# Hulu in the Neighborhood

Dongsu Han, David Andersen, Michael Kaminsky†, Dina Papagiannaki†, Srinivasan Seshan
*Carnegie Mellon University,* †*Intel Labs Pittsburgh*

***Abstract*—Internet Service Providers (ISPs) are in a constant race to meet the bandwidth demands of their subscribers. Access link upgrades, however, are expensive and take years to deploy. Many ISPs are looking for alternative solutions to reduce the need for continuous and expensive infrastructure expansion. This paper shows that there are many forms of local connectivity and storage in residential environments, and that these resources can be used to relieve the access network load. Making effective use of this local connectivity, however, introduces several challenges that require careful application and protocol design. We present a new system for a neighborhood-assisted video-on-demand service that reduces access link traffic by carefully placing VoD data across the neighborhood. We demonstrate that this approach can reduce the access network traffic that ISPs must provision for by up to 45% while still providing high-quality service.**

## I. Introduction

The delivery of multimedia content is growing at a tremendous rate, increasing 76% every year on average [1]. Cisco projects that "the sum of all forms of video (TV, video on demand, Internet, and P2P) will account for over 91% of the global consumer traffic by 2013" [2]. This growth in demand has been especially hard on connectivity to the home. For example, Korea Telecom's access network traffic increased by 125% annually, compared to 40% growth in its core [3].

These trends are driven by the desire for higher quality video, increase in the number of video titles, and changes in viewing patterns. With digital video recorders (DVRs) and on-line video services like Hulu, users now watch videos at their convenience, and "time-shifted" viewing patterns are becoming the norm [1]. Many predictions suggest that on-demand TV and video on demand (VoD) will largely replace broadcast delivery, except for specific popular live content (e.g., sports events) [4], [5]. This transition is creating painful challenges for today's ISPs whose already-strained access networks were never designed to handle this load.

Unfortunately, upgrading access links is expensive and new technologies often take years to deploy.[1] As a result, access technology can vary dramatically from neighborhood to neighborhood, and even home to home. This heterogeneity means that many users lack the bandwidth they need for the latest applications—particularly VoD—while they wait for an upgrade that is still years away.

There is, however, a promising way to bridge this gap. ISPs can use local storage and local connectivity to mask access bandwidth deficiencies, particularly for time-shifted VoD. ISPs already deploy and control local storage in the form of set-top boxes and DVRs. In addition, although the bandwidth between homes and the Internet is limited by the access network, there are a variety of technologies, such as 802.11 and multimedia-over-coax (MoCA), that provide high-speed connectivity between nearby homes. The combination of storage and connectivity provide an opportunity to stage content in the neighborhood during off-peak periods and share it between homes upon request. These devices and links are relatively cheap, highly capable, and much easier to deploy than new access links. However, the inter-home connectivity provided by such technologies varies across homes, and is hard to provision to meet a target level of service. It is especially painful to manage local resources and to collectively use them to provide a guaranteed service for VoD content.

This paper addresses the question: given a set of local neighborhood resources, how might one build a distributed system to reduce the demands that VoD delivery places on the access network? The system must decide, based on connectivity, the bit-rate of the content, and its popularity distribution, what content to stage at each home to maximize the local sharing of content. Our system faces three challenges: First, the system must monitor and automatically adapt to local resources to use them opportunistically while supporting the bandwidth requirement of the streamed content. Second, the system must be resilient to changes in popularity and environment and adjust the placement efficiently over time. Finally, the system must be robust to failures. The system cannot, for example, rely on any given device being available, since they physically reside in users' homes and are subject to their control. No existing systems and placement algorithms [6]–[9] achieve all these goals simultaneously.

We present the design, implementation and evaluation of a system that successfully addresses these challenges. Our system automatically adapts the content placement to the environment and maximizes local sharing of content using a placement algorithm designed for streamed content. The algorithm effectively uses varying levels of connectivity between neighbors and minimizes overhead due to popularity and environment change. We evaluate the overhead and performance of the system during the entire cycle including bootstrapping, normal operation, flash crowd, failure modes and when the system is adapting to changes. Our evaluation shows that the system can reduce the access network traffic by up to 45% and is resilient to various changes.

The rest of the paper is structured as follows. Section II examines challenges in access networks, and Section III describes the opportunity of using local resources. We present our system design in Section IV, based on a novel place-

[1]For example, the twentieth century only saw four new "pipes" to the home: power lines, phone, cable, and fiber.

978-1-4244-8953-4/11/$26.00 

ment algorithm for streamed content in Section V. Detailed experimental evaluation is presented in Section VI. Finally, we review related work in Section VII, and conclude in Section VIII.

## II. Motivation

This section discusses the challenges faced by today's ISPs.

**Video demand is increasing rapidly.** Two trends in home video consumption increase Internet bandwidth demand: time-shifted viewing patterns and high-definition (HD) content. Time-shifted viewing of TV content through Internet video streaming sites and the use of DVRs account for a large fraction of views, and even exceed the number of broadcast TV viewers [1], [10]. The advent of interactive, "connected" TV such as Google TV will accelerate the trend towards streaming. A similar transition was observed in audio from radio broadcast to Internet streaming [10], [11].

At the same time, HD content is gaining popularity. HD video bit-rates range from 1.5 (Web "HD") to 40Mbps (Blu-ray) depending on the quality. Our system targets high quality movies or TV content at 10Mbps which is similar to the rate of IPTV's HD videos [12], [13]. As home displays get larger, delivering higher quality content is important because users "care less about watching [amateur] YouTube videos on the big screen" than they do about watching high-quality movies [14].

**Access networks cannot keep up.** Current networks are not designed to handle such demand. Figure 1 shows the basic network structure of ISPs. Aside from a pure fiber optic network, current designs rely on statistical multiplexing at the aggregation point exploiting the bursty nature of traditional data traffic. The National Broadband Plan (NBP) [15], for example, assumes 17:1 over-subscription at the aggregation point. However, aggressive over-subscription does not work well with streamed video; a typical DSL network designed to provide broadband Internet service to 384 subscribers can only support 27 (7%) simultaneous streams of 1Mbps video. Therefore ISPs are investing to expand their infrastructure, which carries significant cost and typically spans many years. Cable providers invested $161 billion from 1996-2009 [16], and Verizon's $23B investment on fiber spans several years [15].

Access networks are especially hard to upgrade because 1) most access network equipment resides outside the plant; and 2) the number of DSLAM and cable nodes far exceeds the number of COs or headends.[2] For example, the NBP allocated less than 5% of the total cost to upgrade the connectivity between the COs because they are relatively well-provisioned, while most cost is associated to upgrading access networks.

Although techniques such as content distribution networks (CDNs) and peer-to-peer streaming provide scalable media delivery in the core network, access networks do not benefit from them. In this paper, we propose an alternative paradigm to overcome limited access network bandwidth to deliver on-demand content within an ISP network.

[2]The number of headends in the US is 7,853 [16] Based on subscribers per cable nodes and per headends the number of cable nodes is roughly an order of magnitude lager than the number of headends.

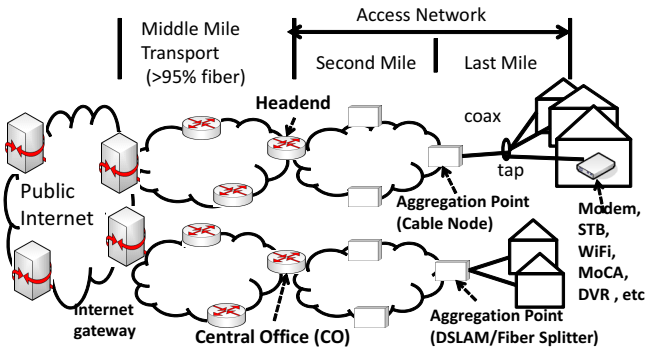


Fig. 1. Basic network structure [15]

## III. Neighborhood-Aware Networks

Although major network upgrades are infrequent, deployment of in-home devices such as home routers, set-top-boxes (STB), and DVRs has seen explosive growth. ISPs often install and connect these devices to customers' home networks using technologies such as WiFi and Multimedia over Coax Alliance (MoCA). Such technologies, originally designed for connectivity *within* a home, can also provide connectivity *between* homes. We argue that this exemplifies a more fundamental property that deploying a network with local connectivity is much easier than upgrading the entire access network. In this paper, we therefore describe a distributed system architecture that permits cooperative use of local resources to augment the infrastructure. In this section, we explore two key aspects of local resources: ISP's deployment and control over such resources and inter-home connectivity.

**Deployment and control.** ISPs already deploy many types of home devices in large scale, and in some cases tightly interact with them. Previous studies have shown that a significant fraction of customers have ISP-provided WiFi APs [17]. MoCA, also driven by a service provider deployment model, shipped over 20 million units to date [18]. Cable operators have installed DVRs in more than 30% of American homes [19]. Deploying and installing these boxes is already a part of ISPs' operational cost and in-home equipment cost is a fraction of the cost to install fiber to the home.[3] With services such as remote recording, VoD, IPTV, and triple play, ISPs often interact with and exercise control over the software and resources (CPU, storage, network) of in-home devices.

**Inter-home connectivity.** We expect the storage and communication capacity of home devices will continue to grow, and inter-home connectivity will be even more viable if technologies are designed with inter-home connectivity in mind. However, diverse forms of high-bandwidth, inter-home links exist even with today's in-home devices.

*Local wireless:* Previous studies [17], [20] show that 802.11 APs are densely deployed in suburban neighborhoods, and wireless signals often reach several neighbors. Such connectivity has been used to provide both commercial and non-commercial Internet service [21], [22]. We examined two sets of data to look at wireless connectivity between neighbors.

[3]Verzion's cost to deliver a fiber per subscriber is $2,125 in 2010 [15].

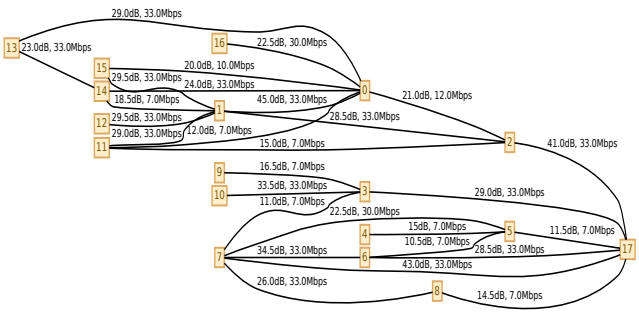


Fig. 2. Topology of a neighborhood mesh network

First, we performed a measurement study using NetStumbler in 16 different homes around Pittsburgh, Portland and Boston. The result suggests that more than 75% of homes can reach two or more neighbors at 10 Mbps or faster. Detailed results are presented in our technical report [23]. Second, we examined a mesh network deployment in a residential area (for Internet access) by Meraki. Figure 2 shows the SNR measurements and estimated throughputs of 802.11g using results from [24]. Relatively high throughput can be achieved between neighbors. With 802.11n technology, the range and throughput is expected to improve significantly.

*Local wired:* Recent advances in home networking use unused high frequency bands on existing in-home power-line and cable TV (coax) infrastructure to provide high bandwidth connectivity for in-home devices. Wall-plugged power-line communication devices deliver up to 200Mbps; MoCA delivers up to 135Mbps per channel using high frequency ranges that cable providers do not use. These technologies can reach well beyond a single house (e.g., 300m for MoCA); cable and electrical wires are naturally connected when the copper from the provider splits to multiple homes [18].[4] Indeed, many of these technologies consider interference between homes in their design and have multiplexing mechanisms such as channel selection as well as security codes. Sometimes providers take explicit action to *confine* their signals to a single house, placing filters at demarcation points. Allowing interconnection between customers instead is often straightforward. The same connectivity gains apply to larger multi-dwelling units that share even more infrastructure (such as in-building Ethernet) in closer proximity. In some cases, the broadband connection itself provides the local connectivity such as an advanced DSL network where the DSLAM has switching capability [25]. [5]

## IV. Design

Despite the richness of local resources, currently they are used independently. In this paper, we explore the other end of the spectrum where local resources are cooperatively used. We propose making software changes to in-home content storage devices, such as DVRs, so that they can store and share content with their neighbors using local connectivity. As a result, only the content that is not available or cannot be delivered on time from a user's neighbors is served from the remote VoD server. In this design, ISPs proactively push content to the DVR disks using broadcast/multicast during off-peak periods for the network. This approach has the potential to reduce the amount of bandwidth that the ISP needs to provision at the aggregation point of the access network while still providing the same level of service as traditional VoD.

[4]In cable TV, coax cables from 2–8 homes are connected by a single tap.

[5]While our system design accommodates various forms of local networks, we focus on using 802.11 and MoCA in our experimental evaluations.

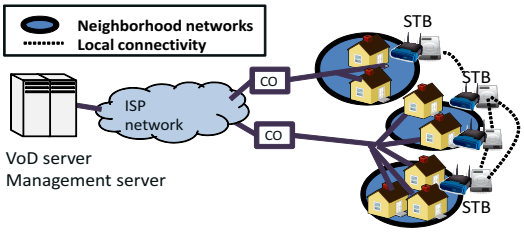


Fig. 3. Overview of the system

However, resources such as local connectivity vary across homes, and it is hard to provision local resources to provide a guaranteed level of service. Furthermore, managing these devices may be painful. Thus, we propose a system that automatically adapts to the amount of local resources and uses them opportunistically to minimize management and operational cost. Next, we present the specific requirements of this design and outline the system components.

### A. System Requirements

The system must address four challenges: First, the placement must take into account the local connectivity between neighbors in a particular neighborhood, the popularity of different content, and the possibility of failures in the local network. Second, the system must keep the content of the DVR disks up-to-date to remain effective despite popularity changes, new content introduction and connectivity changes. Third, the system must continually monitor what inter-home connectivity options are available to inform the placement algorithm and decide dynamically at run-time how to retrieve content from neighbors. Finally, the system must be robust to mishaps such as device failures and popularity misprediction.

### B. System Components

Based on the requirements outlined in the previous section we now describe a VoD system that uses local resources and supports a large number of videos (>10,000) comparable to that of Netflix online.

Our proposed neighborhood VoD system has three basic (physical) components (Figure 3): centralized video servers, a centralized management server, and per-home STBs connected locally to one another and by broadband. The *video servers* store the master copy of all content that the system provides, and the *in-home STBs* have local replicas of popular content distributed across their disks. The *management server* serves two functions: placement generation and directory service. Placement generation involves computing the desired placement of the content in the neighbor-nodes, given system parameters such as the viewing patterns of videos and the

connectivity between neighbors. The directory service maps content to a set of replicas (homes) within the neighborhood.

With this set of physical resources, the logical flow of system operation is as follows:

1) The *placement generator* computes the ideal replication of content across STBs based on current neighborhood connectivity and popularity of movies.
2) The *replication service* uses off-peak bandwidth to download content to the STBs to achieve the target placement.
3) The *transfer service* dynamically decides where to fetch content given users' request and satisfies a significant fraction of the request using local connectivity and content stored on neighbors' STBs.
4) The *monitoring system* monitors user requests each day to update connectivity and popularity information.

We assume that STBs may fail due to various reasons, but remain powered-on in normal operations; they can go into energy-saving mode when they are not being used, but quickly become fully functional when needed. In the following subsections, we describe each of the four logical components of our system: the placement generator, replication service, transfer service and monitoring system.

### C. Placement

For better system efficiency, each video file is partitioned into *chunks* that are placed across the neighborhood according to the placement given by the placement generator. The objective of the placement is to minimize the peak ISP bandwidth use (since peak bandwidth determines network buildout) by maximizing local consumption of content within the neighborhood. This is challenging for several of reasons.

First, a group of neighboring STBs should hold as much content as possible since the size of a movie library is usually much larger than what local storage can hold. 10,000 one-hour HD movies take up more 40TB. Second, because the local bandwidth varies widely from home to home and may be less than the content's playback rate, the placement must consider local connectivity and the bandwidth requirement of the streamed content. A local copy of content (chunk) might not be delivered before its playback time unless it is carefully placed considering each node's connectivity. This should be avoided because the system directly downloads required chunks from the ISP's video server if local resources cannot keep up. Third, the placement should automatically adapt to changes such as new content, node addition, connectivity changes, and content popularity changes. While placement reconfiguration is needed to adapt to such changes, the system should minimize the overhead of reconfiguration for efficient operation. Finally, the placement must be robust against a range of problems. The estimates of connectivity and popularity may be inaccurate. The STBs may be powered down or fail for other reasons despite running software specified by the ISP. The performance of the computed placement must not be overly sensitive to these factors. In Section V, we present a placement algorithm that effectively deals with the challenges described above.

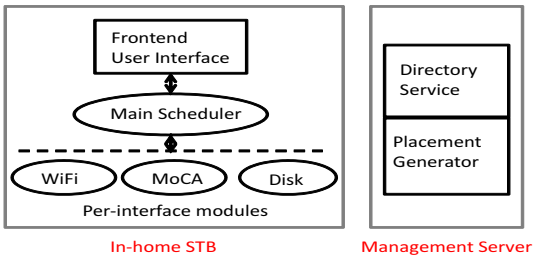


Fig. 4. Software components of the system

### D. Replica Management

Content popularity and node connectivity change over time. Content popularity shifts with users' interests and as new content is introduced. VoD systems such as Hulu introduce about 20 new episodes every day, and studies have shown that user interest changes over hours, days and weeks [26]. Less frequently, connectivity may change as a result of node addition, node removal, or environmental changes in the neighborhood. Unfortunately, the target placement may change as a result of these popularity and connectivity changes.

With these changes, the system should move content between STBs as quickly as possible to remain efficient. This may involve movement within the neighborhood and pushing new content. Note that a one-hour movie encoded at 10Mbps takes up 4GB, and the number of movies the system supports is very large. As a result, moving a significant number of movies can place a very high load on the network.

We minimize the impact of content movement using broadcast channels and excess capacity in the local network. When the target placement changes, each STB is notified of its new placement, and evolves towards the new configuration. When content has to move between neighbors, each STB uses local connectivity during off-peak hours. The replication service handles the rest and pushes (new) movies to the neighborhood using broadcast in order of their popularity. This allows us to effectively accommodate the time-shifted viewing pattern and reduce the peak bandwidth as we will see in Section VI.

### E. Transfer

The software on each STB consists of a *frontend*, a *main scheduler* and *per-interface modules* (Figure 4). When a request for a movie arrives at the main scheduler it first contacts the *directory service*, which handles name resolution.

Entities that need to be named are movies, chunks and nodes, but our system is oblivious to how they are named. In our implementation, movies have ISP-defined IDs, chunks have content-based names, and nodes are named by the numerically lowest MAC address of any of their interfaces. The directory service provides a movie ID to metadata resolution. The metadata contains information such as the size of the movie, list of chunks, chunk size and encoding rate. The directory service also provides chunk ID to location resolution where the location contains the nodes that store the chunk.

In response to a movie request, the main scheduler obtains a list of chunks and their locations from the directory service. The scheduler first determines which chunks need to be

fetched remotely. Then, given the local connectivity bandwidth information and the received playback time of each chunk, the main scheduler dynamically decides which interface to use to request a chunk. The scheduler makes remote requests to neighbor nodes through per-interface modules which provide a uniform API to the local disk storage and to each local network connectivity option that the STB has. In situations where content is not available locally or local bandwidth is insufficient, the scheduler requests the content directly from the ISP's video server.

### F. Monitoring

The key to having our system adapt over time is the ability to monitor the neighborhood for changes in connectivity and viewing patterns. Centralized components in our systems, such as the directory service, make it easy to keep logs of viewing patterns. Keeping track of connectivity is somewhat more complex. Each network interface uses a unique IP address derived from its MAC address and starts a service discovery process that discovers neighbors through periodic broadcast or multicast. When a neighbor is discovered, a point-to-point connection is established between the two neighbors. The system updates the bandwidth between each neighbor when a chunk is transferred.

## V. Placement Algorithm

As noted in the previous section, the placement must reduce the ISPs bandwidth and be robust against changes and failures in the system. To this end, we want the neighborhood nodes to collectively store popular movies with each node storing different chunks of such movies. The placement algorithm takes three parameters as input: content popularity, neighborhood connectivity, and user viewing patterns. It outputs a mapping from a chunk to nodes where it should be placed.

Our solution exhibits three key properties: 1) Once a movie is placed in the neighborhood, every neighbor can instantly stream the movie without fetching the content from the remote ISP server. 2) To support instant playback in case of random seek and jump, minimal ISP bandwidth is used initially, but the content is streamable using local resources. 3) The overhead due to popularity changes and node failures is minimized. The first two requirements for streamability are crucial in supporting streamed content and in making efficient use of the access network by broadcasting the popular movies only once to stage them.

Using a mixed integer program (MIP), we first determine how each chunk should be placed to guarantee the streamability requirements, while minimizing the storage use. We then use the resulting chunk placement to construct the entire placement.

**Optimization.** Let $B$ be the matrix that represents the aggregate bandwidth between peers, where $B_{i,j}$ is the bandwidth from node $i$ to $j$ (the diagonal elements are set to infinity). A movie is divided into $m$ chunks. $x_{i,j}$ is a zero-one indicator variable that indicates whether node $i$ has chunk $j$, and $y_{i,j,k}$ is a variable that indicates whether node $i$ should request chunk $j$ from node $k$. The optimization problem then is:

*Minimize* $\sum (\sum_j x_{i,j})^2$ (storage factor)
*such that* $\sum_k \sum_j y_{i,j,k} = m$ ($m$ chunks requested)

$$\forall i, \forall j, \sum_k y_{i,j,k} = 1 \quad \text{(A chunk is requested exactly once)}$$

$$y_{i,j,k} \leq x_{k,j} \quad \text{(availability constraint)}$$

$$\forall i, \forall j' \frac{j'}{\sum_{k \in Node} \sum_{j \leq j'} \frac{y_{i,j,k}}{p\, B_{i,k}}} \geq R \qquad \text{(bandwidth/streamability)}$$

where $p$ is the fraction of local bandwidth that a node can use. It reflects the expected number of concurrent users that share the local links and may view the local content simultaneously. It is determined as $p = 1/\lceil$ {number of neighbors in a single broadcast domain} $\times$ {fraction of active users at peak} $\times$ {hit rate of movies in the neighborhood} $\rceil$ through iteration.

We minimize the squared sum of storage use instead of the linear sum to spread chunks more evenly to neighbors for resilience to node failures. We call this term *storage factor*. The streamability constraint guarantees that all neighbors can stream the content placed in the neighborhood given the bandwidth constraint.

**Placing content.** The MIP output spreads chunks across the neighborhood while allowing every neighbor to stream the movie instantly without having to use the broadband link. We next replicate this solution across multiple movies. Beginning from the most popular movie, we select movies from the top of the popularity list and start filling the disk until disk space on one of the nodes runs out. At the end of this round, anyone can watch the $N$ most popular movies using only local resources. We call the set of movies that use the same solution from the MIP a *placement group*.

At this point, some nodes may still have disk space available because the number of chunks that each node stores may differ. We then iterate this process by revisiting the MIP. This time we eliminate the nodes whose disks are full, and solve for the next placement group. We apply this across the next most popular movies. After a few iterations, all the disks in the neighborhood are filled.

The number of movies each neighbor can access through local resources varies with its connectivity to others(fan-out and bandwidth), and the placement group reflects this. Movies in the first placement group are locally available to all neighbors, but the availability decreases in the subsequent placement groups. Our placement is oblivious to the hit rate of each movie, but sorts groups of movies in order of its popularity. We do not have to re-run the MIP when the movie popularity changes, but placement of content can change. The placement for a neighborhood can be represented in a compact form with the number of movies in each placement group and the chunk mapping for each placement group.

**Number of chunks.** We carefully choose the value of $m$ (number of chunks) to provide several desirable properties. A large value of $m$ divides a movie into smaller chunks,

which provides more flexibility to the MIP in finding a more efficient solution. On the other hand, larger chunks (smaller $m$) reduce the variables in the MIP, making it faster to solve. As a guideline, we recommend that $m$ be larger than the number of nodes $N$. In practice, a movie may contain several thousand chunks, e.g., a one-hour movie encoded at 10Mbps has 9,000 512KB-chunks. To use a smaller value for $m$, say 30, we treat the movie as if it consists of 300 groups of 30 chunks where each chunk group represents a subsection of a movie.

**Support for random seek.** This grouping helps support random seek/jump during playback. The MIP ensures that no ISP bandwidth is used when the playback is started from the beginning of each chunk group. E.g., our implementation uses fixed size 512KB-chunks and $m = 30$, which translates into 12 second playback time for a chunk group. This means that even if a user jumps to a random location, the ISP bandwidth is only going to be used to deliver at most 30 chunks or 12 seconds in the worst case and 6 seconds in the average case. Each group of chunks can be further assigned different popularity, e.g. when a clip from a movie becomes very popular.

**Minimizing overhead.** Changes in content popularity and node connectivity result in a new placement. For the system to be practical, such changes in placement should be minimized to avoid large overheads of transferring content. Our placement already exhibits some of this property as placement configuration remains the same within the same placement group. However, when the placement group changes for some movies as a result of popularity change, such movies need to change the placement configuration. We further reduce this overhead by relaxing some constraints.

We minimize the number of placement groups so that more movies have the same configuration and also minimize the content shift by minimizing the difference between adjacent placement groups. Full details of the algorithm are presented in the extended technical report [23].

## VI. Evaluation

We now evaluate our neighborhood VoD design and implementation. The evaluation centers around three questions:

- How effective is the placement algorithm in reducing the ISP's network bandwidth and how robust is it to failures or mispredictions? (Section VI-B)
- How much bandwidth is required to replicate or move content into and within the neighborhood to account for changes in popularity, connectivity, and node availability? (Section VI-C)
- How sensitive is our system to its operating parameters? (Section VI-D)

### A. Evaluation Setup

The software components of the in-home STB, a prototype directory server, placement generator in Figure 4 and a video server are implemented. Exhaustive implementation details are presented in [23]. We first evaluate our system using a nine-node testbed. The testbed provides experimental results on a neighborhood-scale deployment using real hardware. Then, these results are used to parametrize our simulator, which can scale the experiments up to the 500–1000 homes that a CO or a node in a cable provider's hybrid-fiber-coax (HFC) network might serve. As such, the simulator mimics the overhead such as medium contention experienced by the real testbed.

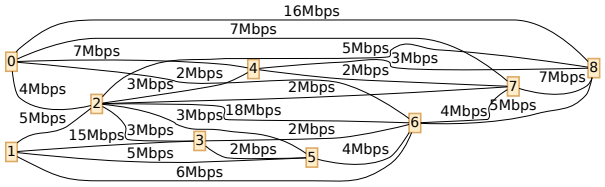


Fig. 5. Wireless connectivity of the testbed

**Testbed.** To emulate a neighborhood, we deployed nine nodes spread across an office building, plus a video server. In this neighborhood, every node is equipped with MoCA and WiFi, and has a 10Mbps downlink from the video server. Groups of four (nodes 5 to 8) or five nodes (0 to 4) are connected by MoCA at 100Mbps. Wireless connectivity between two nodes varies from 0 to 18Mbps, similar to the that between well-connected neighbors (see Section III). Figure 5 shows the wireless bandwidth between the nodes.

**Video content.** We emulate a video content library containing 10,000 one-hour videos, each of which is encoded at 10Mbps. This library is similar in size to the number of on-demand videos in NetFlix. Each node in the neighborhood has a 1TB disk which can hold approximately 233 such videos. As with prior work, we use a Zipf-like distribution with a skew factor $\alpha = 0.3$ to represent the popularity distribution of videos in the library [26], [27]. In a Zipf-like distribution, content popularity (P) is related to its rank (r): $P_r \sim \frac{1}{r^{1-\alpha}}$

**Viewing pattern.** Multimedia viewing varies diurnally with a "prime time" peak. We use a fixed probability distribution that represents this behavior with a 24 hour period to simulate the arrival of video requests. The shape of the distribution is based on the findings by Qiu et al. [28]. We assume that the probability that a home requests a video at prime time is 40%. This probability gradually falls to 10% in the next 12 hours and comes back up to 40% in 24 hours. The videos thus requested are sampled according to the Zipf-like distribution mentioned above. The same video library and workload is used for the testbed experiments and simulations.

**Metrics.** We use three metrics for evaluation: average, peak and 95th percentile access network bandwidth at the second-mile link of Figure 1. The peak bandwidth is more important because the peak determines the amount of bandwidth that ISPs have to provision. However, if the peak is short-lived and users are willing to accept small delays, it may not be as meaningful. Consequently, we also report the 95th percentile, typically used for charging purposes.

### B. Placement Evaluation

We begin by running the placement algorithm for the video library and experimental testbed described above. The resulting placement had two placement groups, which contained 952 and 159 movies respectively. We then measured the bandwidth

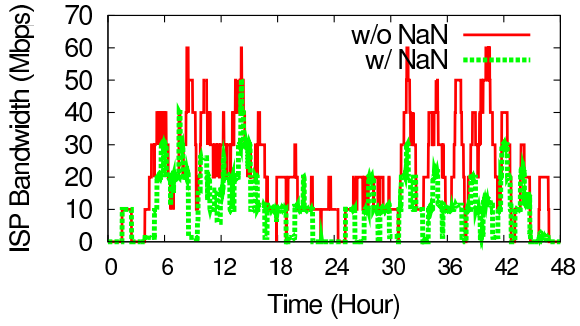


Fig. 6. 48 hour ISP bandwidth usage

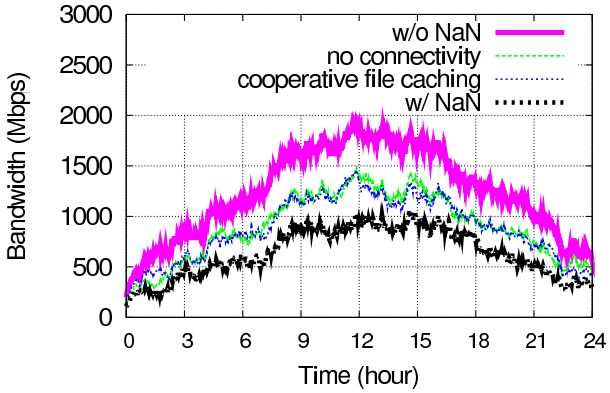


Fig. 7. 24-hour ISP bandwidth use of 500 neighbors

savings on the access network over a 48-hour period. In the 48 hour period, 102 hours of movies are consumed in total, and each house watches 8 to 18 hours of content.

*1) Bandwidth savings:* Figure 6 shows the broadband bandwidth use over the 48 hour period. We refer to our system as *NaN* and *w/ NaN* shows the performance of our system. *W/o NaN* shows the amount of bandwidth that would have used if all the content was coming from the ISP through broadband.

The average second-mile link bandwidth with NaN and without NaN is 10.1Mbps and 21.2Mbps respectively. Our system reduces the average bandwidth by 52%. The peak bandwidth usage is reduced by 16% (from 60Mbps to 50.3Mbps), while the 95th percentile bandwidth usage is reduced by 40% (from 50Mbps to 29.9Mbps).

The peak bandwidth savings in the nine node testbed was small because of its scale. Although the top 1111 movies account for 50% of demand, the hit rate is not always 50% in this small neighborhood. For example, at around the 13th hour, five out of the six movies requested (over 80%) were not stored in the neighborhood and had to be requested from the video server. In practice, however, a single second-mile link serves hundreds of homes, and the probability that 80% of the nodes simultaneously request unpopular movies not stored in the neighborhood would be much smaller. Thus, the peak bandwidth in this case will be much closer to the average (due to the law of large numbers).

To get a better understanding of the ISP's bandwidth savings, we simulated a larger neighborhood of 500 homes with different placement policies. Figure 7 shows the aggregate bandwidth demand for a range of different placement policies for content in the neighborhood. The *w/o NaN* line represents a system where neighborhood nodes store nothing. The *no connectivity* line represents a system where each node stores the 233 most popular movies (i.e. they only fetch from their local disk or from the ISP and use *no connectivity* to their neighbors). The *cooperative file caching* line represents a system where files (movies) are sorted in terms of popularity and each file, in order, is assigned to a node in the neighborhood until all space is exhausted. This represents a cooperative caching of web content that is connectivity-unaware and agnostic to streamed content. This places as much content in the neighborhood without considering the limitations of local connectivity. The *w/ NaN* line represents our proposed content placement that considers both popularity and connectivity. There are several noteworthy observations from this simulation. First, pushing content to the neighborhood provides significant benefits – all designs do much better than w/o NaN. Second, being popularity aware is critical. Popularity unaware placement (simply placing 11% of the library—1111 randomly chosen movies— not shown on graph) results in only a 25% improvement over w/o NaN. Third, the fact that NaN significantly outperforms no connectivity shows taking advantage of connectivity provides significant benefit. Fourth, the difference between the NaN and cooperative file caching lines show that the system must monitor neighborhood connectivity and place content carefully considering the streamability requirement. Our approach (the NaN line) reduces the peak bandwidth demand at the second-mile link by 42% (from 1980Mbps to 1143Mbps), the 95 percentile bandwidth by 45% (from 1890Mbps to 1037Mbps), and the average bandwidth by 45% (from 1210Mbps to 670Mbps). Note that this verifies our assertion that the peak and average savings are likely to be similar in larger neighborhoods.

*2) Placement Robustness and Fault Tolerance:* The *transfer service* masks failure of local resources by fetching unavailable chunks from the ISP's server. This results in degraded performance. We now evaluate the system when there are content popularity mispredictions, unexpected node failures, and link failures. Here, we are interested in performance assuming that the system does not re-run its placement algorithm to find a new placement. Our results show that the system performance is not overly sensitive to such failures.

**Popularity misprediction.** So far we have assumed that the popularity distribution is known in advance. Although a number of VoD studies [26], [29] suggest that popularity in the near future is predictable based on the popularity in the past, mispredictions can happen. To quantify the effect of mispredictions on the system performance, we measure the second-mile link bandwidth savings assuming that some fraction of the top 1111 movies are actually mispredicted. Specifically, we randomly shift the rank of the mispredicted movies by up to 10 times the original rank. For example, what was predicted to be the 10th most popular movie can actually be any one of top 100. The result show that the system is not overly sensitive to popularity mispredition. When 30% of 1111 movies have a mispredicted popularity rank, the aggregate hit-rate decrease by 7%. When 40% are mispredicted, the system retains 40% of bandwidth savings. We ran a 24-hour

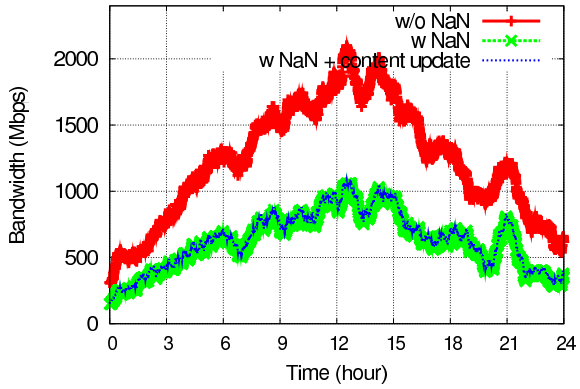


Fig. 8. 24-hour ISP bandwidth use of 500 neighbors

experiment with the 30% misprediction rate in our testbed neighborhood. Our results confirmed a similar 6% reduction in average bandwidth savings. However, the bandwidth saving was still significant at 41%. In Section VI-C, we also explore the case where the system is not in sync with the target placement.

**Node failure.** To evaluate how a node failure affects performance, we failed a node at hour 8.5 of a 24-hour experimental run. We compare the access network bandwidth use at the second-mile link in this case, where the node fails but the placement had been calculated as if the node were available, with the case where the placement had been calculated taking the failed node into account. The failure-aware placement provides an additional 9.2% bandwidth reduction, which is similar to the fraction of node down time (7.2%). The impact of failures on performance is relatively small because the placement algorithm effectively spreads out the popular content across all nodes.

**Link failure.** To evaluate how a link failure affects performance, we fail *Node 0*'s MoCA link. The node must therefore rely on its wireless connectivity to transfer data within the neighborhood. We again compare the bandwidth reduction achieved with i) the original placement plus the link failure and ii) an alternative placement where the failed link never existed. Here, the original placement would 8.6% smaller bandwidth saving than the alternative placement (where the link had never existed). The single link failure had such impact because over 80% of the total local transfer among the nine nodes happen over the MoCA links.

### C. Evaluation of Replica Management

In Section VI-B2, we evaluated the system's robustness to popularity mispredictions, node failures, and link failures. That is, given the change, how does the system perform without re-running the placement. This section discusses how the system moves towards a new placement and how much bandwidth is needed to do so when content popularity and connectivity changes or in bootstrapping new nodes.

*1) Adapting to content popularity change:* As the popularity of content changes over time, the content placement should be changed. In this section, we quantify how much data transfer is necessary to achieve the new target placement after a change in popularity.

The popularity change can have two sources: new content and a shift in users' interests. Given a new set of popular movies, the system re-runs the placement algorithm to compute a new placement. For new popular movies entering the system from the video server, the amount of data that needs to be transferred is proportional to the amount of new content. The specific degree of replication and placement of chunks depends on the movie's placement group. When the popularity of an existing movie changes, the placement will need to be adjusted if the movie moved across placement groups.

**Using broadcast.** We use broadcast to replicate new data or to shift existing data. Cable and IPTV providers, for example, can use one of their broadcast channels to send data, and STBs and DVRs usually have multiple TV tuners so each neighbor can tune in to multiple channels at the same time to download and/or prefetch data in the background. The nodes will store only the content suggested by the new placement. For networks and devices that can not support additional channels, the device can opportunistically tune into the broadcast channel, when its link to the ISP is idle, and record titles that it should be storing locally. Since each neighbor watches the video up to several hours a day, the link is idle most of the times. The cost of the broadcast channel is amortized across many neighbors, as we quantify later.

According to a large-scale VoD study [26], the membership of the top 200 movies changes by 5% to 35% (20% on average) over one week. Thus, we produce a new movie popularity distribution by randomly changing the rank of a set fraction of the movies. We vary the fraction from 5% to 50%, and re-run the placement algorithm. New movies or movies that shift to a different placement group require data transfer. The amount of unique data determines the time it takes to transfer the content using a broadcast channel shared by all users. The result shows that a 10Mbps broadcast channel can accommodate up to 17% change.

We evaluate a more extreme scenario where the top 35% of popular movies are replaced by new content in a given week. Since a 10Mbps broadcast channel can only re-supply 15% of total content stored in the neighborhood in a week, the remaining 20% are not going to be in the neighborhood. However, because the replication service prioritizes popular content, movies that are not locally stored are relatively less popular. In this case, our 24-hour experimental result shows that the average access network bandwidth saving is 43%. This bandwidth reduction is only 7 percentage points lower than the desired target placement because the remaining 20% of new popular movies must come from the video server. Thus, even with a major shift in popularity, the system remains effective by prioritizing the popular content.

**Overall performance.** In the steady state, the ISP's bandwidth is used for on-demand content that is not stored in the neighborhood and for replicating new content into the neighborhood to accommodate popularity changes. We simulate a neighborhood of 500 homes, in which one channel is allocated for demand fetching and two additional channels are dedicated

to pushing new content into the neighborhood. The results are shown in Figure 8. *w/o NaN* shows the amount of bandwidth used to serve 500 homes using only the video server. *w/ NaN* is the bandwidth demand when the content is pre-seeded according to the target placement. *w/ NaN + content update* shows the aggregate bandwidth of *w/ NaN* plus two broadcast channels used to update the neighborhood storage. Note that these additional pre-fetch channels do not add to the ISP's costs as they only affect average bandwidth consumption; in fact, they trade average bandwidth for a dramatic reduction in peak bandwidth: Our approach reduces the peak bandwidth that ISPs have to provision by 44% (from 2100Mbps to 1159Mbps), even with the two broadcast channels. With ISP dynamic channel allocation, ISPs can deallocate the broadcast channel during the peak hours and achieve 45% savings in peak bandwidth. The 95 percentile bandwidth was reduced by 45% (from 1890Mbps to 1037Mbps), and the average bandwidth by 46% (from 651Mbps to 1197Mbps).

*2) Adapting to connectivity changes:* The connectivity between nodes may change over time as a result of network reconfiguration or environmental changes. These changes do not happen daily, but may occur during the lifetime of the system. For example, network devices may be upgraded, new buildings may degrade wireless signals, and added splitters may degrade MoCA signals.

Although we do not try to adapt the placement to small variations in connectivity, the placement has to adapt when connectivity between nodes changes significantly. To quantify the amount of data movement caused by placement shifts from connectivity changes, we decrease or increase the bandwidth between every two nodes by up to 50%. When the connectivity is decreased, the streamability requirement is no longer met, so more replicas are needed across the neighborhood. With 50% decrease in bandwidth, bandwidth equivalent to 4.5 days of a 10Mbps channel is needed to supply the additional replica. When the connectivity is increased, the placement does not need to change but the current placement may have too much replication for the new connectivity. In our testbed, the new target placement would now allow to store an additional 44 movies in the neighborhood. This lost opportunity could provide an additional 0.6% bandwidth savings on the access network. In our implementation, if this opportunity cost exceeds a certain threshold (e.g., 5%), we treat the case similar to the new node arrival case described below.

*3) Node additions and bootstrapping:* To evaluate system bootstrapping and incremental deployment, we evaluate a scenario where a new node enters the system. We start with eight nodes in the testbed and add a new node. We use the same topology from Figure 5. Nodes 0 to 3 and nodes 5 to 8 form MoCA groups connected by coax cables. Node 4 enters the system and joins the first MoCA group.

When the new node enters the system, we assume (worst case) that its disk is initially empty (the ISP could alternatively ship the disk seeded with some popular content). The new node joins the existing network and discovers neighbor nodes. The new node measures the network throughput to its neighbors and reports the result to the ISP's management server. The placement generator then generates a new placement based on the existing placement. Because the placement algorithm tries to evenly distribute chunks for each movie, some chunks from existing nodes move to the new node. This reshuffling is already minimized by the placement algorithm and can be accommodated by the excess capacity in the local network when it is not being used. As a result of this movement, each node now has some empty space which the system fills with the next most popular content. Previously, this content was not available in the neighborhood, so it has to come from the ISP.

The total amount of data transfer from existing nodes to the new node using local connectivity (MoCA and wireless) is 579GB, requiring 13.2 hours to transfer. The amount of new content from the ISP is 950GB, which requires approximately 9 days to transfer via a single 10Mbps broadcast channel.

### D. Sensitivity Analysis and Abnormal Behavior

So far we have focused on addressing the first order challenges in designing a neighborhood network VoD system. However, there are additional problems that need to be considered in deploying such a system. In particular, we provide sensitivity analysis with varying level of connectivity and disk space, compare the performance of our approach with that of other solutions, and address abnormal events such as flash crowd events using a small on-demand cache. Detailed results are presented in the extended technical report [23].

## VII. Related Work

Our VoD system makes opportunistic use of the local neighborhood networks to augment the traditional ISP-managed infrastructure. The system builds on previous work in multicast-VoD and peer-assisted VoD.

**Muticast-VoD.** Traditionally, periodic broadcasting schemes [30]–[32] have been used to serve very popular content by broadcasting that content continuously on several channels. Prefix caching [33], combined with these schemes, can reduce the number of broadcast channels [34], [35]. Similarly, caching has been used in conjunction with multicast to disseminate popular content [36], [37]. These schemes typically require very frequent accesses to popular content to batch multiple requests or support a relatively small number (20–40) of popular videos due to various limitations [35]; the system falls back to unicast for the remaining content. Our system, by cooperatively using local storage and connectivity, makes a different and more radical bandwidth trade-off appropriate to today's environment.

**Peer-assisted VoD.** Many Peer-assisted VoD studies take a content provider-centric view whose goal is to reduce the provider's bandwidth, using overlay multicast [38]–[40] or peer unicast [41]–[44]. Recent work combines the P2P and CDN approaches to deliver video streams [45], [46]. Others, who take an ISP-centric view [4], [6], [8], [9], [47], [48], also propose an ISP-driven deployment of ISP-controlled in-home storage, and explore content pre-placement and caching

strategies in DSL networks. Suh et al. [8] propose pushing the content to homogeneous peers in a DSL network and present placement algorithms and theoretical analysis. NaDa [9] demonstrates the approach's effectiveness in saving energy cost. Borst et al. [6] propose a cooperative caching algorithm to minimize the total network footprint using hierarchical caches. Our approach, instead, uses varying level of local connectivity to limit the bandwidth demand at the access network. Our approach can also coexist with caches at different parts of the network.

CPM's [4] primary goal is to reduce the server bandwidth. It combines on-demand multicast with peer assisted unicast. Our primary goal is to reduce the access network bandwidth using alternative connectivity. We did not use broadband peers to avoid consuming access network bandwidth. On-demand multicast is not effective in our setting because two neighbors watching same movie, at a given time, out of 10,000 movies in access networks' scale is very rare. Instead, we use aggressive prefetching and treat flash crowd events separately. We believe ISPs can benefit from both CPM and our approach.

## VIII. Conclusion

In this paper, we have explored the idea of using storage in the home and local connectivity between homes to improve the hard-to-upgrade network infrastructure's ability to deliver on-demand multimedia content. Our proposed system uses a sophisticated content placement algorithm that aims to satisfy demand for content using local resources alone; thus, limiting the utilization of the ISP's network infrastructure. The system adapts to changes in popularity of content, accommodates abnormal behavior such as flash crowd events, and is robust to link and node failures. Our experimental evaluation and simulation results suggest that the resulting savings could be quite significant, reaching up to 45% in terms of bandwidth savings at the access. As local connectivity continues to grow in speed, and storage prices continue to drop, the techniques presented in this paper could form the basis for bridging the gap between major network infrastructure upgrades.

## References


[1] K. K. Ramakrishnan, "CPM - adaptive VoD with cooperative peer assist and multicast," Keynote Speech at IEEE LANMAN, Cluj-Npoca, Romania, Sep. 2008.

[2] "Cisco visual networking index: Forecast and methodology, 20082013," http://www.cisco.com/, 2009.

[3] H. Yoon, "KT FTTH network evolution," Talk at the Optical Fiber Communication Conference, San Diego, CA, Mar. 2009.

[4] V. Gopalakrishnan *et al.*, "CPM: Adaptive video-on-demand with cooperative peer assists and multicast," in *Proc. IEEE INFOCOM*, 2009.

[5] J. Poniewozik, "Here's to the death of broadcast," http://www.time.com/time/magazine/article/0,9171,1887840-2,00.html, 2009.

[6] S. Borst *et al.*, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, San Deigo, CA, 2010.

[7] A. Chankhunthod *et al.*, "A Hierarchical Internet Object Cache," in *Proc. USENIX ATC*, Jan. 1996.

[8] K. Suh *et al.*, "Push-to-peer video-on-demand system: Design and evaluation," *IEEE JSAC*, vol. 25, no. 9, Dec. 2007.

[9] V. Valancius *et al.*, "Greening the Internet with nano data centers," in *Proc. ACM CoNext*, Dec. 2009.

[10] "Industry briefs: Q308 streaming trends and drivers," www.nielsenpreview.com, 2009.

[11] "Jacobs media technology survey: Latest media usage findings," http://www.radiostreamingnews.com/2010/04/jacobs-media-technology-survey-latest.html, 2010.

[12] "Here's what fake HD video looks like," http://www.zdnet.com/blog/ou/heres-what-fake-hd-video-looks-like/962, 2008.

[13] K. Kerpez *et al.*, "IPTV service assurance," *IEEE Commun. Mag.*, vol. 44, no. 9, Sep. 2006.

[14] C. Warren, "Will google TV be a game-changer in the realm of connected devices?" http://mashable.com/2010/05/20/google-tv-future/, 2010.

[15] "National broadband plan - the broadband availability gap," http://www.broadband.gov/plan/broadband-working-reports-technical-papers.html, 2010.

[16] "Industry data - ncta.com," http://www.ncta.com/Statistics.aspx, 2010.

[17] D. Han *et al.*, "Mark-and-Sweep: Getting the "inside" scoop on neighborhood networks," in *Proc. ACM IMC*, Oct. 2008.

[18] "Multimedia over coax alliance," http://mocalliance.org.

[19] M. Learmonth, "Hulu's Desktop App: A Wake-Up Call For Cable Operators," http://www.businessinsider.com/hulus-desktop-app-a-wake-up-call-for-cable-operators-2009-6, Jun. 2009.

[20] A. Akella *et al.*, "Self-management in chaotic wireless deployments," in *Proc. ACM Mobicom*, Cologne, Germany, Sep. 2005.

[21] I. F. Akyildiz *et al.*, "Wireless mesh networks: a survey," *Comput. Netw. ISDN Syst.*, vol. 47, no. 4, 2005.

[22] R. Shaw, "And we'll have FON, FON FON: Skype, Google to help fund global WiFi network," blogs.zdnet.com/ip-telephony/?p=890, 2006.

[23] D. Han *et al.*, "An access network architecture for neighborhood-scale multimedia delivery," Carnegie Mellon University, Tech. Rep. CMU-CS-10-128, Jun. 2010.

[24] D. Naudts *et al.*, "A wireless mesh monitoring and planning tool for emergency services," in *Proc. E2EMON*, May 2007.

[25] "Alcatel 7330 isam fttn product brochure," http://broadbandsoho.com/FTTx/Alcatel_7330_ISAM.pdf, Last Accssed, Jul. 2010.

[26] H. Yu *et al.*, "Understanding user behavior in large-scale video-on-demand systems," in *Proc. EuroSys*, 2006.

[27] S. Acharya and B. Smith, "Characterizing user access to videos on the world wide web," in *Proc. MMCN*, 2000.

[28] T. Qiu *et al.*, "Modeling channel popularity dynamics in a large IPTV system," in *Proc. ACM SIGMETRICS*, Seattle, WA, Jun. 2009.

[29] M. Cha *et al.*, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, 2009.

[30] S. Viswanathan and T. Imielinski, "Pyramid broadcasting for video on demand service," in *Proc. IEEE MMCN*, 1995.

[31] "A permutation-based pyramid broadcasting scheme for video-on-demand systems," in *Proc. IEEE ICMCS*, 1996.

[32] K. A. Hua and S. Sheu, "Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems," in *Proc. the ACM SIGCOMM*, 1997.

[33] S. Sen *et al.*, "Proxy prefix caching for multimedia streams," in *Proc. IEEE INFOCOM*, 1999.

[34] D. Eager *et al.*, "Bandwidth skimming: A technique for cost-effective video-on-demand," in *Proc. IEEE MMCN*, 2000.

[35] Y. Guo *et al.*, "Prefix caching assisted periodic broadcast: Framework and techniques to support streaming for popular videos," in *Proc. IEEE ICC*, 2001.

[36] K. A. Hua *et al.*, "Patching: a multicast technique for true video-on-demand services," in *Proc. ACM MULTIMEDIA*, 1998.

[37] S. Sheu *et al.*, "Chaining: A generalized batching technique for video-on-demand systems," in *Proc. IEEE ICMCS*, 1997.

[38] S. Banerjee *et al.*, "Scalable application layer multicast," in *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002.

[39] M. Castro *et al.*, "SplitStream: High-bandwidth content distribution in cooperative environments," in *Proc. SOSP*, Oct. 2003.

[40] D. Kostic *et al.*, "Bullet: High bandwidth data dissemination using an overlay mesh," in *Proc. SOSP*, Oct. 2003.

[41] C. Huang *et al.*, "Can Internet video-on-demand be profitable?" in *Proc. ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.

[42] S. Liu *et al.*, "Performance bounds for peer-assisted live streaming," in *Proc. ACM SIGMETRICS*, Annapolis, MD, Jun. 2008.

[43] V. Janardhan and H. Schulzrinne, "Peer assisted VoD for set-top box based IP network," in *Proc. ACM P2P-TV Workshop*, 2007.

[44] Y. Huang *et al.*, "Challenges, design and analysis of a large-scale P2P-VoD system," in *Proc. ACM SIGCOMM*, 2008.

[45] C. Huang *et al.*, "Understanding hybrid CDN-P2P: why limelight needs its own Red Swoosh," in *Proc. ACM NOSSDAV*, 2008.

[46] H. Yin *et al.*, "Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky," in *Proc. ACM Multimedia*, 2009.

[47] S. Bessler, "Optimized content distribution in a push-VoD scenario," in *Proc. Next Generation Internet Networks*, Apr. 2008.

[48] M. Cha *et al.*, "On next-generation telco-managed P2P TV architectures," in *Proc. IPTPS*, 2007.