Programme ANR VERSO

Projet VIPEER

Ingénierie du trafic vidéo en intradomaine basée sur les paradigmes du Pair à Pair

Décision n° 2009 VERSO 014 01 à 06

du 22 décembre 2009

T0 administratif = 15 Novembre 2009

T0 technique = 1er Janvier 2010

Livrable 4.1

State of the Art

Auteurs :

D. Alliez (NDS), J. Garnier (NDS), Z. Li (Telecom Bretagne), G. Simon (Telecom Bretagne, F. Albanese (Eurocom), P. Michiardi (Eurocom)

Compilé par :

J. GARNIER (NDS)

Décembre 2010

Telecom Bretagne ; Eurecom ; INRIA ; France Telecom ; NDS ; ENVIVIO

Résumé : *(15 lignes)*

VIPEER builds upon the collaboration between a traditional CDN and a peer-assisted CDN or "distributed CDN" (dCDN), i.e. an overlay controlled by the network operator using P2P paradigms. This document presents the state of the art in this area.

Keywords : Content Delivery Network, P2P, Content Centric Network, Peer Assisted

Table des matières

Table des figures

1 Terminology

Term	Definition
BGP	Border Gateway Protocol
CCN	Content Centric Network
CDN	Content Delivery Network
CS	Content Store
DHT	Distributed Hash Table
DVR	Digital Video Recorder
FIB	Forwarding Information Base
IS-IS	Intermediate System to Intermediate System
LRU	Least Recently Used
LSA	Link State Advertisement
MRU	Most Recently Used
OSPF	Open Shortest Path First
P2P	Peer-to-Peer
P4P	Proactive network Provider Participation for P2P
PAS	Peer Assisted
PIT	Pending Interest Table
POP	Point of Presence
QoE	Quality of Experience
STUN	Session Traversal Utilities for NAT
VCR	Video Recorder
VoD	Video on Demand

2 Content Distribution - State of the Art

The scalability of Internet video services is a major concern for many business actors. The current approach, which combines huge data-centers and massive data replication in the so-called content delivery network (CDN), is known to admit some limitations; consequently the research activity has flourished in this area. Most of the proposals (VoD, Catch-up TV) have to face with the scaling issue. For example, on YouTube more than 100 million videos are watched everyday [1], Hulu, a web video site, delivers about 18 million videos each day in Feb 2009 [2]. Consequently, many techniques have emerged to ease the scaling issue. Most of these techniques can be divided into two categories: content oriented networks [3, 4, 5, 6, 7, 8, 9] and peer-assisted content delivery network [10, 11, 12, 13, 2].

2.1 Content Delivery Network (CDN)

The main idea of CDNs is to decrease the load of the origin content server by serving clients from managed content caches that have been strategically placed close to the clients. This is achieved with the orchestrated operation of four sub-systems: delivery, distribution, redirection, and accounting [La-01]. The delivery system consists of a set of content caches, called edge servers, which deliver content replicas to end-users. Edge servers are usually located at network operator Points-of-Presence (POP). Determining the optimal placement of edge servers has received a lot of attention [40, 41, 42]. More recently, set-top-boxes have been considered as edge servers [43, 44, 45, 46]. The distribution system replicates content from the origin server to the edge servers in a consistent manner. The main question is what content to replicate. Selection of content in partial replication can be on the basis of heuristics, popularity [47], or sensitivity to QoS [48]. Another question is when to replicate selected content. The foremost approaches are cooperative push, where content is pushed from the origin server before request, and edge servers cooperate to optimize distribution [49], non-cooperative pull, where content is pulled from the origin server when there is a cache miss at the edge server [50], and cooperative pull, where content is also pulled from nearby edge servers on a cache miss [51]. For the case of box-powered CDN, previous works have focused on centralized push approaches where the contents are spread on boxes so that the availability of the maximal amount of data is ensured [46]. In the same context, the maximal size of a catalogue of data has also been theoretically addressed [44]. Finally, as boxes can crash, several works have developed techniques to maintain in the CDN a number of replicas proportional to their popularity as a countermeasure to crashes [52, 53]. The redirection system routes client requests to appropriate edge servers instead of the origin server. Typical request-routing mechanisms include Global Server Load Balancing (GSLB) [54], DNS-based request-routing [55], HTTP redirection [56], URL rewriting [57], and anycasting [58]. Finally, the accounting infrastructure maintains logs of client accesses, CDN servers' usage, and network statistics. This information is used for prediction, monitoring, and enhancement of end-to-end performance. Typical metrics are cache hits, origin server upload bandwidth, response latency, edge server utilization, packet loss, and proximity.

Most operational CDNs deployed over the Internet are developed and owned by commercial companies such as Akamai and Limelight Networks. In Akamai's CDN , content update is on demand, redirection uses DNS-based request-routing, whereas accounting is extensive with both internal and external measurements. Limelight Networks' CDN uses partial replication for content selection, non-cooperative pull for content replication, and on-demand content update. It also uses DNS-based request-routing for redirection. CDNs specific to IPTV and

VoD have been developed by companies such as Intracom Telecom and Minerva Networks. The entire delivery chain is fully managed and controlled by the CDN and service providers, constituting a "walled garden".

Recent work has demonstrated the need of a cooperation between content providers (through CDN) and ISP [59]. A joint system design between those two actors is needed to face the amount of content-centric traffic but the implementation and deployment of such optimization is still to be defined.

2.2 Peer-to-Peer (P2P) overlay for Video on Demand (VoD)

A different approach to deal with the scalability problem in VoD systems is to use P2P overlays where any peer can store a part of a video, then send and receive data to and from any other peer. This approach has recently become very popular [60]. However, the application of P2P technology to interactive VoD streaming is not yet a mature area of research [61]. The basic principle of P2P video streaming is as follows. A server storing a video that will be viewed by many users chooses to cut the video into several chunks, then send these chunks separately to some users and let them exchange data so that all users receive all chunks of the stream before they are played. Related works include proposals where peers self-organize into a structure [62], or, on the contrary, use epidemic diffusion through gossip algorithms to spread the chunks [63]. One may also distinguish between systems where the data are pushed from data owners to other peers [64] and pull-based systems where a peer has to explicitly request a missing chunk [65]. A peer may implement various strategies for selecting which chunks it should send and to which peer it should send this chunk. And also, a mechanism to manage the pool of peers (new peer and peer lost/exit) need to be implemented as it is more recurrent than set-top boxes crashes.

Recent works [66] attempted to provide a theoretical analysis of these strategies.

2.3 Hybrid CDN-P2P overlay

Recently, there have been attempts to combine CDN and P2P overlays to address the limitations of a single approach, e.g. Coral CDN [51]. More specifically for VoD and streaming systems, few works have recently emerged [53, 67, 68]. The seminal work for hybrid CDN-P2P streaming is [67]. The authors study how the upload capacity of peer nodes can be exploited in such architectures. The paper investigates different contribution policies for the peer population and the dynamics of the instance when no CDN assistance will be needed to serve clients that connect later in the streaming process. Similarly, the authors in [69] study how much involvement from the CDN side should be in the streaming process delivering a media presentation to a population of peer clients in order to provide some performance guarantees. The propagation properties in terms of data units of existing solutions are investigated in [44]. The authors show for the given architectures some of the dissemination. Other papers, e.g., [70], have proposed some mechanisms to implement such systems. Finally, recent papers have raised some new applicative problems related with P2P-CDN VoD systems, e.g. prefetching data for VCR functions [71].

2.4 Content Centric Network (CCN)

In content centric network (CCN) proposed in [14, 15, 9], authors suggest to reform the principle of internet data transmission pattern. All the data and requests are transferred based on content name but not IP address, since users are interested in the content itself but not in its location. During transmission, a data packet can be replicated and multi-casted simultaneously to several requests asking for the same data, so that the usage of each data packet is optimized to relieve the scaling issue.

In CCN, proxy alike routers are necessary to reduce the total traffic in the network. These routers are required to store data packets according to the least recently used (LRU) policy. So the request for certain data can be immediately satisfied by a router if the router has forwarded the same data, and the data is still in its memory. This mechanism fits the request for content with small data size (e.g. newspaper or 5 minutes long YouTube video). However, for service with large data size (e.g. high quality video), routers with huge storage capacity are essential to keep good performance of CCN, which will definitely degrade the deployability of it. Furthermore, the pure CCN given in [9] may not be able to support the applications as catch-up TV allowing viewers to watch their favorite broadcast TV programs within an expanded time window. Because a program generated several weeks ago may be requested by an end user, but at that time no server supports that program anymore and the buffer in CCN router is obviously not large enough to store all the live video streams generated in several weeks.

In CCN, communication is driven by the consumers of data. There is no communication link, or source and destination, but two kinds of packets: interest and data. A consumer broadcasts its interest packet over all available connections and a provider hosting the corresponding data satisfies the interest by responding a data packet.

2.4.1 Hierarchically Named Content

The ContentName of both interest and data packets are explicitly defined in [9] so that the data can be easily found and match the interest. Because of the successful experience of IP network layer, the authors use a hierarchical structure of name space, which is similar to the structure of IP address. A typical example is a name as */parc.com/videos/WidgetA.mpg/_v<timestamp>/_s3*. When the ContentName of an interest packet is the prefix of the ContentName in the data packet, the consumer is satisfied by that data.

2.4.2 Routing Scheme

The principle of CCN router and IP router is very alike. A longest-match lookup of ContentName is done when a packet arrives on a *face* (face is a term used in [9]. The authors use it because packets in CCN are not only forwarded over hardware network interfaces but also exchanged directly with application processes within a machine.), then different actions is taken based on the lookup results. There are three main data structures in a CCN router: Forwarding Information Base (FIB), Content Store (CS) and PIT (Pending Interest Table). FIB is used to forward Interest packets toward potential data source. CS has the same function as IP buffer memory, but its *forgot* policy is different from IP. In order to minimize upstream bandwidth demand and downstream latency, CCN replaces most recently used (MRU) policy in IP by least recently used (LRU) policy. This is the most important improvement of CCN that makes it more efficient than IP network, especially when the number of consumers is large.

When an interest arrives at a CCN router, a longest-match lookup executes following the priority that CS match is the highest and FIB match is the lowest. If a CS match is realized, then the corresponding data in the ContentStore can be directly sent out the face the interest arrived and the interest is discarded. If there is a PIT match, the interest arrival face is added to the PIT requesting list and the interest is discarded. Otherwise, if it is a FIB match, the interest is sent out all the faces, which can potentially offer the corresponding data. Then a new PIT entry is created from the interest and its arrival face. If no match is found, the interest is discarded. The treatment of a data packet is relatively simple. Only when there is a PIT match, the data packet is sent out over all the faces in the requesting list. Otherwise, the packet is discarded, since it is either duplicated or unsolicited.

Because CCN use the same forwarding model as IP network, any routing protocol that works well for IP should also have a good performance in CCN. Therefore current Link-state routing protocols such as IS-IS or OSPF can well satisfy CCN intra-domain routing. When a CCN router received an announcement from a provider saying that the provider can offer data with a certain prefix, the router installs a local CCN FIB entry for the prefix pointing at the face where it heard the announcement, and packages the prefix into link state advertisement (LSA) and floods the LSA to all nodes. When another CCN router receives the LSA for the first time, it creates a CCN FIB entry for the prefix pointing at the router that sends the LSA. Then an interest can be forwarded to the content provider following FIB entries. The same scheme can be implemented on inter-domain level by BGP.

2.4.3 Advantages and drawbacks

The first improvement of CCN comparing with current CDN is that the content routing process is simpler. In CCN only one round-trip instead of three is needed for client to access content provider, which makes the routing overhead decrease. The anycast nature of CCN routing automatically solve the server selection problem. Furthermore, the replication and multi-cast of data packet can reduce the service rejection rate.

As there is no communication links in CCN, but interest and data, CCN is intrinsically against the attacks toward communication links. Moreover, since there is no source and destination, it is also difficult to attack a particular object in CCN. The network is secure as long as the data content is well encrypted. Therefore, the security issue of CDN is resolved without complicated management.

Thanks to the proxy like function of CCN router, the usage of a data packet is optimized and the traffic in the network is minimized. As the example given in [9], a popular YouTube video will traverse the link between youtube.com and its ISP millions of times in current network. But in CCN just one transmission were necessary since the video were stored in the closest CCN routers to consumers.

In fact, the buffer function of CCN router is similar to the dynamic content storage of CDN edge server [27, 28]. We confess that the buffer function fits the request for the content with small size. But for the content with large size, huge storage capacity of CCN router is required, otherwise the data in the buffer will be frequently updated, which will definitely degrade the performance of CCN. However, installing routers with great capacity all over the internet may frustrate the deployment of CCN. Therefore, the memory of CCN router may not large enough to support its good performance. Moreover, since the CDN edge server, which is only responsible for the delivery of content, faces the scaling problem, CCN router, which has the additional workload of forwarding packets, may suffer from the same problem. One the CCN drawback is this concept is currently defined and then there is no CCN solution of the shelf.

2.5 Peer assisted (PAS) CDN

Peer-assisted CDN improves scalability of the existed content distribution by merging P2P technology and CDN. End users are organized in a P2P pattern, so that idle resources of clients are used to alleviate the workload of CDN server. Therefore, more clients can be served by CDN.

Peer-assisted CDN suffers from problems caused by DNS redirection. Moreover, during the initial stage of P2P organization the system still shows high service reject rate since the number of peers is not enough to offer sufficient contribution [12, 13].

2.5.1 Globule: a PAS-CDN for Web service

Globule is an open source module of a collaborative CDN. Although the authors of [16] call it a collaborative CDN, we regard it as a PAS-CDN since end users of Globule are organized in a P2P fashion to contribute their resources. In this section, we introduce Globule from several crucial aspects of PAS-CDN: replica placement, client redirection and content availability.

Replica placement: To decide an optimal replica placement, the first task is to define a cost metrics in the network. In Globule, the authors take internodes latency as their proximity measure. Latency measurements are totally transparent to clients. When a Web browser accesses a Web server, the browser is requested to download a small image from landmarks assigned by the system. The latency between end user and landmark is measured during the TCP connection phase, then it is sent back to the origin server and an M-dimension coordinates of the user is calculated by the method given in [17] and [18]. After that the location of end users for a specific web site is determined, the authors partition the space into cells with identical size and ranking the cells according to the number of users that each of them contains. Finally, they put the replicas to the users in top k ranked cells.

Client redirection: Globule supports HTTP and DNS redirection. In [16], the authors use DNS redirection. So the principle of client redirection of Globule is the same as traditional CDN systems.

Content availability: Globule guarantees the availability of both origin server and replica server (end user hosting a service). For origin server, the authors propose using backup servers. To make sure that clients are always redirected to alive replica servers, redirectors are required to periodically probe the availability of replica servers.

2.5.2 PAS-CDN for VoD

Hybrid CDN-P2P architecture for VoD system is studied in [10, 12, 13, 2]. In [12] and [13], the authors focus on the system architecture and service process, and their propositions are quite similar. In the system of [12] there are three elements: directory server performs as a tracker; streaming server is responsible for streaming delivery; peer is end user machines participating in the streaming system. The service of each video stream experiences three stages. The initial stage begins when a video is pushed to a server. During the initial stage, requests of end users are served only by CDN server. With the increment of requesting peers, CDN server reaches its bandwidth limitation. Then the service turns to a CDN-P2P stage. CDN server selects a sub set of peers holding the video to be the complementary suppliers. After a large number of peers have downloaded the video, the contribution of supplying peers can satisfy the playback rate of requesting peers. Then the service enters a pure P2P stage for the video. The video in the CDN server is finally replaced by another one, and the CDN server works only as a tracker of the original video.

In [13], the authors integrate tracker function and streaming server into CDN servers. The service of each video is also divided into the same phases as in [12]. The difference is that in [13] each peer has its contribution commitment. After the commitment is fulfilled, a peer becomes retired peer and does not upload the video anymore. This configuration eases the problem of peer overloading and unfairness among peers. One important metrics to evaluate these PAS-CDN systems is the request rejection rate. If the rejection rate is low, then the scaling issue of VoD service is well solved. Unfortunately, both of the system in [12] and [13] still suffer from high rejection rate during the initial and CDN-P2P stage, especially many peers request a video simultaneously at the beginning of the service. The rejection rate is low only after the handoff to pure P2P stage.

Besides the research on system architecture, [10] and [2] study the PAS-CDN system from other angles. The authors of [10] highlight the potential benefits of implementing prefetching policies in PAS-CDN. No prefetching means that a peer downloads a video stream only when the peer needs it, otherwise, in prefetching system each peer pre-downloads some streams for further use. The authors assume two modes of in PAS-CDN system: 1) surplus mode, where total service capacity of peers is higher than total demand; 2) deficit mode, where total service capacity cannot afford total demand in the system. In their simulation, they show that in a totally surplus or deficit mode, where service capacity is much higher or lower than demand, no prefetching system works well and CDN server load keeps low. But when service capacity is similar to total demand - so the system may change between surplus mode and deficit mode frequently due to peer churn - the workload of CDN server increases dramatically as the service scaling up. Therefore two prefetching policies, water-leveling and greedy, are proposed by the authors to solve the previous scaling issue.

On the other hand, [2] improves the PAS-CDN from a business point of view. Although managed PAS-CDN system using ALTO or P4P protocol works well theoretically, they need peer information to construct a controlled environment.

2.5.3 PAS-CDN for live streaming

In this section, we introduce the architecture of a real-world CDN-P2P live video streaming system called LiveSky given in [19], which has been deployed in China. The system is designed to solve a set of problem in current CDN and P2P live video streaming systems such as scaling, fast startup and upload fairness. It is composed by three modules:

1) management module;

2) cache servers module;

3) clients module;

which is organized in an hybrid client-to-server and peer-to-peer model.

Server Side Organization. The CDN overlay is constructed on a tree based structure. It consists of several tiers: Servers in tier n - 1 are edge servers; other servers are core servers. Core servers are responsible for deliver content to edge servers. In order to provide better reliability, a core server is allowed to retrieve content from servers in up tiers and also the servers in its same tier (server side P2P). The task of edge servers is to serve end users. Considering the work load of edge servers, they are not allowed to transfer content between each other. To realize a P2P organization at client side, an edge server has several roles:

1) a regular server for legacy clients;

2) a tracker for the P2P operation;

3) a seed in the P2P system.

Client Side Organization. There are two types of clients: legacy clients and P2P clients. Legacy clients are served in the traditional CDN manner and receive low quality streams. P2P clients are organized in a hybrid scheme proposed in [20, 21] that combines the multi-tree and mesh schemes. As usual a video is divided into several sub streams. Each sub stream contains inconsecutive frames. The peers that host a same sub stream compose a tree-based overlay. This ensures the upload fairness of each peer. On the other hand, peers also use a mesh-style pull mechanism to retrieve missing frames for continuous playback. This enhances the robustness of the network. Moreover, P2P clients are allowed to access to high quality videos.

Adaptive Scaling and Improvements. In the system each edge server decides whether a new arrival client should be treated as a legacy client or a P2P client independently. A threshold is pre-configured in every edge server. When the number of clients is below the threshold, all clients retrieve the content directly form edge server. If the number of clients exceeds the threshold, new arrival clients will be redirected to other clients to form a P2P organization. Both of the threshold and the capacity of an edge server is calculated by some parameters including: level of the P2P tree overlay, peer arrival rate, peer leaving rate and peer contribution rate. When an edge server reaches its capacity limitation, new clients will be redirected to another less loaded edge server.

Fast startup. Problem in P2P streaming system is optimized in LiveSky in two ways. First, the buffer size is reduced to 15 seconds. Second, the first request of a client is always replied directly by an edge server, thus it is very quick to retrieve startup streams. For upload fairness besides using upload bandwidth restriction mechanisms, LiveSky also adopt techniques such as STUN [22] to ensure the contribution of the clients behind NAT.

Evaluation. The performance is evaluated by scaling adaptation and QoE in real-world deployment. In deployment the authors use edge servers with 200Mbps bandwidth capacity. If the media translation rate is 400Kbps, each edge server can afford 1000 end users. For QoE, they measure several aspects including: startup delay, re-buffering dynamics and stability. They show that LiveSky can provide a convincing performance.

2.5.4 Advantages and Shortcomings

P2P organization of end users can significantly lower the workload of edge server by sharing the delivery task. In other words, peers contribute their upload bandwidth so that the system can serve more requests, since edge server has limited service bandwidth. Moreover, peer-assisted CDN takes advantages of idle resources in the system, thus the cost-effectiveness of service provider increases.

As pure CDN system, peer-assisted CDN also uses DNS-based redirection when an end user first accesses to the network [16]. In DNS redirection, a request needs at most three round-trips to access the content. First, the client sends its request to a local DNS server and, the local DNS server should access the DNS root server to obtain the address of the authoritative name server of the content provider. Then, the local DNS server has to ask the name server and CDN provider to send back the address of a nearby content server. Finally, it incurs the round-trip for the client to access the content on the designated server. Therefore, the current routing scheme may cause higher round-trip times for request redirection than the round-trip times to access the content, if the client, DNS server and the name server are located faraway from each other.

Besides the long round-trip overhead, DNS redirection leads to other side-effects in CDN. The most critical problem is the mis-designation of edge server. In DNS redirection, the location of local DNS server is regarded as the originated location of a request [23]. However, end user is not always close to its local DNS server [24, 11]. Consequently, the end user is redirected to a remote edge server and causes an inefficient transaction. In P2P organization, the mis-designation of edge server brings about high cross AS traffic between supplying peer and requesting peer. One way to solve the problem is the recently proposed ALTO/P4P framework [20, 25], which further adds the complexity of the network

Furthermore, peer-assisted CDN still suffers from high service rejection rate during the initial and peer-assisted stage because of the limited server bandwidth and lack of peer resources [13, 12]. Finally, concerning security issue, malicious users can intentionally stick to a particular edge server, which can cause significant degradation of system performance [26].

2.6 Catch-up TV

The delivery of television over the Internet is expected to offer viewers new ways to enjoy TV content. One of the most promising services, often called catch-up TV or time-shifted TV, consists in allowing viewers to watch their favorite broadcast TV programs within an expanded time window. Let's say that a program is normally broadcasted from a given time t. In a catch-up TV, this program is made available for viewing at any time from t to t + δ hours where δ can be excessively long (several weeks). In this context, a viewer is also able to surf the TV content history using pause, rewind or fast forward commands, hence he/she can switch from a live experience to a shifted one. The delivery of the live content (multicast, peer-to-peer, etc.) is out of the scope of this chapter. Rather, we focus on the issue of serving a large number of clients requesting past portions of the stream.

2.6.1 Current solutions

Today, to enjoy catch-up TV requires to record the stream on a Digital Video Recorder (DVR) connected to Internet. Of course, this is unacceptable for TV providers, which would like to control the delivery of their content. However, building a large-scale time-shifted streaming service is not trivial. Indeed, the disk-based servers that are currently used in on-demand video services (VoD) have not been designed for concurrent read and write operations. In particular, a VoD server can not massively ingest content. Moreover, delivery systems for IPTV can not be utilized because, contrarily to live streaming systems, time-shifted systems can not directly use group communication techniques like multicast protocols, for the reason that clients require distinct portions of the stream. Several works have highlighted the problems met by classic centralized architectures [29, 30]. New server implementations are described in [31]. Cache replication and placement schemes are extensively studied by the authors of [29]. When several clients share the same optical Internet access, a patching technique described in [30] is used to handle several concurrent requests, so that the server requirement is reduced.

Some works have recently sketched a peer-to-peer architecture for time-shifted TV systems. In [32], every client stores all downloaded video parts. A Distributed Hash Table (DHT) is used to keep trace of the owner of every video part, so that a peer that is able to upload a past video part can be found upon a simple request to the DHT. Similarly, a DHT is used to locate video parts in [33]. Some additional proxies aim to ensure the continuity of some video that are locally downloaded. However, these works appear to suffer from critical drawbacks. First, the use of the hash function seems irrelevant in this context where chunks are iteratively produced. A structure that takes into account the stream linearity would be more appropriate. Second, a peer departure should conduct to multiple deletions in the DHT. For peers that store vast amounts of chunks in catch-up TV, a huge number of messages should be generated. Furthermore, the DHT could not guarantee the availability of all chunks, particularly for early and unpopular chunks.

Finally, in our previous work, we have explored another approach where the stream provider uses a peer-assisted system to ensure the content delivery [34]. Here, clients store the content that has been consumed, and may serve other viewers requesting this content. This system based on a tracker reduces significantly the traffic at the server side. However, this architecture still requires a server for data backup.

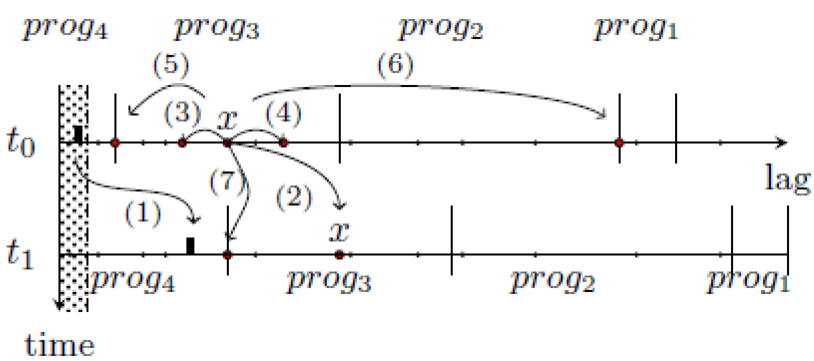

Figure 1 Use cases of time-shifted IP¨TV system

2.6.2 Model

We distinguish the stream source, which continuously generates the content, and the stream provider, which is responsible both to continuously store the whole stream in a persistent manner, and to serve clients issuing requests on any past stream portion.

Figure 1 depicts use cases of catch-up TV. The y-axis represents the playing time of the video source, and the x-axis shows the time lag distribution of playing positions with the source at a particular time. The shaded area denotes the time range for the live streaming service. Peers in this area are synchronous with the source, but with a small time lag because of the buffer mechanism and network delay. The small black points denote the playing positions of peers in the system, and the black rectangle represents a data chunk.

In an idle context, peers and source are shifting in the same direction with a uniform velocity. For the scenario of a shifting from time t0 to t1, the time lags between peers and the source are constant. From the viewpoint of chunks in case (1), chunks move along x-axis at the same speed as the source that continues outputting new content. Therefore, between time t0 and time t1, a chunk moves *t1 - t0* far from the stream producer.

In television, the stream is basically cut into successive programs (news, movie, sport, etc.). In this example, we represent four programs (from prog1 to prog4). From time t0 to t1, all programs move *t1 - t0* far from the source. We detail the main events that are possible for a peer x in a catch-up system. They are usually referred to as VCR operations.

Pause: it occurs when a user leaves for a moment, and is expected to resume streaming later from this current position. This is represented by case (2) in Figure 1. If the peer x performs a pause at time t0 and continues playback at time t1, the lag between x and the source will increase by *t1 - t0*. This operation is frequently implemented in current live streaming systems. In these systems, x continues to download the fresh content and buffers it.

Forward and Backward: a viewer in catch-up TV can perform forward or backward in a program, as depicted in cases (3) and (4), or between different programs in (5) and (6). We distinguish these two scenarios because both start and end times of a program are special points where the behavior of clients can be very different from other stream points.

Churn: a peer x may join the system as a live client, but it can also immediately start at a past position. As in other peer-to-peer systems, a peer should be assumed to be able to leave at any time, sometimes abruptly. We highlight however in the case (7) that it is also more probable that peer leaves at the end of a program, as shown in studies [35].

2.6.3 Catch-up TV versus VoD

Peer-to-peer systems have been abundantly explored for VoD systems. We describe now the characteristics of catch-up TV that differ from the VoD context.

Beyond the obvious differences of the content itself (the length of a catch-up TV stream is several orders of magnitude longer than a typical movie in VoD) and beyond the aforementioned inability of current VoD servers to both ingest and deliver a stream at large-scale, a critical difference has to be highlighted: the dynamicity of chunk request. In [36], it is shown that a quarter of shifters have a stream lag that is less than one hour, around 40% of them watch their program less than 3 hours after the live program, and more than half of shifters are enjoying a program that has been broadcasted the same day. On the contrary, the popularity of chunks is static in a VoD system. A VoD service provider can pre-determine the amount of upload capacity it should reserve in order to serve each chunk to all clients. In catch-up TV, the popularity of every chunk is variable, so the upload capacity management should constantly be adjusted.

Another difference is that viewers do not exhibit a standard playing behavior. In [37], a peak has been identified at the beginning of each program, where many viewers start streaming the content. Then, as can be also stated in VoD systems, the spikes of departure have been shown to occur mostly either at the end of the program, or because the user does not find any interest after browsing the beginning of the program. That is, peers usually leave immediately or simultaneously at the end of programs in [35]. Moreover, a large number of sessions end in the first minute, which means that these clients are not interested in the programs after browsing through the beginning. In this study, it is shown that more than half of the population quits during the first ten minutes of a program in average. This behavior makes that a casual user of time-shifting system is interested in a few sets of continuous chunks that can be far from each other. Although a few papers have recently addressed the VCR problem in peer-to-peer VoD systems [38, 39], no previous work has actually assumed that it is a massive behavior of users because it does not make sense in a VoD system. It is also worth saying that VoD user behavior will probably be different than Catch-up TV users.

3 Bibliography

[1] X. Cheng, C. Dale, J. Liu, Statistics and social network of youtube videos, in: Processding of IEEE IWQoS, 2008.

[2] T. Mori, N. Kamiyama, S. Harada, H. Hasegawa, R. Kawahara, Improving deployability of peer-assisted cdn platform with incentive, in: Globalcom, 2009.

[3] M. Gritter, D. R. Cheriton, An architecture for content routing support in the internet, in: Usenix Symposium on Internet Technologies and Systems (USITS), 2001.

[4] H. Balakrishnan, S. Shenker, M. Wal_sh, Semantic-free referencing in linked distributed systems, in: IPTPS, 2003.

[5] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, M. Wal_sh, A layered naming architecture for the internet, in: SIGCOMM, 2004.

[6] M. Wal_sh, H. Balakrishnan, S. Shenker, Untangling the web from dns, in: NSDI, 2004.

[7] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, V. Vercellone, Design consideration for a network of information, in: Proceeding of ACM CoNEXT, 2008.

[8] C. Esteve, F. L. Verdi, M. F. Magalhaes, Towards a new generation of information-oriented internetworking architectures, in: Proceeding of ACM CoNEXT, 2008.

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking named content, in: Proceedings of the 5th international conference on emerging networking experiments and technologies, 2009.

[10] C. Huang, J. Li, K. W. Ross, Peer-assisted vod: Making internet video distribution cheap, in: Proceeding of IPTPS, 2007.

[11] C. Huang, J. Li, K. W. Ross, Can internet video-on-demand be pro_table?, in: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, 2007.

[12] Y.-C. Tu, J. Sun, M. Hefeeda, S. Prabhakar, An analytical study of peer-to-peer media streaming system, ACM Transactions on Multimedia Computing, Communications, and Applications 1 (4).

[13] D. Xu, S. S. Kulkarni, C. Rosenberg, H.-K. Chai, Analysis of a cdn-p2p hybrid architecture for cost-effective streaming media distribution, ACM/Springer Multimedia Systems 111 (4).

[14] V. Jacobson, A new way to look at networking, http://video.google.com/videoplay?docid=-6972678839686672840&ei=iUx3SajYAZPiqQLwjIS7B&q=tech+talks+van+jacobson. (2006).

[15] V. Jacobson, Making the case for content-centric networking: An interview with van jacobson, acm queue (2009).

[16] G. Pierre, M. van Steen, Globule: A collaborative content delivery network, IEEE Communications Magazine 44 (8).

[17] T. Ng, H. Zhang, Predicting internet network distance with coordinates-based approaches, in: IEEE INFOCOM Conference, 2002.

[18] F. Dabek, R. Cox, F. Kaashoek, R. Morris, Vivaldi: A decentralized network coordinate system, in: ACM SIGCOMM Conference, 2004.

[19] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, B. Li, Design and deployment of a hybrid cdn-p2p system for live video streaming: Experiences with livesky, in: Proceedings of ACM Multimedia, 2009.

[20] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, A. Silberschatz, P4p: Provider portal for applications, ACM SIGCOMM Computer Communication Review 38 (4).

[21] L. Z. Meng Zhang, Jian-Guang Luo, S.-Q. Yang, A peer-to-peer network for live media streaming - using a push-pull approach, in: Proceedings of ACM International Conference on Multimedia, 2005.

[22] J. Rosenberg, J.Weinberger, C. Huitema, R. Mahy, Stun simple traversal of user datagram protocol (udp) through network address translators (nats), IETF RFC 3489.

[23] A. Barbir, B. Cain, R. Nair, O. Spatscheck, Known cn request-routing mechanisms, RFC 3568.

[24] Z. M. Mao, C. D. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, J. Wang, A precise and e_cient evaluation of the proximity between web clients and their local dns servers, in: USENIX Annual Technical Conference, 2002.

[25] R. Alimi, R. Penno, Y. Yang, Alto protocol, http://64.170.98.42/pdf/draft-ietf-alto-protocol-02.pdf.

[26] A. jan Su, A. Kuzmanovic, Thinning akamai, in: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, 2008.

[27] K. Stamos, G. Pallis, C. Thomos, A. Vakali, A similarity based approach for integrated web caching and content replication, in: Proceedings of 10th International Databased Engineering and Applications Symposium, 2006.

[28] A.-M. K. Pathan, R. Buyya, A taxonomy and survey of content delivery networks, Technical Report University of Melbourne.

[29] J. Zhuo, J. Li, G.Wu, S. Xu, E_cient cache placement scheme for clustered time-shifted TV servers, IEEE Transactions on Consumer Electronics 54 (4) (2008) 1947-1955.

[30] W. Xiang, G. Wu, Q. Ling, L. Wang, Piecewise Patching for Time-shifted TV Over HFC Networks, IEEE Transactions on Consumer Electronics 53 (3) (2007) 891-897.

[31] C. Huang, C. Zhu, Y. Li, D. Ye, Dedicated Disk I/O Strategies for IPTV Live Streaming Servers Supporting Timeshift Functions, in: Proc. of IEEE CIT, 2007, pp. 333-338.

[32] F. V. Hecht, T. Bocek, C. Morariu, D. Hausheer, B. Stiller, LiveShift: Peer-to-Peer Live Streaming with Distributed Time-Shifting, in: Proc. of 8th Int. P2P Conf., 2008, pp. 187-188.

[33] D. Gallo, C. Miers, V. Coroama, T. Carvalho, V. Souza, P. Karlsson, A Multimedia Delivery Architecture for IPTV with P2P-Based Time-Shift Support, in: Proc. of 6th IEEE CCNC, 2009, pp. 1-2.

[34] Y. Liu, G. Simon, Peer-to-peer time-shifted streaming systems, CoRRabs/0911.1226.

[35] X. Hei, C. Liang, J. Liang, Y. Liu, K. W. Ross, A Measurement Study of a Large-Scale P2P IPTV System, IEEE Transactions on Multimedia 9 (8) (Dec. 2007) 1672-1687.

[36] Nielsen, How DVRs Are Changing the Television Landscape, Tech. rep.,Nielsen Company (April 2009).

[37] T. Wauters, W. V. de Meerssche, F. D. Turck, B. Dhoedt, P. Demeester, T. V. Caenegem, E. Six, Management of Time-Shifted IPTV Services through Transparent Proxy Deployment, in: Proc. of IEEE Globecom, 2006, pp. 1-5.

[38] X. Yang, M. Gjoka, P. Chhabra, A. Markopoulou, P. Rodriguez, Kangaroo: Video Seeking in P2P Systems, in: Proc. of IPTPS, 2009.

[39] X. Wang, C. Zheng, Z. Zhang, H. Lu, X. Xue, The design of video segmentation-aided VCR support for P2P VoD systems, IEEE Transactions on Consumer Electronics 54 (2).

[40] P. Krishnan, D. Raz, and Y. Shavitt, The Cache Location Problem, IEEE/ACM Transactions on Networking, Vol. 8, No. 5, 2000.

[41] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, On the Placement of Web Server Replicas, Proceedings IEEE INFOCOM, pp. 1587-1596, April 2001.

[42] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, On the placement of Internet Instrumentation, Proc. IEEE INFOCOM, pp. 295-304, March 2000.

[43] M. Allen, B. Zhao, and R. Wolski. Deploying Video-on-Demand Services on Cable Networks. In IEEE Int. Conf. on Distributed Computing Systems (ICDCS), 2007.

[44] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot. Achievable catalog size in peer-to-peer video-on-demand systems. In Int. Workshop on Peer-To-Peer Systems (IPTPS), 2008.

[45] A. Nafaa, S. Murphy, and L. Murphy. Analysis of a Large-Scale VOD Architecture for Broadband Operators: A P2P-Based Solution. IEEE Communications Magazine, December 2008.

[46] K. Suh, C. Diot, J. Kurose, L. Massouli´, C. Neumann, D. F. Towsley, and M. Varvello. Push-to-peer video-on-demand system: Design and evaluation. IEEE Journal on Selected Areas in Communications, 25(9):1706–1716, 2007.

[47] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. H. Katz, Efficient and Adaptive Web Replication using Content Clustering, IEEE Journal on Selected Areas in Communications, Vol. 21, Issue 6, pp. 979-994, August 2003.

[48] B. Wu and A. D. Kshemkalyani, Objective-optimal algorithms for long-term Web prefetching, IEEE Transactions on Computers, Vol. 55, No. 1, pp.2-17, 2006.

[49] J. Kangasharju, J. Roberts, and K. W. Ross, Object Replication Strategies in Content Distribution Networks, Computer Communications, Vol. 25, No. 4, pp. 367-383, March 2002.

[50] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek, The Measured Performance of Content Distribution Networks, Computer Communications, Vol. 24, No. 2, pp. 202-206, February 2001.

[51] M. J. Freedman, E. Freudenthal, and D. Mazires, Democratizing Content Publication with Coral, Proceedings of 1st USENIX/ACM Symposium on Networked Systems Design and Implementation, March 2004.

[52] T. T. Do, K. A. Hua, and M. A. Tantaoui. Robust video-on-demand streaming in peer-to-peer environments. Comput. Commun., 31(3):506–519, 2008.

[53] L. Guo, S. Chen, and X. Zhang. Design and evaluation of a scalable and reliable p2p assisted proxy for on-demand streaming media delivery. IEEE Transactions on Knowledge and Data Engineering, 18(5):669–682, May 2006.

[54] M. Hofmann, and L. R. Beaumont, Content Networking: Architecture, Protocols, and Practice, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 129-134, 2005

[55] N. Bartolini, E. Casalicchio, and S. Tucci, A walk through Content Delivery Networks, Proc. MASCOTS 2003, LNCS Vol. 2965/2004, pp. 1-25, April 2004.

[56] A. Vakali, and G. Pallis, Content Delivery Networks: Status and Trends, IEEE Internet Computing, IEEE Computer Society, pp. 68-74, November-December 2003.

[57] F. Douglis, and M. F. Kaashoek, Scalable Internet Services, IEEE Internet Computing, Vol. 5, No. 4, 2001, pp. 36-37.

[58] G. Peng, CDN: Content Distribution Network, Technical Report TR-125, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, Stony Brook, NY, 2003.

[59] W; Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, Cooperative Content Distribution and Traffic Engineering in a ISP network, SIGMETRICS/Performance ’09, June 15-19, Seattle, WA, USA

[60] Y. Liu, Y. Guo, and C. Liang, A survey on peer-to-peer video streaming systems, Peer-to-Peer Networking and Applications, vol. 1, Springer New York, 2008.

[61] 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 SIGCOMM Comput. Commun. Rev., vol. 38, no. 4. ACM, 2008, pp. 375–388

[62] Y. Guo, K. Suh, J. Kurose, and D. Towsley, P2Cast: peer-to-peer patching for video on demand service, Multimedia Tools and Applications, vol. 33, pp. 109-129, 2007.

[63] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez, Exploring VoD in P2P swarming systems, Proc. IEEE INFOCOM, pp. 2571-2575., 2007

[64] F. Pianese, D. Perino, J. Keller and E. W. Biersack, PULSE: An adaptive, incentive based, unstructured P2P live streaming system, IEEE Transactions on Multimedia, vol. 9, pp 1645-1660, 2007.

[65] A. A. Hamra, E. W. Biersack, and G. Urvoy-Keller, A Pull-Based Approach for a VoD Service in P2P Networks, High Speed Networks and Multimedia Communications, vol. 3079/2004; pp. 995-1006. , 2004

[66] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg, Epidemic streaming algorithms: optimal performance tradeoffs, ACM SIGMETRICS, 2008.

[67] D. Xu, S. S. Kulkarni, C. Rosenberg and H. K. Chai, Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution, Multimedia System, vol. 11, no. 4, pp. 383-399, 2006.

[68] C. Huang, J. Li, and K. W. Ross, Can video on demand be profitable, Proc SIGCOMM’07, Kyoto, Aug. 2007.

[69] E. Setton and J. Apostolopoulos, Towards Quality of Service for peer-to-peer video multicast, IEEE ICIP, Sept. 2007.

[70] E. Kusmierek, Y. Dong and D. Dong, Loopback: exploiting collaborative caches for large-scale streaming, IEEE Transactions on Multimedia, vol. 8, pp 233-242, 2006.

[71] Y. He, G. Shen, Y. Xiong, and L. Guan, “Optimal prefetching scheme in p2p vod applications with guided seeks,” IEEE Transactions on Multimedia, vol. 11, no. 1, pp. 138–151, Jan. 2009.