. Zhang, I. Lin, and L. Wang, "LiveBT: Providing video-on-demand streaming service over BitTorrent systems," in *Proc. Int. Conf. on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2007, pp. 501–508.

[27] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, "Push-to-peer video-on-demand system: design and evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, p. 1706, 2007.

[28] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale P2P-VoD system," in *Proc. ACM SIGCOMM Conf. on Data Communication*, 2008, pp. 375–388.

[29] J. Mol, J. Pouwelse, M. Meulpolder, D. Epema, and H. Sips, "Give-to-get: Free-riding-resilient video-on-demand in P2P systems," in *Proc. Multimedia Computing and Networking (MMCN)*, 2008.

[30] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. R. Rodriguez, "Is high-quality VoD feasible using P2P swarming?" in *Proc. Int. WWW Conf.*, 2007, pp. 903–912.