Boosting Streaming Video Delivery
with WiseReplica

Guthemberg Silvestre[1]([✉]), David Buffoni[2], Karine Pires[2],
Sébastien Monnet[2], and Pierre Sens[2]

[1] CNRS, LAAS, 7 Avenue du Colonel Roche, 31400 Toulouse, France
gdasilva@laas.fr
[2] UPMC Sorbonne Universités, LIP6, CNRS, INRIA, 4 Place Jussieu, Paris, France
{david.buffoni,karine.pires,sebastien.monnet,pierre.sens}@lip6.fr

Abstract. Streaming video consumption has risen sharply over the last years. It has not only reshaped the Internet traffic, it has also changed the manner of watching videos. Users are progressively moving from the old-fashioned scheduled television to video-on-demand (VoD) services. As broadcasting future seems to be online, customers have become more sensitive to VoD quality, expecting ever-higher bitrates and lower rebuffering. In this context, average bitrate is a key quality of service (QoS) metric. Therefore, content delivery networks (CDNs) and content providers must be committed to enforce average bitrate through service-level agreement (SLA) contracts. Adaptive content replication is a promising technique towards this goal. However, this still offers a major challenge for CDN providers, particularly as they aim to avoid waste of resources. In this work, we introduce WiseReplica, an adaptive replication scheme for peer-assisted VoD systems that enforces the average bitrate for Internet videos. Using an accurate machine-learned ranking, WiseReplica saves storage and bandwidth from the vast majority of non-popular contents for the most watched videos. Simulations using YouTube traces suggest that our approach meets users expectations efficiently. Compared to caching, WiseReplica reduces the required replication degree for the most-watched videos by two orders of magnitude, and under heavy load, it increases the average bitrate by roughly 85 %.

Keywords: Peer-to-peer (P2P) · Video on-demand (VoD) · Caching · Replication · Service-level agreement (SLA) · Prediction

1 Introduction

The increasing consumption of Internet videos has made fundamental changes in the Internet traffic and consumers' behaviour. Cisco System, Inc[1] forecasts that the sum of all forms of video traffic will be in the range of 80 to 90 percent of

---

[1] Cisco Visual Networking Index: Forecast and Methodology, 2013–2018. www.cisco.com, 2014.

global consumer traffic by 2018, including video on-demand (VoD), live streaming, and peer-to-peer (P2P) file sharing. In fact, as the Internet access has become ubiquitous, continuously faster, and cheaper, streaming video has become mainstream. Users are progressively moving from the old-fashioned scheduled television to VoD services. This contributes to increase the expectations of consumers on Internet video delivery.

Since broadcasting future seems to be online, customers have become more sensitive to VoD quality, expecting ever-higher bitrates and lower rebuffering. Contrary to many traditional workloads, e.g. social network messaging or search engines, specifying just latency as quality of service (QoS) metric does not suffice. Instead, streaming traffic requires proper average bitrate to avoid rebuffering and improve user experience. For example, Dobrain *et al.* [13] found that a 1 % increase in buffering ratio can reduce the consumer's expected viewing time by more than three minutes. Balachandran *et al.*, observe that increased average bitrate in Internet video delivery leads to a better user experience for viewers with mobile devices [3]. This suggests that service-level agreement (SLA) contracts must include average bitrate as a key QoS metric.

Yet, current Content Delivery Networks (CDN) platforms are not ready to fulfil the requirements of the increasing demand for VoD services and meet consumers' expectations. Through fine-grained client-side measurements from over 200 million client viewing sessions, Liu *et al.* [21] showed that 20 % of these sessions experience a rebuffering ratio of at least 10 %, 14 % of users have to wait more than 10 s for video to start up, more than 28 % of sessions have an average bitrate less than 500 Kbps, and 10 % of users fail to see any video at all.

To deal with these issues, CDN providers have started to combine datacenters and edge network resources in hybrid designs[2]. This includes *peer-assisted VoD systems* [17] whose deployment requires hybrid CDN platforms. The aim of peer-assisted VoD systems is to take advantage of both infrastructure-based resources and P2P communication facilities. Huang *et al.* [17] suggest the use of peer-assisted VoD systems to improve resource allocation for Internet video delivery. They argue that devices on edge networks, e.g. set-top-boxes, contribute with storage and bandwidth to video delivery, reducing dramatically the burden on infrastructure-based servers, and cutting operations costs. Many recent studies [10,18,25] confirm that exploring peer-assisted VoD system permits enhancing resource allocation for streaming videos, but none has properly evaluated the performance of video delivery regarding SLA enforcement.

In fact, there exists an increasing need for more research in easy-to-deploy, self-adapting techniques for ensuring tough QoS guarantees brought by the cloud paradigm. However, efficient resource allocation on hybrid CDNs to meet user expectations imposes big challenges, particularly for resource-hungry services as VoD. This paper identifies *adaptive content replication* as one of such challenges. Adaptive replication plays an important role on the content availability of distributed systems, contributing directly to both storage and bandwidth provision.

---

[2] Akamai acquires Red Swoosh. http://www.akamai.com/html/about/press/releases/2007/press_041207.html, April 2007.

As the popularity of a video varies, the number of replicas, or peers serving that video, must be adapted accordingly. Generally speaking, the faster and more precise the replication scheme reacts to changes on videos demand, the better is the resource allocation and content availability.

Considering average bitrate as target QoS metric, we make a case for a SLA-driven replication scheme named WiseReplica that allows us to meet users' expectations in peer-assisted VoD system properly. We assume the system must enforce the right average bitrate for each video through SLA contracts. Our ultimate goal is two-fold: (i) to prevent SLA violations and (ii) to reduce the number of video replicas. To perform efficient Internet video replication, WiseReplica relies on a novel, accurate machine-learned ranking of Internet videos. To rank video in order of demand, our prediction model encompasses multiple measurements of Internet video activity in peer-assisted VoD system, including active viewers, video duration, average serving time, and mean time between requests view. The use of this prediction model in WiseReplica provides the ability to adapt the replication degree of videos dynamically according to their encoding settings and popularity, reducing storage usage and enhancing network provision. We make two main contributions:

Investigate how predictable is a ranking of Internet videos. We design a learning model to capture the dynamic behaviour of streaming video demand. The model makes predictions based on lightweight measurements of the request arrival process. Using a novel machine-learned ranking, we predict demand of a video accurately. Thus, the higher the rank position, the higher the demand for fresh replicas. According to the video ranking position, VoD services operators can define and evaluate different replication policies. For instance, top-ranked Internet videos may be twice as much replicated as those ranked in the second position. This intuitive model allows us to decouple streaming demand from replication policy. Our model is flexible and can learn from different sources and big amounts of data, providing a robust framework for controlling VoD resource allocation. Simulations using YouTube traces, with non-stationary behaviours, suggest that our model is very accurate in predicting the ranking of Internet videos. Since our ranking of videos is based on random forests, a parallelizable, state-of-the-art machine learning method, it fits runtime requirements of large VoD systems.

Enforce average bitrate through SLA-based video replication. Based on our machine-learned ranking of Internet video, we designed and evaluated WiseReplica, an easy-to-deploy, SLA-based replication scheme that meets users' expectations for VoD services. WiseReplica is fully compliant with peer-assisted VoD systems in hybrid CDN platforms. It operates adaptive replication over sets of devices located close to each other in edge networks, namely *storage domains*. WiseReplica functioning per storage domain is straightforward. Gradually, it verifies the rank position of a video whenever a new local request arrives, and adapts the replication degree accordingly. Using a collaborative caching, video replicas are either pre-fetched or removed randomly. We show through simulations using YouTube traces that WiseReplica outperforms a non-collaborative

caching approach by preventing violations, reducing storage usage, and enhancing network resources provision. Furthermore, our replication scheme is easy to adopt and flexible enough to offer interoperability with de facto approaches, including HTTP adaptive streaming technique and BitTorrent protocol [32].

This work is organized as follows. In Sect. 2, we present the context and challenges of this research. We describe in details our prediction model to rank Internet videos in order of demand in Sect. 3. Section 4 describes the approach of our adaptive replication scheme, WiseReplica. We explain our simulation methodology in Sect. 5. We then analyse the performance of WiseReplica in Sect. 6. Related works are discussed in Sect. 7, just before the conclusion in Sect. 8.

2 Context and Challenges

In this section, we describe the context and the challenges of this work.

2.1 Improving Content Availability to Better User Experience

Many studies have shown that quality of user experience while watching online videos is related to the good quality of content transmission. They presented many strategies to enhance the video content availability and its distribution. Most of these studies analyse in the field are focused on Youtube, being this the major player of video content distribution [1,5,6,14]. These studies include the analysis of crawled data from Youtube APIs and comparisons of caching strategies from collected data of users' point of view (HTTP logs from ISP or local networks).

Dobrian *et al.* [13] study has shown the correlation between the user engagement and the video quality, being the *Buffering Ratio* (fraction of the total session time spent in buffering) and *Rendering Rate* (frames per second) the most critical metric over the total played time for short videos, the current target of our method. This characterizes the relation between quality of service and user experience and endures the importance of avoidance of SLAs violations, which minimizes buffering ratio, confirming the main metric of evaluation in our work.

Furthermore, Finamore *et al.* [15] stated that the download bitrate of the video plays a fundamental role in video playback quality. They measured the smoothness of the playback by the bitrate ratio, defined as the ratio between the average session download bitrate and the video encoding bitrate. Considering different bitrates in the input dataset is a key aspect of the rendering rate metric in user engagement and playback quality.

Another important concern in studying the availability of Internet videos was to provide a SLA-based solution with the minimum constrains regarding deployment. However, recent studies [2,34] are based in substantial changes in the normally used stack of protocols and network infra-structure and they become hard to be considered as a feasible solution. Our solution takes in account a

well-know and largely used infra-structure and have no changes in the stack protocol. Despite all these research efforts, enforcing video availability in large peer-assisted VoD systems remains a challenging issue.

2.2 On the Track of YouTube Popularity Growth Curves and High Quality Videos

A fair reproduction of user interactions to Internet videos is essential to evaluate peer-assisted VoD systems properly. Hence, we study in this work a workload that combines YouTube traces [14] to well-known videos' access patterns [33]. We are particularly interested in reproducing realistic popularity growth curves, considering advanced coding setting and common VoD demand patterns.

We study the data crawled by Figueiredo *et al.* [14], whose datasets are currently available online[3]. The dataset allows us to characterize the growth patterns of YouTube videos. In particular, they analysed three types of YouTube videos sets: videos that appear on YouTube top list, videos that were banned from YouTube due to copyrights violations, and videos that were randomly selected through API calls. They crawled once a number of videos' daily features. For each video, there are up to 100 daily measurements, or daily available samples, per feature. In this work, we are mostly interested in the measurements of *view data* feature, that depicts the popularity growth curve of a video through a array of cumulative number of daily views ranging from 0 to the total number of views.

In order to reproduce realistic, high quality videos encodings, we consider the YouTube advanced encoding settings[4]. Table 1 depicts the set of high definition (HD) video encodings that we use in this work.

Table 1. Advanced encoding settings for YouTube videos used in this work.

Type	Video bitrate	Mono audio bitrate	Stereo audio bitrate	5.1 Audio bitrate
1080p	50 Mbps	128 kbps	384 kbps	512 kbps
720p	30 Mbps	128 kbps	384 kbps	512 kbps
480p	15 Mbps	128 kbps	384 kbps	512 kbps
360p	5 Mbps	128 kbps	384 kbps	512 kbps

2.3 Investigating Network Resources Provision for Internet Videos

Replication schemes have become an important building block for Internet video providers to improve content availability and meet consumers' expectations. An

---

[3] The Tube over Time: Characterizing Popularity Growth of YouTube Videos. http://www.vod.dcc.ufmg.br/traces/youtime/data/, January 2013.

[4] Advanced encoding settings for YouTube videos. http://support.google.com/youtube/bin/answer.py?hl=en-GB&answer=1722171, June 2014.

adaptive replication scheme should offer content replica maintenance to handle popularity growth properly.

Non-collaborative caching remains the simplest approach to provide adaptive replication of web content [20]. They adapt the replication degree to the content popularity using cache replacement policies, and assuming fair-sharing as a key scheduling strategy. But, Internet videos' workloads on peer-assisted VoD systems bring major obstacles to non-collaborative caching, e.g. the resource imbalance in peers for replicas, and a growing need for high bitrate provision for meeting consumers' expectations. Therefore, relying just on cache replacement policies and fair-sharing scheduling can undermine the performance of the whole system.

Recent studies have sought an optimal solution to this problem. For instance, Chang and Pan [10] propose a modelling framework towards optimal caching strategies, including collaborative caching. They confirm that this problem is NP-hard, and only suboptimal solutions can be found.

2.4   Challenges

In order to meet increasing consumers' expectations on Internet videos, a *good* peer-assisted VoD system must overcome the following challenges:

1. It must cope with dramatic, unexpected variations in videos popularity.
2. It must avoid waste of resources, and reduce as much as possible storage and network usage on peers of edge networks.
3. It must prevent rebuffering of VoD streaming through a self-adaptive, easy-to-deploy technique.

Our simulations suggest that meeting consumers' expectations in terms of average bitrate is a difficult task, specially under heavy load. State-of-the-art approaches fail to handle these challenges mostly because they are not able (i) to capture VoD demand, and (ii) to define a metric to measure consumers' expectations. WiseReplica copes with these issues by inferring users' expectations for videos and predicting the amount of resources to fulfil the demand in a self-adaptive way. Our findings show that this approach produces a good balance between resource allocation and users' satisfaction.

3   A Machine-Learned Ranking of Internet Videos

We designed a prediction model for ranking Internet videos in order of demand. In this work, video demand involves both popularity and QoS requirements. Our main goal is to provide an intuitive, accurate method to capture requesting behaviours of streaming videos. In this section, we highlight the foundations of our statistical learning approach. First, we present a brief overview of statistical learning. Then we explain the model, describing our learning-to-rank problem. Finally, we describe our implementation and we present a framework for ranking predictions.

3.1   An Overview About How to Learn from Data

Statistical learning is about learning from seen data in order to predict unseen data with minimal error. Data comprise inputs $\mathbf{x}$ represented by a vector with a fixed number of dimensions p ($\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$) from the input space $\mathcal{X}$. In our problem, $\mathbf{x}$ is a video, represented by a vector of measurements from video sessions' activity.

In supervised learning, each input measurement is coupled with a y, a label selected by an oracle, from the output space $\mathcal{Y}$. To learn, we take N pairs $(\mathbf{x}, y)$ drawn *independently and identically distributed* (i.i.d.) from a fixed but unknown joint probability density $Pr(X, Y)$. This is true for both training and testing datasets. For instance, we consider the training dataset $S = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ of N pairs $(\mathbf{x}, y)$. Using this dataset, the supervised learning algorithm searches for a function $f \colon \mathcal{X} \to \mathbb{R}$ in a fixed function class $\mathcal{F}$. State-of-the-art algorithms, such as *support vector machines* (SVM) [11] or *ensemble methods* [16], aim to find $f^\star$ in $\mathcal{F}$ with the lowest empirical risk defined as:

$$f^\star \in \arg\min_{f \in \mathcal{F}} \mathbf{r}_{emp}(f) \tag{1}$$

where $\mathbf{r}_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} I_{\{f(\mathbf{x}) \neq y_i\}}$ is computed over the training set, and $I_{\{.\}}$ is the indicator function which returns 1 if the predicate $\{.\}$ is true and 0 otherwise. In other terms, $\mathbf{r}_{emp}$ is a quality measure relating the label to the prediction provided by the function f on the training dataset S.

To model our prediction problem, we use a statistical learning approach called learning-to-rank. This approach has been a hot topic in Machine Learning community for the last 10 years. It combines properties of two well-known other approaches: regression, where $y \in \mathcal{Y} \subset \mathbb{R}$; classification where $y \in \mathcal{Y} \subset \{0, 1, ..., K\}$ with $K \geq 1$. In learning-to-rank approach, y gives an indication on the target order (formally represented by a permutation $\sigma \in \Sigma$).

3.2   A Ranking Model for Internet Videos

The main purpose of our learning model is to capture popularity growth dynamics and system resources availability of peer-assisted VoD systems. Therefore, we assume that prediction model must allow us to rank Internet videos in order of demand. This can be modelled as a learning-to-rank problem.

Given an i.i.d. sample $(\mathbf{x}, y)$ such as described in Subsect. 3.1, we model inputs and outputs as follows.

Inputs. We represent the input space $\mathbf{x}$ is a video described as 10 lightweight measurements from the request arrival process. These measurements are video size, network availability, network usage (load), current number of viewers and replicas, inter-arrival time between requests (delta), aggregate number of views, mean of time between requests (mtbr), life time, and average bitrate. We compute averages and means from up to the five last requests. Our goal is to gather as much information about users' interactions as possible in an easy manner to

make accurate predictions about the ranking of videos. To extend our model, one can easily add further features or measurements such as geographical location, social network interactions data, buffering ratio, rate of buffering events, session join time, rendering quality, rate of bitrate switch, *etc.*

Outputs. The supervision y associated to each input video $\mathbf{x}$ is based on four possible ordered values which gives an indication for the final target ranking. In our model, $\mathcal{Y} \in \{0, 1, 2, 3\}$, whose labels are {*non-popular*, *popular*, *very popular*, *viral*} respectively. It represents a natural ranking for Internet videos. Using this ranking model, we intend to provide a measure of video demand, which is closely related not only to the popularity, but also to the consumption of system resources.

Finally, the learning-to-rank module finds a function f from Eq. (1) with the constraint of maintaining the prediction order: $\forall i, j, i \neq j, y_i > y_j$ then $f(\mathbf{x}_i) > f(\mathbf{x}_j)$ explained in [7]. In that case, theoretical performance guarantees are provided. Practically, the use of the mean square error $(y - f(\mathbf{x}))^2$ instead of the indicator function $I_{\{.\}}$ (which is hard to optimize because it is non-differentiable) allows us to ensure a calibrated learning to rank algorithm. A calibrated algorithm means there is a theoretical link between the approximation of the empirical risk, that is easier to optimize, and its non-differentiable version [7,9,31,35].

3.3   Framework for Learning and Predicting, and Implementation

We implement our model using *ensemble methods*. According to Friedman *et al.*, ensemble learning consists of a set of very popular supervised methods, that are robust, simple to train and tune, and have a remarkable prediction performance. Our implementation is based on SCIKIT-LEARN, a general-purpose machine learning library [26].

We designed a simple framework to use our learning module, depicted in Fig. 1. Our framework has two phases: (i) learning and (ii) predicting. Each phase has its own YouTube-like workload. Learning is a preliminary phase that commonly runs offline in a batch mode, while the prediction can go online. In this work, both phases are performed with data from simulations. In the learning phase, we first generate the training dataset, described in Subsect. 5.4. Then we feed this training dataset to our learning model, represented here as module of
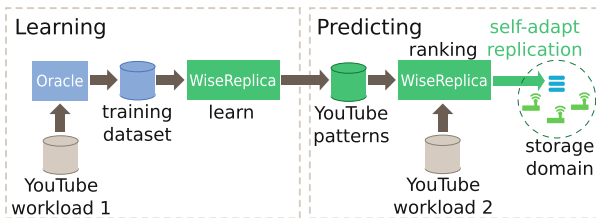

Fig. 1. Framework for learning and predicting ranking of Internet videos.

WiseReplica, in order to identify YouTube ranking patterns. Once the learning phase has been accomplished, WiseReplica can use its learning module in a predicting phase, as indicated in the left-hand side of Fig. 1. In this phase, inputs come for measurements of the request arrival process of workload 2, that permit accurately ranking Internet videos in order of *hotness* and instrumenting replication accordingly inside storage domains. We highlight WiseReplica functioning, including storage domains, in the next section.

4 Boosting VoD Delivery: The WiseReplica Approach

In this section, we describe WiseReplica replication scheme. First, we highlight how WiseReplica operates in edge networks, by in introducing the concept of storage domains. Then, we explain its replication strategy based on predictions of ranking of video demand.

4.1 Distributing VoD with Storage Domains

We assume that WiseReplica operates in peer-assisted VoD systems deployed on hybrid CDN platforms. We consider the hybrid CDN design called Caju, that is detailed in our previous work [30] as our target platform. It is based on sets of devices located close to customers, named storage domain. A storage domain is a logical entity that combines resources from both datacenters and edge networks in the last mile of the content delivery chain. As Fig. 2 shows, devices in a storage domain can play either a coordinator or peer role.

Coordinator is a server or a small-sized cluster of servers deployed in the nearby datacenter. We assume that the coordinator performs scheduling of video requests for the local storage domains. Therefore, it runs the main instance of WiseReplica, and keeps information about resources consumption. Its main goal is to maintain the right number of replicas per video in the local peers, by prefetching or deleting sources. Instead of always contacting the content providers, coordinators might interoperate in logically centralized way to fetch videos that have been vanished from a storage domain. They store the most recent videos in their own cache for replication purposes. Whenever a new replica is necessary, the coordinator pushes it to a randomly, uniformly selected peer. Similarly, coordinators send video deletion requests to local peers.

Peers is a set of devices located close to each other through which customers get network access, e.g. home gateways connected to the same digital subscriber line access multiplexer (DSLAM). These devices actually deliver videos to customers in a storage domain, being the main source of storage and network resources. They execute scheduling and replication commands sent by the storage domain's coordinator. Each peer contributes with a percentage of storage and network resources to the system, as in a collaborative caching. In the local cache is applied the LRU policy for videos replacement.

This model is specially interesting for the problem of videos delivery as it takes advantage of nodes geographical position [6]. It provides two main
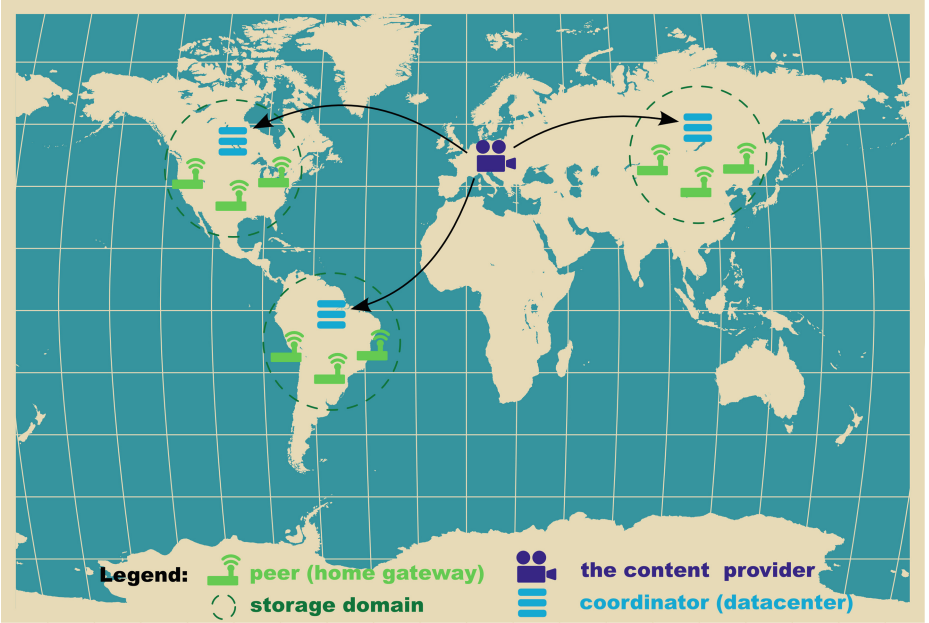
Fig. 2. Storage domains.

infrastructure properties to WiseReplica: replication group and hop limit. The replication group allows WiseReplica to adapt video replication for smaller sets of peers, most likely connecting customers with similar content interests. By enforcing a hop limit, storage domains avoid jitter, ensure low latencies, and permit WiseReplica improving the efficiency of network resource provision.

In addition, we assume that a storage domain enforces an initial placement policy. This policy defines the minimum replication degree m for initial copies for any new, just fetched Internet video. Request scheduling is simple. A view request is served by at most R nodes with uniform load. Available sources come from $r = min(n, R)$, where n is the number of current replicas. In this work, we consider m equals to two and R equals to five as default settings. For requests scheduling, this approach enforces well-known policies for peers in edge networks, including nearest source selection and multi-sourcing.

4.2 Self-Adapting Replication According to the Ranking of Internet Videos

Our utmost goal is to contribute to meet increasing customers expectation on Internet videos using peer-assisted VoD systems. To enhance VoD delivery, we assume that rebuffering is a major issue to be addressed. We propose to cope with this issue by enforcing minimum average bitrate of each streaming as the main QoS metric. In this scenario, content and CDN providers must be committed to enforce minimum average bitrate for videos through SLA contracts. Since

Internet videos delivery is a resource-hungry service, we must also adapt the network provision and storage usage as we aim to prevent violations. To this end, we propose WiseReplica, an adaptive replication scheme for peer-assisted VoD systems based on storage domains.

WiseReplica maintains replicas inside a storage domain. Running on the coordinator, it adapts the replication degree of Internet videos of a storage domain according to a machine-learned ranking. Our scheme follows a three-part procedure:

Collect Information from the Request Arrival Process. For each video request, WiseReplica collects 10 lightweight measurements. The goal is to gather comprehensive information for measuring the video demand and accurately predicting the raking. As described in Sect. 3, they are video size, network availability, network usage (load), current number of viewers and replicas, inter-arrival time between requests (delta), aggregate number of views, mean of time between requests (mtbr), life time, and average bitrate. We compute averages and means from the last up to five requests. It is important to notice that all these measurements can be easily collected in the storage domain's coordinator.

Rank Internet Videos in Order of Demand. Based on the measurements of the request arrival process, we use the learning model described in Sect. 3 for predicting the rank position of Internet videos demand. We can predict the video rank for each view from the second request. The ranking comprises information about demand and QoS requirements. Predictions are quite essential for enhancing VoD delivery. Since our learning model make predictions on a request basis, WiseReplica can react to the video demand as promptly as the rank position evolves. Indeed, ranking is an intuitive way to capture the demand of videos in peer-assisted VoD systems. The higher is the demand rank position of an Internet video, the higher is the demand for it. There are four positions on our machine-learned ranking: *non-popular, popular, very popular,* and *viral.* WiseReplica has a straightforward strategy to perform replication according to hotness rank positions. Videos that fall into the lowest rank position can have their replication degree reduced, otherwise they need more replicas. Thus, the maintenance of replication degree of Internet video, including video creations and deletions in peers, relies on replication policies.

Enforce Replication Policy Accordingly and in Time. The goal of replication policies is two-fold: first (i) ensure consumers' expectations in time and (ii) reduce the total number of replicas as much as possible. For that, WiseReplica must adapt replication of videos according to the forecasts of their rank positions. Our replication scheme enforces two types of replica maintenance policies: deletion and creation policy. In this work, we enforce a single video deletion policy. Whenever the coordinator receives a request to a video in the non-popular rank, the deletion policy says that one replica is deleted until the minimum replication degree m is reached. Similarly, our scheme periodically runs a maintenance procedure (e.g. each five minutes) to smoothly enforce the deletion policy for inactive videos. This allows WiseReplica to reduce the total number of replicas. To cope

with SLA violations and meet customers' expectations, we evaluate four quite simple policies, namely uniform, linear, quadratic, and exponential. They are respectively defined as follows: B, Br, Br^2, and B^r, where B is a constant that represents the target number of replicas, and $r \in \{1, 2, 3\}$ the rank positions. We report on creation policies' performances in Sect. 6.

Our findings show that this approach produces a good balance between resource usage and consumers' satisfaction. It is important to note, however, WideReplica does not cover video durability, neither does fault-tolerant mechanisms (e.g. failure detection/recovery procedures). Rather, our goal is to improve VoD availability, boosting network provision, meeting consumers' expectations on VoD services, and reducing storage usage as much as possible. To this end, WiseReplica combines lightweight measurements, accurate predictions of Internet videos ranking, and replication policies enforcement in a particularly novel, flexible way. In peer-assisted VoD systems, it can easily interoperate with de facto approaches, including HTTP adaptive streaming technique and swarming protocols, such as BitTorrent.

5   Simulation Methodology

We simulate a peer-assisted VoD system based on a hybrid CDN design called Caju [30]. We evaluate WiseReplica using YouTube traces. We compare WiseReplica performance with other two adaptive replication schemes, namely non-collaborative caching and Oracle-like collaborative caching. The aim of our simulations is to study in details the variability of demand and resource allocation of VoD services on edge networks, and the performance of replication schemes in enforcing expected Internet video availability.

5.1   Workload from YouTube Traces and SLA Definition

The workload and SLA definitions are at the core of our evaluation. We define a workload that captures the main features of VoD services using YouTube traces, and a SLA contract that meets users' expectations.

In the workload definition, we are particularly interested in reproduce a realist request arrival process, placing the emphasis on popularity growth and video encodings. Thus, we use YouTube traces, presented in Subsect. 2.2. Before integrating YouTube traces to our workload, we first preprocessed their YouTube datasets to remove inconsistent measurements, such as videos with no views. Basically, we got rid of videos with small number of total views (those smaller than the first quartile) and videos with few daily measurements (those smaller than the third quartile). That allowed us to pick off 20 % most representative YouTube growth patterns, accounting for 21827 distinct curves. Then, we randomly selected, with a uniform distribution, curves from this preprocessed data to be assigned to videos of our workload. Similarly, we assigned high quality YoutTube video encodings to our workload videos, based on advanced settings depicted in Table 1. To summarize, Table 2 lists default values for workload

Table 2. Default values for workload parameters.

Workload	
Requests per user	uniform
Experiment duration	4 h
Mean requests per second	100
Requests fractions	5 % of creations, 95 % of views
Video size (follows Pareto)	shape = 3, between 13 MB and 1.6 GB
Video popularity (Zipf-Mandelbrot)	shape = 0.8, cutoff = number of videos
Videos' creation (Poisson)	λ = creations per second
Popularity growth from YouTube traces	21827 distinct patterns
YouTube encoding settings (bitrates)	5 Mbps, 15 Mbps, 30 Mbps, 50 Mbps

parameters. Finally, videos are always divided and distributed in chunks or segments of fixed size, 2 MB.

In terms of SLA definition, we assume that content and content delivery providers are committed to improving the Internet video availability for customers in a content-oriented approach. In our case, a *good* peer-assisted VoD system must ensure videos availability by avoiding rebuffering. Therefore, we consider a global, simple SLA contract drawn up to provide a minimal average bitrate according to each Internet video encoding setting. A SLA violation happens whenever the system fails to enforce the minimal average bitrate for any viewer session.

5.2   Evaluation Scenario

Our evaluation scenario (Fig. 3) includes 4002 nodes, arranged across two storage domains. There are one coordinator and 2000 peers per storage domain. Storage and network capacities differ according to the device role. Coordinators have 20 TB of storage capacity and full-duplex access link of 4 Gbps. Peers contribute 200 GB each, equipped with 100 Mbps full-duplex links. Note that the two coordinators contribute with a small fraction of aggregate edge resources, i.e. 5 % of the storage capacity and only 2 % of the total network capacity. This draws our attention to the performance of replication schemes towards peers resource allocation. We assume only 1 % peers' storage is available for caching additional replicas, namely 2 GB.

We implemented and evaluate this work using simulation. To this end, we developed a simulation tool on top of PeerSim [24] to implement storage domains in edge network and bandwidth scheduling.Our design focus on network's resource allocation accuracy for simulating bitrate enforcement and concurrent videos views properly. Design and implementations details of our tools to simulate network resource scheduling are available in our previous work [28]. We have performed our simulations using servers equipped with Intel Xeon E5450 3.00 GHz, and a RAM of 4 GB.
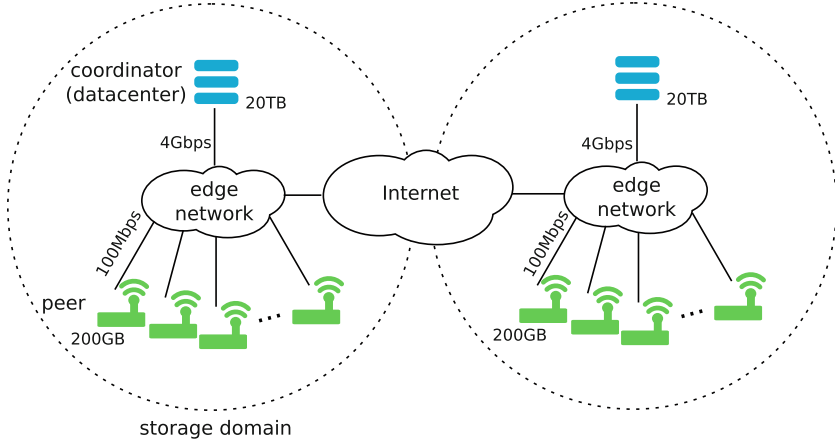
Fig. 3. Evaluation scenario.

5.3 Comparable Replication Schemes

We compare WiseReplica with two other schemes.

Non-collaborative Caching. Adaptive replication schemes based on non-collaborative caching, such as those that uses Least Recent Used (LRU) algorithm, are easy to implement and deploy. A new replica is created in a peer whenever a user requests to view a video. LRU replacement is enforced regarding the static percentage of the local storage capacity for caching of 1 %.

Oracle-Like Collaborative Caching. This is an idealized benchmark case. Here, we assume a peer-assisted VoD system deployed in a network that runs a deadline-aware transport protocol, similar to Wilson *et al.* [34] work. Based on our previous work with AREN [29], an adaptive replication scheme for edge networks, we implemented a benchmark replication scheme that relies on bandwidth reservation and collaborative caching to provide an adaptive number of replicas for videos. We replicate videos according to aggregate network usage by enforcing a low and high thresholds. This makes the video replication a function of bandwidth reservation, and ensures that network and storage provision follows video demand properly, as depicted in Fig. 4. Per video, we consider two percentage thresholds for aggregate
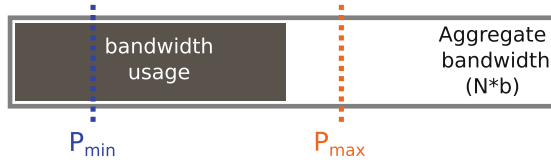

Fig. 4. Oracle-like bandwidth management for a video, illustrating aggregate bandwidth for N replicas and b available bandwidth, bandwidth reservation (bandwidth usage) and thresholds (P_{min} and P_{max}).

network usage: P_{min} and P_{max}. Our replication strategy works as follows. A video v that has N replicas in peers with network capacity of b requires more replicas if the current bandwidth reservation $U(v) > P_{max} \sum_{i=1}^{N} b$. Similarly, if $U(v) < P_{min} \sum_{i=1}^{N} b$, replicas can be deleted. Otherwise, keep the replication degree. Although this empirical approach is hard to be adopted in a real deployment, our previous results [29] suggest that it allows us to achieve near-optimal results, preventing *all* SLA violations, enhancing network usage and decreasing storage usage dramatically.

5.4   Collecting the Datasets for Learning

To perform rank predictions of Internet videos, we need training datasets from which we can learn the behaviour of video demand in peer-assisted VoD systems. In this section, we explain the methodology to gather data for these predictions.

The training dataset of our prediction model comes from measurements of the request arrival process on per-assisted VoD systems, as described in Subsect. 3.2. Each line of our training dataset has 11 values, 10 input measurements about a video current state, and a rank position. Although, the datasets evaluated in this work were synthetically collected by performing simulations with the Oracle-like benchmark replication approach (detailed in Subsect. 5.3), similar datasets can be collected from monitoring systems of running CDN systems.

In this work, Oracle-like benchmark replication approach (Subsect. 5.3) represents the near-optimal way to serve VoD service according to video encodings and popularity, whose functioning we are very interested in learning. In this empirical approach, a video requires additional replicas only if there exists a certain number of concurrent accesses, where concurrence is measured by checking a high threshold of the current reserved bandwidth, as detailed in Subsect. 5.3. We assume that popular videos are those that have additional replicas during its lifetime. Since Internet videos popularity distribution follows a Zipf-like distribution [33], concurrent access are rare events as well as popular videos classified by this approach, thus it provides a quite fair approach to identify popular videos.

Raw data from Oracle-like technique permits easily distinguishing between two ranking positions only, non-popular and popular videos, i.e. requests to non-popular videos are all those that do not trigger any replica creation, or those that resulted in deletions. However, there is a lack of information about different ranking positions of popular videos. Hence, depending on the frequency of replica creation, we add information to requests to popular videos classifying them in popular, very popular, or viral. To define these three levels of *hotness*, we run simulations with YouTube traces, collected the distribution of replicas creation in milliseconds, and split it in three nearly equal parts by observing the 66-percentile and 33-percentile inter-creation time for new replicas. This means that the higher is the frequency of replica creation, the hotter is the video, and the higher is the ranking position. Now, collected data suit model's definitions well.

6 Evaluation

The utmost goal of our performance evaluation is two-fold: (i) measure the accuracy of our learning model in ranking Internet videos in order of *hotness*, and (ii) evaluate the performance of our replication scheme in meeting viewers' expectations in peer-assisted VoD systems. Further details about evaluation set-up are available in Sect. 5.

6.1 Performance Evaluation Metrics

We aim to evaluate the performance of two main WiseReplica modules: machine-learned ranking and replication strategy. Hence we group evaluation metrics as follows:

Machine-Learned Ranking Accuracy. We adopt the normalized Discounted Cumulative Gain (nDCG) criterion as the main evaluation metric for our learning model. nDCG is a standard quality measure in information retrieval, especially for Web search [19,22]. We implement DCG measure proposed by Burges *et al.* [8]. Therefore, DCG is defined as $DCG_L = \sum_{i=1}^{L} \frac{2^{F(i)}-1}{\log_2(1+i)}$, where L is the global set of ranked videos, and $F(i)$ is the rank position of ith video. To compute nDCG, we divide DCG measure by the idealized DCG with perfect order of the set L. Thus, the perfect model scores 1. Unlike typical information retrieval problems, as a ranking of web content, our model does not have the notion of *query*. Instead, we rely on nDCG robustness to measure the performance of our learning model as a global ranking problem. Since the ranking problem shares properties with both classification and regression problems, we compare nDCG to other three popular machine learning metrics: the mean square error, a standard metric for regressions; precision, for classification; and a less robust, well-known variant of nDCG, namely in this work nDCG(2), described by Croft *et al.* in [12]. We evaluate three different state-of-the-art ensemble learning methods available in Scikit-learn library: Random Forest, Extremely Randomized Trees, and Gradient Tree Boosting. Moreover, we report briefly on the sample size for learning, number of estimators or learners of ensemble methods, measurements or features importance, and the computational overhead of our model, including memory usage and computation time for prediction.

Metrics for Replication Strategies in Peer-Assisted VoD Systems. Assuming that content and CDN providers are committed to enforcing bitrate as main QoS metric through SLA contracts, we consider SLA violation as the primary performance metric. Thus, a SLA violation happens whenever the peer-assisted VoD system does not provide the minimum average bitrate for preventing rebuffering. This measures the WiseReplica capacity of meeting consumers' expectations. We also investigate the impact of our replication scheme using storage domains in peer-assisted VoD systems. To this end, our evaluation metrics are network and storage usage. Finally, we compare WiseReplica results with a non-collaborative caching and the Oracle-like assumption, described in Subsect. 5.3.

6.2   Fitting and Measuring the Accuracy of Our Ranking Model

The evaluation of our learning model comprises: ensemble method selection, number of *estimators*, sample size for learning, and inputs' relative importance. In this subsection, we aim to evaluate the most important settings and tune our model towards higher accuracy, using the learning framework described in Subsect. 3.3.

Selecting and Fitting an Ensemble Method. Ensemble methods have become very popular in statistical learning. Their algorithms combine several *estimators* or *week learners* to provide robust learning models and prevent overfitting. We fit and evaluate our model with three methods from SCIKIT-LEARN library: RANDOM FOREST(RF), EXTREMELY RANDOMIZED TREES(ET), and GRADIENT TREE BOOSTING(GB). We consider two distinct samples with 124,000 lines each, one for training and other for testing. We set to 10 the number of estimators as a common setting. All other parameters have default settings. Based on four metrics detailed on Subsect. 6.1, RANDOM FOREST fits our model better. Figure 5 depicts three of these metrics. RANDOM FOREST performs particularly well in nDCG score, the main metric for ranking problems. While EXTREMELY RANDOMIZED TREES and GRADIENT TREE BOOSTING score 0.9126 and 0.4128 respectively, RANDOM FOREST scores 0.9594. In terms of precision, RANDOM FOREST slightly better, with a score of 0.9922. EXTREMELY RANDOMIZED TREES scores 0.9899, and GRADIENT TREE BOOSTING scores 0.9502. It also outperforms the other two methods regarding the mean square error metric, scoring 0.0094 compared to 0.0122 with EXTREMELY RANDOMIZED TREES and 0.1021 with GRADIENT TREE BOOSTING. nDCG(2) metric
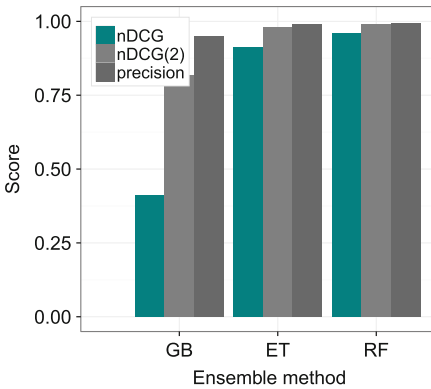

Fig. 5. Ensemble methods evaluation: RANDOM FOREST(RF), GRADIENT TREE BOOSTING(GB) and EXTREMELY RANDOMIZED TREES(ET).
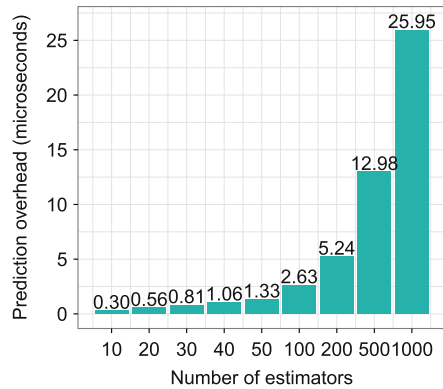
Fig. 6. Overhead for different number of estimators of RANDOM FOREST.

confirms these results. Therefore, we select RANDOM FOREST method for implementing our prediction model and nDCG as the key accuracy metric for ranking predictions.

Adjusting the Number of *Estimators* to Learn. According to Friedman *et al.*, RANDOM FOREST performs predictions by building a collection of *de-correlated* trees, namely estimators, and then averages them. We investigated the impact of the number of estimators in ranking accuracy, memory and computation time. We varied the number of estimators progressively from 10 to 1000, with the same previous samples. Results show that the number of estimators has a negligible impact in the accuracy of our model. While a model with 10 estimators scores 0.9594, 1000 scores 0.9569, slightly worse. One reason for this might be the number of inputs, relatively small, that is likely to require a small number of estimators. Yet, the number of estimators impacts on the model overhead, specially for computation time. As depicted in Fig. 6, the computation time ranges from 0.3 ms with 10 estimators to almost 26 ms with 1000 ones. Although the worst case still represents low overhead, the lower the better. Memory overhead is rather negligible, ranging from 30 to 32 MB. Overall, our model has a quite low overhead, suitable for going online in large peer-assisted VoD systems. Since there is no evidence to increase the number of estimators, we keep 10 estimators as a default, fair setting.

Evaluating Bigger Samples for Fitting the Model. Towards a higher accuracy, we evaluated bigger samples for fitting our prediction model in its learning phase, described in Subsect. 3.3. We collected more information by running longer simulations. As expected, Fig. 7 confirms that we improve accuracy through bigger samples. The improvement in accuracy was slight, about 0.03 as we use a sample size almost six times bigger, i.e. 683,000. It is quite important to highlight, though, that this has no impact on computation time of predictions. Thus, we use the biggest sample for the remaining evaluations.

Analysing the Relative Importance of Model's Inputs. We were particularly interested in evaluating the contribution of each input of our model, described in Subsect. 3.2. SCIKIT-LEARN library allows us to measure the relative importance of each input for predicting the ranking position using the RANDOM FOREST method. Figure 8 highlights the relative importance for all 10 inputs of our ranking model. The two most relevant inputs are the current number of viewers and network availability. These inputs alone account for 99.6 % of the all model's accuracy. It seems quite reasonable, since the former measures the demand for a video and the later depicts the offer of network resources, the main system feature for enforcing average bitrate. Based on the analysis of the current datasets, the remaining eight inputs are less important to the ranking model's accuracy. Surprisingly, the number of replicas, current network load, and video
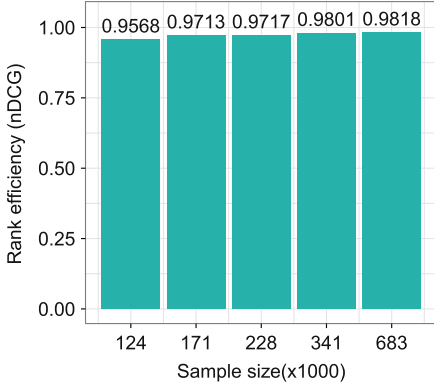
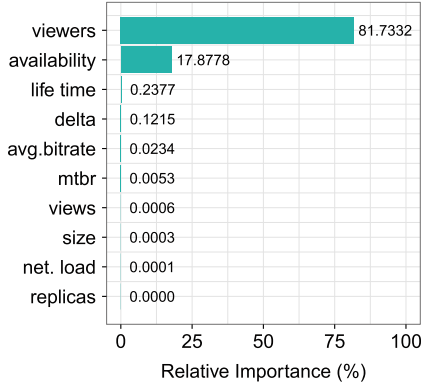**Fig. 7.** Accuracy with different sample sizes.

Fig. 8. Relative importance to ranking of the 10 model's inputs.

size seem to be useless to our model. It is likely that network availability is a particularly good measurement, making these eight inputs rather redundant. For simplicity, we include all inputs in the rest of the work. This is harmless for the model's accuracy.

6.3   Evaluating Replication Strategies in Peer-Assisted VoD Systems

In this subsection we analyse the replication strategy used in WiseReplica. First, we evaluate four simple replication policies. Then, we compare WiseReplica with a non-collaborative caching and Oracle-like benchmark replication approach, both described in Subsect. 5.3. We evaluate their capacity to meet consumers' expectation by observing the number of violations. In addition, we compare their resource allocation performance regarding network and storage usage.

Enforcing Simple Replication Policies on Ranked VoD. For the three highest rank position, WiseReplica enforces a replica creation policy, described in Subsect. 4.2. It defines the replication degree growth factor. Considering the smallest evaluated system load (with mean video size of 20 MB), we analyse four simple creation policies, namely uniform, linear, quadratic, and exponential. Table 3 shows the number of violations by varying B from 2 to 6. Overall, creation policies that take into account the rank positions, i.e. linear, quadratic, and exponential, performed better. Results show that there is relatively small difference for $B \geq 3$, suggesting that our ranking model reacts promptly to modifications on network availability, preventing over-replication. However, for $B \geq 5$, it appears that replication increases the network load system load, causing few more violations. We selected the linear policy with $B = 4$ that seems to be the most resilient towards proper resource allocation, providing a fair replication degree growth factor.

Table 3. Replication policies.

	Parameter c				
Policy	**2**	**3**	**4**	**5**	**6**
Uniform	867	567	44	28	23
Linear	123	77	6	9	16
Quadratic	102	21	42	46	58
Exponential	118	32	19	27	28

Load Resiliency. A good replication strategy must cope with changes on the system load. We vary the global load of the system by changing the mean video size, described in Subsect. 5.1. Assuming the three mean video sizes, namely 20 MB, 30 MB and 40 MB, caching had 1814, 3864, and 7049 violations respectively, while WiseReplica had only 6, 77, and 106. Figure 9 compares the number of violations using WiseReplica and a non-collaborative caching. As the load of the system increases, concurrency in bitrate allocation also increases, causing more violations. WiseReplica outperforms caching mostly because it predicts and prevents useless replication. Therefore, we set to the highest evaluated system load, 40 MB, as the default mean video size workload setting.

Benefits of Prediction on Storage Usage. We aim to adapt the number of replicas to the number of views of a video, especially for the most popular ones. Figure 10 plots the maximum number of replicas for the 1 % most popular videos. Using caching, the maximum number of replicas is high, ranging from 816 to 1367. The Oracle-like assumption allows to decrease significantly the lower and upper limits, to 10 and 190. WiseReplica also reduces the maximum replica range, which is from 19 to 160. More interestingly, the shape of the replication
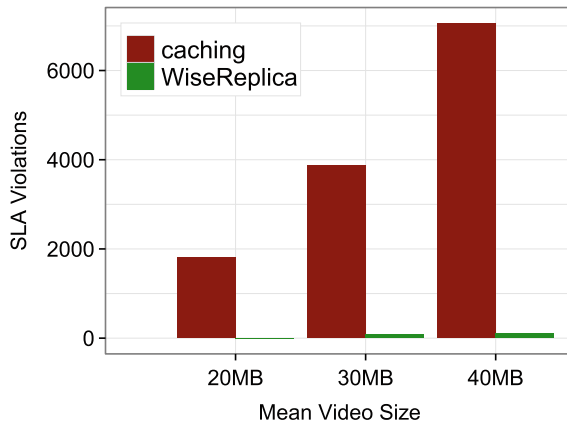

Fig. 9. Mean video size. Higher loads increase the concurrence in network resources, as a result, more violations.

curves of WiseReplica and Oracle-like are quite similar indeed. It confirms that
our predictions are accurate, and that a simple replication policy works properly.

Reducing the number of replicas implies that the systems requires less storage
for replication. Figure 11 shows storage usage for replicas by replication scheme.
Although WiseReplica utilizes more storage than Oracle-like, its usage remains
two orders of magnitude smaller than a non-collaborative caching. The maximum
storage usage for Oracle-like, WiseReplica, and a non-collaborative caching were
34, 85, and 7921 GB respectively. WiseReplica creates more replicas than Oracle-
like because it does not rely on bandwidth reservation to prevent violations. Yet,
WiseReplica maintains replicas efficiently, keeping storage usage very low, and
making cache replacement policies unnecessary. This suggests that the LRU
policy, which would eventually be enforced in peers cache, has no impact on the
WiseReplica performance.

Enhancing Bitrate Provision for Meeting Consumers' Expectation.
WiseReplica performance is also quite similar to Oracle-like regarding preventing
violations. Each point of the Fig. 12 represents the number of SLA violations for
intervals of five minutes. Overall, caching caused 7049 violations affecting 86 %
of all viewers, WiseReplica had just 106 violations, and Oracle-like, evidently,
none. Compared to caching, WiseReplica prevents nearly 99 % of violations.
It copes with violations by (i) creating new replicas for hot videos only, and
(ii) adapting the number of replicas according to the rank position. Vertical
lines in Fig. 12 represent the first access to the 10 videos with the worst content
provision through caching. They account for 80.62 % of all caching violations.
The appearance of these videos puts the system under heavy load, which makes
caching fail to prevent violations.

Figure 13 depicts the average bitrate for viewers of the 10 videos with the
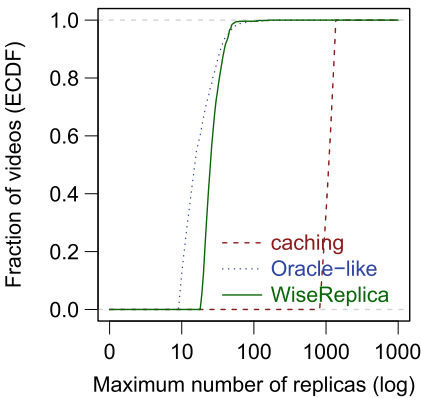worst content provision using caching. When caching was under heavy load,

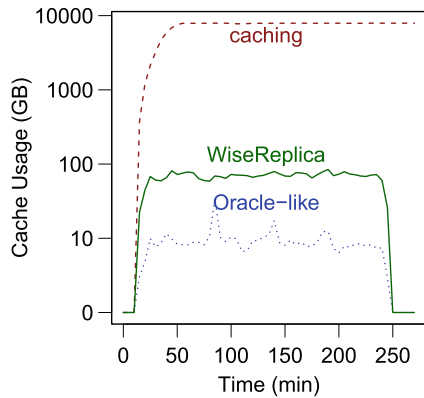**Fig. 10.** The maximum number of replicas for the 1 % most popular videos.
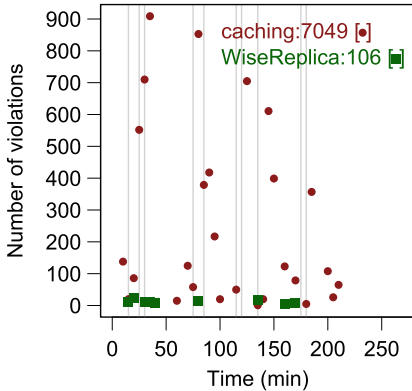
Fig. 11. Storage usage for replication.

Fig. 12. SLA violations. Vertical lines highlight the first view to 10 videos with the worst content provision using caching.
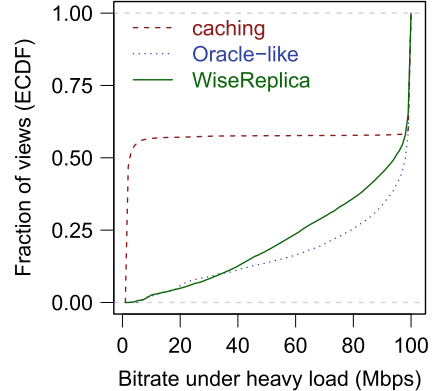
Fig. 13. Bitrate for viewers of the 10 most popular videos under heavy load.

half of viewers experienced a very low bitrate, ranging between 230 Kbps and 2575 Kbps. The mean bitrate with caching was 43 Mbps. On average, WiseReplica improved this bitrate by roughly 85 % under heavy load. Actually it performs almost as well as the Oracle-like assumption, that improved bitrate provision by 93 %. These finds suggest that WiseReplica largely outperforms caching, fairly meeting consumers' expectations under heavy load conditions.

7   Related Work

Our related work is two-fold: Internet videos and adaptive replication schemes.

Internet videos: Recent studies [14, 33] have drawn attention to reach a better understanding of Internet videos properties, such as popularity growth. They point out that well-known popularity characteristics are applicable to multimedia content. For instance, Internet videos popularity distribution follows power law, and popularity bursts have a short duration and are quite likely to happen just after the content publication. Dobrian *et al.* [13] shed some light on the performance of Internet videos provision on CDNs. They show that average bitrate plays an important role in videos availability. A hybrid solution between CDNs and P2P is presented by Mansy *et al.* [23]. Their purpose is to model and analyze a live video system and one of their main concerns is to adapt bitrate for guarantee user satisfaction. Adhikari *et al.* [1] work described the YouTube video delivery system through measurements of DNS resolutions and video playback traces. One of their findings is that over a globally distributed network (Planet-Lab) most part of the nodes have a nearby Youtube video cache server to delivery the video data. Moreover, Brodersen *et al.* [6] presented a detailed study over the strong connection between popularity and geographic locality of Youtube videos.

These facts endure our decision of a locality aware solution for infrastructure. Liu *et al.* [21] make a case for a video control plane that can use a global view of client and network conditions to dynamically optimize the video delivery in order to provide a high quality viewing experience despite an unreliable delivery infrastructure. However, the granularity of their server selection mechanism is at a CDN, ignoring edge resources. WiseReplica addresses this issue by adapting replication close to the viewers. Thus, WiseReplica can be play an important role in collaborating with an Internet control plane.

Adaptive replication schemes: Non-collaborative caching remains the simplest approach to provide popularity-aware replication of web content through cache replacement policies [20]. However, we showed when we adapt the number of replicas according to the Internet video popularity properly, cache replacement policy becomes redundant. EAD [27] and Skute [4] adapt the number of replicas by using a cost-benefit approach over decentralized and structured P2P systems. EAD creates and deletes replicas throughout the query path with regard to object hit rate using an exponential moving average technique. Similarly, Skute provides a replication management scheme that evaluates replicas price and revenue across different geographic locations. Despite presenting an efficient framework for replication, they provide an inaccurate bitrate provision, hence inappropriate for high-quality video delivery. WiseReplica copes with this issue by analysing the request arrival process, performing accurate predictions about the ranking of Internet videos, and maintaining replication degree accordingly.

8   Conclusions

In this work, we presented WiseReplica, a SLA-based, adaptive replication scheme for meeting customers' expectations and enhancing resource allocation in peer-assisted VoD systems. To adapt replication, WiseReplica relies on a prediction model for ranking Internet videos in order of demand. Our intuitive model is flexible, and can learn from different sources and big amounts of data, providing a robust framework for controlling VoD resource allocation. Simulations using You Tube traces suggest that our ranking predictions of videos are important to enhance video delivery in peer-assisted VoD systems, allowing us to self-adapt replication degree to video demand properly. WiseReplica increases the average bitrate provision by roughly 85 %, contributing decisively to enhance viewing experience of users. Our future work will mainly cover a proof-of-concept prototype for evaluating WiseReplica in a real testbed.

References

1. Adhikari, V.K., Jain, S., Chen, Y., Zhang, Z.-L.: Vivisecting youtube: an active measurement study. In: INFOCOM (2012)
2. Alizadeh, M., Greenberg, A., Maltz, D.A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., Sridharan, M.: Data center TCP (DCTCP). In: SIGCOMM (2010)

3. Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., Zhang, H.: Develop-
ing a predictive model of quality of experience for internet video. In: SIGCOMM
(2013)
4. Bonvin, N., Papaioannou, T.G., Aberer, K.: A self-organized, fault-tolerant and
scalable replication scheme for cloud storage. In: SOCC (2010)
5. Braun, L., Klein, A., Carle, G., Reiser, H., Eisl, J.: Analyzing caching benefits
for youtube traffic in edge networks - a measurement-based evaluation. In: NOMS
(2012)
6. Brodersen, A., Scellato, S., Wattenhofer, M.: Youtube around the world: geographic
popularity of videos. In: WWW (2012)
7. Buffoni, D., Calauzenes, C., Gallinari, P., Usunier, N.: Learning scoring functions
with order-preserving losses and standardized supervision. In: ICML (2011)
8. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender,
G.: Learning to rank using gradient descent. In: Proceedings of the 22nd International
Conference on Machine Learning (2005)
9. Calauzenes, C., Usunier, N., Gallinari, P., et al.: On the (non-) existence of convex,
calibrated surrogate losses for ranking. In: Neural Information Processing Systems
(2012)
10. Chang, L., Pan, J.: Towards the optimal caching strategies of peer-assisted VoD
systems with HD channels. In: ICNP (2012)
11. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**, 273–297 (1995)
12. Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in
Practice. Addison-Wesley, Reading (2010)
13. Dobrian, F., Sekar, V., Awan, A., Stoica, I., Joseph, D., Ganjam, A., Zhan, J.,
Zhang, H.: Understanding the impact of video quality on user engagement. In:
SIGCOMM (2011)
14. Figueiredo, F., Benevenuto, F., Almeida, J.M.: The tube over time: characterizing
popularity growth of youtube videos. In: WSDM (2011)
15. Finamore, A., Mellia, M., Munafò, M.M., Torres, R., Rao, S.G.: Youtube every-
where: impact of device and infrastructure synergies on user experience. In: IMC
(2011)
16. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learn-
ing: Data Mining, Inference, and Prediction: With 200 Full-Color Illustrations.
Springer, New York (2001)
17. Huang, C., Li, J., Ross, K.W.: Can internet video-on-demand be profitable? In:
SIGCOMM (2007)
18. Huang, Y., Fu, T.Z., Chiu, D.-M., Lui, J.C., Huang, C.: Challenges, design and
analysis of a large-scale P2P VoD system. In: Sigcomm (2008)
19. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques.
ACM TOIS **20**, 422–446 (2002)
20. Jin, S., Bestavros, A.: Popularity-aware greedy-dual-size web proxy caching algo-
rithms. In: ICDCS (1999)
21. Liu, X., Dobrian, F., Milner, H., Jiang, J., Sekar, V., Stoica, I., Zhang, H.: A case
for a coordinated internet video control plane. In: SIGCOMM (2012)
22. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval.
Cambridge University Press, Cambridge (2008)
23. Mansy, A., Ammar, M.H.: Analysis of adaptive streaming for hybrid CDN/P2P
live video systems. In: ICNP (2011)
24. Montresor, A., Jelasity, M.: PeerSim: a scalable P2P simulator. In: P2P (2009)
25. Parvez, N., Williamson, C., Mahanti, A., Carlsson, N.: Analysis of bittorrent-like
protocols for on-demand stored media streaming. In: SIGMETRICS (2008)

26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. JMLR **12**, 2825–2830 (2011)
27. Shen, H.: An efficient and adaptive decentralized file replication algorithm in P2P file sharing systems. IEEE Trans. Parallel Distrib. Syst. **21**, 827–840 (2010)
28. Silvestre, G., Fernandes, S., Kamienski, C., Sadok, D.: Most wanted internet applications: a framework for P2P identification. In: CNSR (2010)
29. Silvestre, G., Monnet, S., Krishnaswamy, R., Sens, P.: Aren: a popularity aware replication scheme for cloud storage. In: ICPADS (2012)
30. Silvestre, G., Monnet, S., Krishnaswamy, R., Sens, P.: Caju: a content distribution system for edge networks. Technical report, UPMC Sorbone Universités (2012)
31. Steinwart, I.: How to Compare different loss functions and their risks. Constr. Approx. **26**, 225–287 (2007)
32. Bittorrent. http://bittorrent.com
33. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**, 80–88 (2010)
34. Wilson, C., Ballani, H., Karagiannis, T., Rowstron, A.: Better never than late: meeting deadlines in datacenter networks. In: SIGCOMM (2011)
35. Zhang, T.: Statistical behavior and consistency of classification methods based on convex risk minimization. Ann. Stat. **32**(1), 56–134 (2004)