

# Video-on-demand Equipment Allocation

Frederic Thouin and Mark Coates  
McGill University  
Dept Elec. and Comp. Eng.  
3480 University, Montreal, Quebec, Canada H3A 2A7  
{fthou,coates}@ece.mcgill.ca

Dominic Goodwill  
Nortel Networks  
3500 Carling Avenue  
Ottawa, ON, Canada K2H 8E9  
goodwill@nortel.com

## Abstract

*Video-on-demand (VoD) service providers are intensely interested in transport, storage, streaming and caching in content delivery networks. Today's 5,000-hour library may grow toward the 750,000-hour "Long Tail" movie and TV-series catalog. We propose a method to calculate how much of a library should be cached. Much previous work focused on theoretical caching concepts, or the dynamics of cache filling and reclamation. Our method explicitly considers the impact of the available video server equipment; we present a VoD design tool comprising a novel cost function, hit ratio estimation and heuristic.*

## 1. Introduction

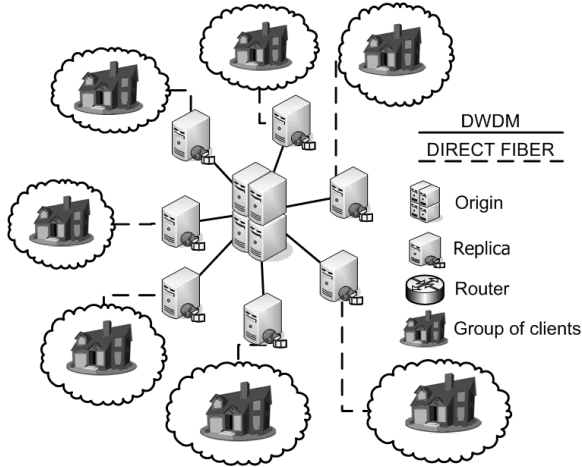
Network-based video-on-demand (VoD) deployments are today very limited in scope. The largest deployed libraries are just 0.7% (5,000 hours) of the global movie and TV-series catalog and peak utilization of VoD targets are 10-15% of broadcast TV peak viewing numbers. Recognizing that libraries and usage may grow, service providers are intensely interested in the scalability of content delivery networks that provide content propagation, storage, streaming, and transport. Three dominant factors must be considered when planning scalable deployments for VoD: library size, peak concurrent streams and rate of ingest of new content. Each could increase by one or more orders of magnitude after the deployment. In this paper we describe a method and design tool for the planning of VoD systems; our work strives to address the future 750,000-hour global movie and TV-series catalog, to better exploit the heavy tail of the popularity distribution, called the "Long Tail" of content [6].

Content delivery networks (CDNs) are designed to distribute content to a set of clients, as streams or as files [4, 5, 11, 14, 15]. Through approaches such as replication of content at multiple servers (replicas, proxies or caches), CDNs attempt to minimize latency at the end-user while reducing bandwidth consumption and load at the origin server. This

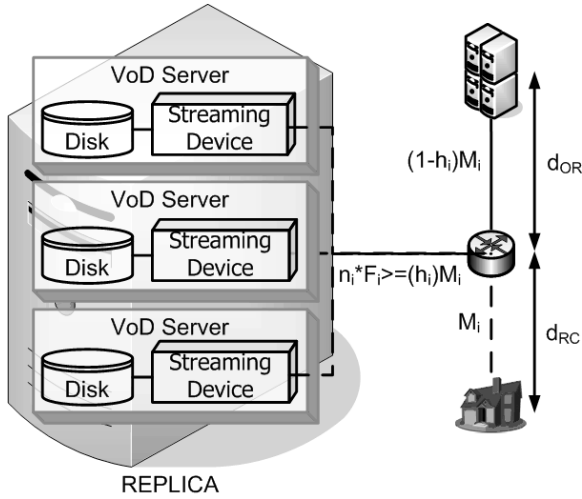
paper deals primarily with the case of a streaming CDN, in that the client is assumed to have buffering capability but not caching capability. Nonetheless, the mathematics and the model are easily extended to the client-cached scenario. The CDN delivery of streaming media causes new problems that did not apply to the distribution of HTTP objects: streaming objects are much larger than web objects and hence create much more traffic [10]. Furthermore, it is no longer possible to assume infinite storage size at the replica locations [15].

The design of a CDN consists of making resource planning decisions and the development of in-service intelligent request routing, resource control policies, and performance monitoring. In this paper, we focus on the first challenge: the allocation of resources during network planning, generally performed when planning greenfield and incremental deployments. Of particular interest is VoD delivery across metropolitan area networks (MANs). A well-known formulation of this task is the replica (or cache) placement problem. This consists of determining a subset of  $N$  potential locations in a given topology (Fig. 1(a)) to place  $k$  replicas, such that the total network cost is minimized for a given quality of experience (QoE) impairment, such as delay, packet loss, frame loss, or packet jitter. Bandwidth requirements and transmission distances have generally been considered as the important resource parameters. Storage capacity also plays a role. The location of the replicas is important, but very different designs and costs are achieved depending on the capabilities of the available network equipment. We define the *VoD equipment allocation problem* as choosing the number of streaming and storage devices for each of the replica locations, such that the deployment cost of the VoD system is minimized. Referring to Fig. 1(b), the method is implemented by determining the number of VoD servers present at each replica site. As part of this analysis, we also determine the fraction of the total library that should optimally be stored at each location.

The remainder of this paper is organized as follows. Section 2 introduces related previous work. In Section 3, we ex-



(a) Requests from a group of clients are routed to the associated replica or to the origin in case of a cache miss.



(b) A VoD server has storage capacity of  $G_i$  TB and streaming capacity of  $F_i$  Gbps. The total streaming capacity  $n_i F_i$  must be greater or equal to  $h_i M_i$  where  $h_i$  is the hit ratio and  $M_i$  is the worst-case demand at site  $i$ .

**Figure 1. (a) Logical connectivity between clients, replicas and origin. (b) Replica with  $n_i = 3$  VoD servers.**

press the equipment allocation problem as an optimization problem, and we state our assumptions. In Section 4, we present our solution to this problem, developing a novel cost function, hit ratio estimation function and heuristic. In Section 5, we apply our heuristic to three scenarios with different demand, equipment capabilities, and topologies (or geographies). We also compare the optimal cost, as generated by our heuristic, to a scenario where no replicas are installed at the available locations in the topology. Thus, we illustrate our method of determining when a centralized VoD deployment should be evolved to a hierarchical-distributed VoD deployment. In section 6, we summarize our observations.

## 2. Related work and Contribution

Many techniques have been proposed to place replicas or allocate storage in CDNs in a near-optimal manner [3–5, 12, 15]. They differ primarily according to the nature of the cost function, which specifies the optimization parameters (distance, demand, storage, etc.). The simpler approaches consider only the aggregate user demand; the specific objects requested are not important [3, 4, 15]. Another avenue is to treat objects individually, associating with each a different popularity (user demand) [5, 12]. Although most of these functions account for transport costs and some for storage costs, none of them explicitly includes the number and cost of the equipment installed at each location. We extend the cost function proposed in [15], which is based on topology, demand and hit ratio, by also considering the cost incurred through installation of the VoD servers. These cost functions are often complex and minimizing them to determine the optimal design is NP-hard. A popular heuristic that offers near-optimal performance, considered by many authors, is greedy selection [5, 7, 15]. It consists of iteratively adding replicas to the design to achieve the lowest cost at every step.

It is not always possible to have complete replicas of the origin server because the large size of multimedia objects leads to a high storage cost. An alternative is to store only specific objects from the origin at the surrogate servers [12]. The empirical study performed in [7] indicates that approximately 80% of requests for Web objects are for 10% of the objects. This suggests that a relatively small amount of storage can achieve a high hit ratio if the cached content is carefully chosen. Previous studies in the distribution of multimedia files in CDNs or in VoD applications have used Zipf’s Law to characterize the popularity of the different files [2, 12, 13]. However, empirical data indicates that a Zipf model is not a good fit for the most popular files [1, 9].

**Contributions:** Most previous work on content distribution considered replica placement or content allocation. In contrast, we address the *VoD equipment allocation problem*, which focuses on identifying the optimal number of VoD servers at a set of locations. We present four main contributions to solving this problem. We design a parametric function for estimating the worst-case hit ratio for given system parameters (cache size, library size and file arrival rate) using discrete-time simulations based on an extension of the file access model proposed in [9]. We propose a cost function based on the hit ratio, the distributed demand and the number of VoD servers at each location. We develop a two-step heuristic to generate a solution to the problem, which first relaxes the integer constraint and then searches for an integer solution in the neighborhood of this initial solution. Finally, we develop an *interactive* design tool that implements our cost function, hit ratio function and heuristic.

### 3. Problem statement

We address the problem of determining the number of storage and streaming devices needed at each potential replica location. We require a topology of a metro-area network (MAN) indicating the set of inter-nodal distances and the specifications (cost and capacity) of network elements and available equipment. We consider the case where only one type of equipment (VoD server) is installed at each site but allow this equipment type to vary from site to site. We define the *VoD equipment allocation problem* as choosing the equipment for each of these replicas such that the deployment cost of the network is minimized.

As illustrated in Fig. 1(a), this topology contains one origin server and a maximum of  $N$  replicas. Each replica is responsible for a group of clients representing a fraction of the population; any request made by a client in that group is routed to that replica. The origin server hosting the entire library (the complete set of objects) can be located anywhere and serves all the requests that replicas are unable to fill.

This paper does not consider the management of the content at the replicas. We suppose that there exists an external mechanism to maintain the most popular files at the replicas, which can be executed during off-peak hours when more bandwidth is available. Because content delivery itself is also out of the scope of this paper, we are assuming unicast delivery to the user-end.

#### 3.1. Mathematical formulation

Let  $\mathcal{S} = \{s_i : i = 1, \dots, N\}$  and  $\mathcal{T} = \{t_i : i = 1, \dots, N\}$  where  $s_i$  is the number of streaming devices with capacity  $F_i$  (Gbps) and  $t_i$  the number of storage devices with capacity  $G_i$  (TB) of replica site  $i$ . Let  $C_{\text{TOT}}(\mathcal{S}, \mathcal{T})$  be a positive function that maps the number of devices installed at each location to the total network cost. The objective is to determine  $\mathcal{S}$  and  $\mathcal{T}$  to minimize total system cost:

$$\{\mathcal{S}^*, \mathcal{T}^*\} = \arg \min_{\mathcal{S}, \mathcal{T}} C_{\text{TOT}}(\mathcal{S}, \mathcal{T}) \quad (1)$$

This formulation is only valid when the streaming and storage devices can be deployed independently ( $s_i$  does not need to be equal to  $t_i$ ). However, in practice, the two devices are often deployed as a joint unit called a VoD server, so that  $s_i = t_i$  (Fig. 1(b)). In that case, we let  $\mathcal{N} = \{n_i : i = 1, \dots, N\}$  and define  $n_i$  as the number of VoD servers with streaming capacity  $F_i$  and storage capacity  $G_i$  at location  $i$ . The objective in this second formulation (used in the rest of this paper) is to choose the number of VoD servers at each replica that minimizes the total cost:

$$\mathcal{N}_{\text{opt}} = \arg \min_{\mathcal{N}} C_{\text{TOT}}(\mathcal{N}) \quad (2)$$

We denote the worst-case demand  $M_i$  at replica  $i$  as the total bandwidth required to serve all client requests using

unicast streaming during the peak utilization hours. We assume that we either know  $M_i$  or can approximate it from a given population size and peak usage ratio. We define the hit ratio  $h_i$  as the smallest fraction of requests satisfied by replica  $i$  at any given time (worst-case). If the desired object is not present at the replica or the replica does not have enough streaming capacity, the request is unsatisfied (cache miss) and routed to the origin server. Although the hit ratio could be used as a measure of service quality, we decided not to add this constraint to the optimization problem because of the imperceptible difference in quality of streaming video between the origin and a replica in a MAN.

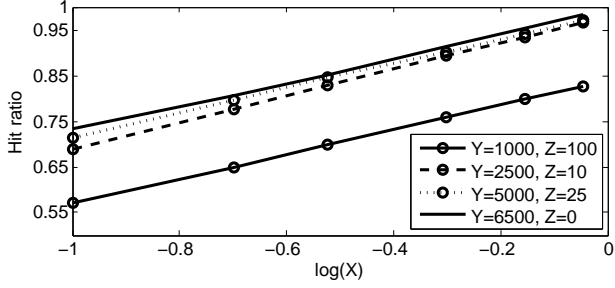
### 4. Proposed solution

In this section, we present the three components of our solution: the hit ratio function, the cost function and the heuristic. The first step of the heuristic produces an initial solution  $\mathcal{X}$  that represents the fraction of the library that should be stored at every replica  $i$  to minimize our cost function. This value is then mapped to an estimate of the hit ratio, which allows us to calculate the number of servers needed,  $\mathcal{N}_{\text{ini}}$ . The second step of the heuristic consists of searching the neighborhood of  $\mathcal{N}_{\text{ini}}$  to determine a near-optimal integer solution  $\mathcal{N}_{\text{opt}}$ . By feeding these values into our cost function, we generate the infrastructure and transport cost for the entire network.

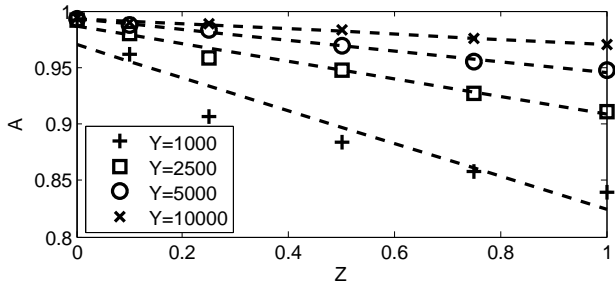
#### 4.1. Hit ratio function

The purpose of a file popularity model is to predict the access frequency of a given file, which can be estimated by dividing the number of requests for this file by the total number of requests. The authors of [9] propose a file access model that is driven by Zipf's Law, but takes into account the "fetch-at-most-once" and "new arrivals" factors. VoD system users rarely access the same file twice because the files are not modified (fetch-at-most-once). Also, new files are periodically added to the system and since the popularity of a titles diminishes in time, these objects become the most popular titles. Although it is possible to estimate the worst-case hit ratio through simulations, for the purpose of an interactive design process where we need to modify parameters repetitively, it is impractical and time-consuming. Our objective is to train a parametric function that provides an estimate of the worst-case hit ratio based on specified system parameters in a few seconds compared to the tens of minutes simulations would take.

We designed a simulation environment with a library of size  $Y$  and a cache of size  $X \cdot Y$  where files are accessed according to the model described by Gummadi [9]. We calculate the hit ratio by dividing the number of requests for objects in the cache by the total number of requests. Let each client's library  $L_j$  be a subset of the complete library



(a)  $H$  as a function of  $\log(X)$  for different pairs  $Y, Z$ . The linear curves indicate that  $\hat{H} = A + B \log(X)$  achieves an adequate fit.



(b) Markers show values of  $A$  as a function of  $Z$  for different values of  $Y$ . Dashed lines (–) show the linear fits.

**Figure 2. Plots showing data fitting curves to construct the form of  $\hat{H}$ ,  $A$  and  $B$ .**

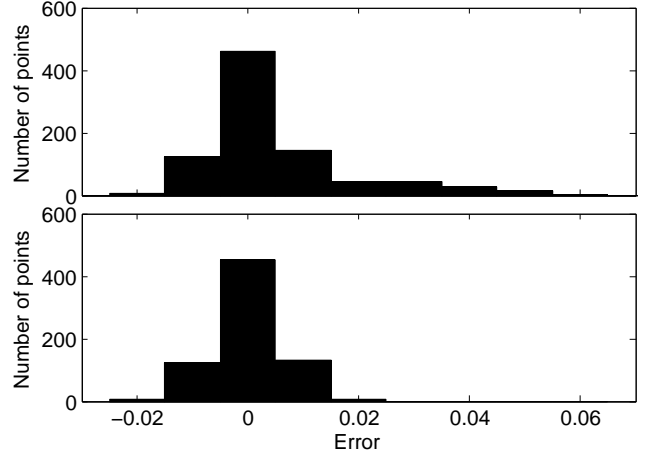
$L$  that excludes all files client  $j$  has selected in the previous weeks. During each iteration (one week) of the discrete-time simulation, the following sequence of events occurs:

1. Clients are added to the population at a specified rate.
2. New files are added to the library  $L$  at a specified rate.
3. The cache is filled with the most popular files.
4. Each client  $j$  selects an object from his library  $L_j$ .
5. The weekly hit ratio is calculated.

The users' requests are generated using a Zipf distribution with coefficient  $\alpha = 1$ . The probability of selecting the file at rank  $i$  in library  $L_j$  is given by  $p_j(i)$ :

$$p_j(i) = \frac{i^{-\alpha}}{\sum_{i \in L_j} i^{-\alpha}}$$

Files that have already been fetched by the user cannot be selected again (fetch-at-most-once-model). After every request a user makes, the selected file is removed from his library  $L_j$  and file selection probabilities are recalculated. New files are introduced in the library  $L$  and each library  $L_j$  at a specified rate. The insert position of a file is determined using a Zipf distribution (with  $\alpha = 1$ ); the ranks of existing files which are less popular are decreased and selection probabilities are recalculated. We ran extensive simulations with different values for the following parameters:



**Figure 3. Histograms of the error  $\hat{H} - H$  between our function estimate  $\hat{H}$  and the value observed during simulations  $H$ . Top: entire dataset ( $1000 \leq Y \leq 10000$ ). Bottom: reduced dataset  $2500 \leq Y \leq 10000$ .**

1. Number of weeks (length of the simulation).
2. Size of the client population.
3. Number of new clients every week.
4. Size of the initial library of objects  $Y$ .
5. Number of files added to the library every week  $Z$ .
6. Size of the cache as a fraction ( $X$ ) of the library size.

From our simulation results, we determined that the only parameters that have a significant impact on the hit ratio were the library size  $Y$ , the number of files added every week  $Z$  and the cache size ratio  $X$ . We generated 864 points for the hit ratio  $H$  by running the simulation four times for 216 possible combinations of  $X, Y$  and  $Z$ . In Fig. 2(a), we observe the linear behavior of  $H$  as a function of  $\log(X)$  for different values of  $Y$  and  $Z$  and propose the form in (3) for our estimate  $\hat{H}$ , where  $0 \leq X \leq 1$ ,  $1000 \leq Y \leq 10000$  and  $0 \leq Z \leq 100$ . We construct the bilinear functional form for  $A$  and  $B$  represented respectively by (4) and (5) based on the linear relationship between  $A$  and  $Z$  observed in Fig. 2(b) and similar observations made for  $B$ . Performing further plot analysis, we observed that using  $\log(Y)$  in (4) and  $Y$  in (5) generated the best fit for  $A$  and  $B$ .

$$\hat{H} = A + B \cdot \log(X) = f_3(X) \quad (3)$$

$$A = K_1 + K_2 Z + K_3 \log(Y) + K_4 Z \log(Y) \quad (4)$$

$$B = K_5 + K_6 Z + K_7 Y + K_8 ZY \quad (5)$$

We determine the values of the coefficients  $K_1$  to  $K_8$  by solving in the least squares sense the system  $KV(X, Y, Z) \cong H$  obtained by substituting (4) and (5) into (3). Our resulting function for  $h_i$  is quite accurate, showing

less than 2% error eighty-five percent of the time and less than 5% error ninety-nine percent of time. In the top plot of Fig. 3, we show the histogram of the error distribution for the entire dataset used ( $1000 \leq Y \leq 10000$ ) for our simulations where the error is less than 5% ninety-nine percent of the time. In the bottom plot we show the histogram of a reduced dataset where the values for library size below 2500 are discarded. The accuracy of this reduced dataset is much higher (less than 1.5% error ninety-eight percent of the time) because  $Y = 1000$  did not fit the proposed forms as well as the other data points (see Fig. 2(b)).

## 4.2. Cost function

We can express the total cost,  $C_{TOT}$ , as the sum of the cost of infrastructure,  $C_T$ , and the cost of transport,  $C_S$ .

$$C_{TOT} = C_T + C_S \quad (6)$$

The cost of infrastructure,  $C_T$ , includes the software and start-up cost of a location ( $A_i$ ) and the cost of VoD servers ( $B_i$ ) for every replica site  $i$  and the origin server. In (7), we express  $C_T$  as a function of the number of VoD servers installed at location  $i$ ,  $n_i$ , and the origin,  $n_o$ .

$$C_T = \sum_{i=0}^N A_i + B_i n_i = f_1(n_o) + \sum_{i=1}^N f_1(n_i) \quad (7)$$

The cost of transport consists of two components: transport from the origin to replicas and clients,  $C_{S_{OR_i}}$ , and transport from replica  $i$  to client  $i$ ,  $C_{S_{RC_i}}$ . It includes the cost of node interfaces ( $C_{IF}$ ) and of fiber ( $C_f$ ). The transport from replicas to the user-end (small distances) uses direct fiber whereas the transport from the origin to the replicas uses DWDM connections.

$$C_S = \sum_{i=1}^N C_{S_{OR_i}} + C_{S_{RC_i}} \quad (8)$$

$$C_{S_{RC_i}} = n_{RC_i} \cdot (2 \cdot C_{IF} + d_{RC_i} \cdot C_f) \quad (9)$$

$$C_{S_{OR_i}} = n_{OR_i} (2 \cdot C_{IF}) + \frac{n_{OR_i}}{dw_{max}} \left[ 2C_{DWDM} + d_{OR_i} \cdot C_f + \left( \frac{d_{OR_i}}{max_{amp}} \right) \cdot C_{LA} \right] \quad (10)$$

$n_{OR_i}$ : Num. of interfaces (fibers) toward the origin.

$n_{RC_i}$ : Num. of interfaces (fibers) toward the user-end.

$c$ : Fiber capacity. (Gbps)

$C_{IF}$ : Node switch interface cost. (\$)

$C_f$ : Cost of fiber. (\$/km)

$C_{DWDM}$ : Cost of DWDM equipment (\$)

$w_{max}$ : Number of fibers supported by DWDM equipment.

$C_{LA}$ : Cost of line amplifier. (\$)

$d_{amp}$ : Max. distance between two amplifiers. (km)

The number of fibers at each node depends on the amount of traffic on the various links, the hit ratio at the replica and the fiber capacity. On the link between location  $i$  and the clients ( $RC_i$ ), the traffic is equal to the demand from the user,  $M_i$ . On the link between the origin and a location  $i$  ( $OR_i$ ), all the requests that cannot be served by the replica (cache misses) are handled by the origin server and generate a traffic equal to  $(1 - h_i) \cdot M_i$ . Notice that we are using non-integer values for the network components because we assume that the unused fraction can be used for other applications and does not need to be included in the cost.

$$n_{OR_i} = \frac{(1 - h_i) \cdot M_i}{c} \quad n_{RC_i} = \frac{M_i}{c} \quad (11)$$

Because the worst-case demand between location  $i$  and the group of clients is fixed,  $C_{S_{RC_i}}$  does not depend on any of the optimization variables. However,  $C_{S_{OR_i}}$  indirectly depends on  $n_i$  because the hit ratio  $h_i$  changes with the number of VoD servers installed. We express the cost of transport  $C_S$  as follows, where  $f_2(h_i, M_i)$  is obtained by substituting (11) into (10) and (9):

$$\begin{aligned} C_S &= \sum_{i=1}^N C_{S_{OR_i}} + C_{S_{RC_i}} = \sum_{i=1}^N f(n_{OR_i}) + f(n_{RC_i}) \\ &= \sum_{i=1}^N f_2(h_i, M_i) \end{aligned} \quad (12)$$

By substituting (7) and (12) in (6), we can express the total cost as a function of the number of VoD servers installed ( $n_i$ ), the hit ratio ( $h_i$ ) and the demand ( $M_i$ ) at each location:

$$C_{TOT} = f_1(n_o) + \sum_{i=1}^N f_1(n_i) + f_2(h_i, M_i) \quad (13)$$

The required number of VoD servers is determined by either the streaming or storage requirement ( $n_i = \max(s_i, t_i)$ ), expressed as functions of  $X_i$ :

$$s_i = \frac{h_i \cdot M_i}{F_i} = \frac{f_3(X_i) \cdot M_i}{F_i} \quad t_i = \frac{X_i \cdot Y}{G_i} \quad (14)$$

$$s_o = \frac{\sum_i (1 - f_3(X_i)) \cdot M_i}{F_o} \quad t_o = \frac{Y}{G_o} \quad (15)$$

Define  $f_4(X_i) \triangleq \max s_o, t_o$  and  $f_5(X_i) \triangleq \max s_i, t_i$ . By substituting (3), (14) and (15) into (13) and assuming that the demand  $M_i$  is known, we express  $C_{TOT}$  as a function of the optimizing variable  $X_i$ :

$$C_{TOT} = f_1(f_4(X_o)) + \sum_{i=1}^N f_1(f_5(X_i)) + f_2(f_3(X_i)) \quad (16)$$

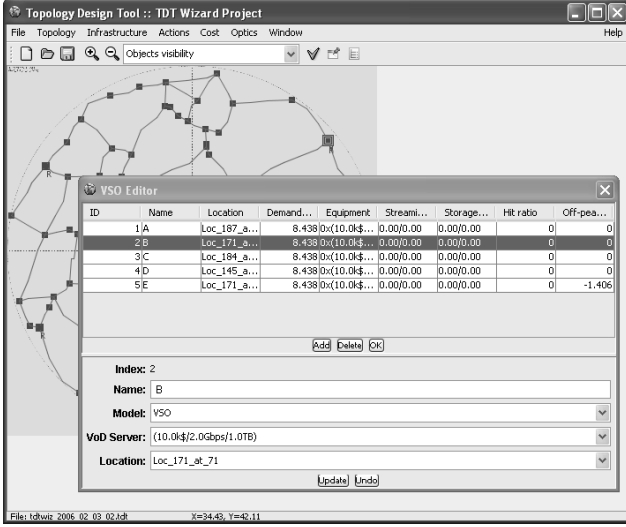


Figure 4. Screenshot of our design tool.

### 4.3. The heuristic

**Step 1:** The first step of the heuristic is to provide an initial solution,  $\mathcal{X} = \{X_i : i = 1, \dots, N\}$ , representing the optimal fraction of the library to store at each replica. We obtain  $\mathcal{X}$  by performing a constrained nonlinear optimization on  $C_{TOT}$  (as expressed in (16)) where  $\mathbf{0} \leq \mathcal{X} \leq \mathbf{1}$ , which is solved using a sequential quadratic programming (SQP) method [8]. From  $\mathcal{X}$ , we calculate  $\mathcal{N}_{ini}$ , the set of fractional number of VoD servers, by expressing  $s_i$ ,  $t_i$ ,  $s_o$  and  $t_o$  as functions of  $X_i$ . From this initial solution  $\mathcal{N}_{ini}$  we can calculate the lower-bound of  $C_{TOT}$  and upper-bound by rounding  $\mathcal{N}_{ini}$  up to the nearest integer.

**Step 2:** The second step of the heuristic consists of searching the surroundings of  $\mathcal{N}_{ini}$  for a near-optimal integer solution. For each replica, we calculate the cost for integer values of  $n_i$  near the initial value calculated ( $n_i \pm 2$ ) looking for a lower cost. We repeat this procedure until no changes in  $\mathcal{N}_{opt}$  can decrease the total cost. For each value of  $n_i$  we maximize the hit ratio to decrease the transport cost from the replica to the origin:

$$X_{storage} = \frac{n_i G}{Y} \quad X_{streaming} = f_3^{-1}(h_i) = f_3^{-1}\left(\frac{n_i F}{M_i}\right)$$

$$X_i = \min(X_{storage}, X_{streaming})$$

## 5. Results

In this section we examine the results obtained from applying our heuristic to three scenarios using our design tool. The tool, shown in Fig. 4, allows the quick creation of topologies, easy modification of system parameters and

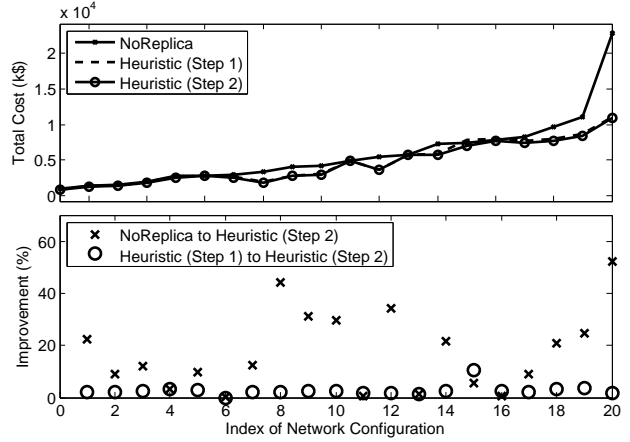


Figure 5. Scenario 1. Impact of each step of the heuristic on total network cost (top) and cost improvement (bottom). In both plots, 20 different cases are displayed in increasing order of cost of NoReplica.

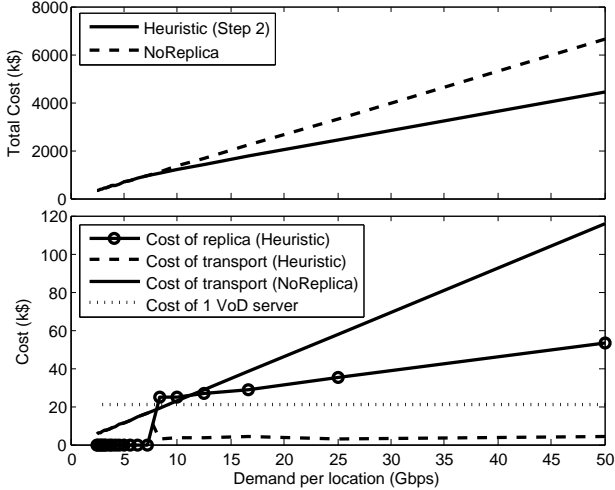
visualization of the resulting design. In the figure, we can see the topology of a MAN (small squares represent interconnected locations) and the object editing window which allows the user to create and edit objects in the topology; for example, the user can edit a replica object by changing the type of VoD servers available.

**Scenario1** 20 different sets of inputs where the number of locations  $N$  in the topology is between 1 and 100, files in the library between 1000 and 10000 and file arrival rate per week between 0 and 100. System parameters are all uniformly distributed within specified ranges: demand  $M$  (1-20Gbps), startup cost  $A$  (6-37k\$), VoD server cost  $B$  (1-53k\$), streaming capacity  $F$  (1-5Gbps), storage capacity  $G$  (1-10TB), distance to the origin  $d_{OR}$  (0-50km), average distance to the client  $d_{RC}$  (0-5km), cost of bandwidth (0-4k\$/Gbps) and cost of storage (0-3k\$/TB);

**Scenario2** Topology of 25 locations with the system parameters uniformly distributed within the same ranges as in Scenario1. For each trial, demand  $M_i = M$  at each replica, where  $M$  varies from 2.5Gbps to 50Gbps;

**Scenario3** Topology of 14 locations with the system parameters uniformly distributed within the same ranges as in Scenario1. The specifications of the equipment and demand at each node appear in Table 1.

The top panel of Fig. 5 shows three different total network costs  $C_{TOT}$  for Scenario1: cost without any equipment installed at the replicas ( $n_i = 0$  for all  $i$ ) and cost after the first and second step of the heuristic. The bottom panel of Fig. 5 shows the percentage reduction of the cost achieved by the two-step heuristic. Our two-step heuristic



**Figure 6. Scenario 2. Total network cost (top) and individual location cost (bottom) for  $M_i = M$  at each replica, where  $M$  varies from 2.5Gbps to 50Gbps.**

yields average improvements of 17% over the case where no equipment is deployed. The majority of the heuristic improvement comes from the first step.

In Scenario2 we illustrate the impact of the demand  $M$  on the deployment cost. Fig. 6 illustrates that as the demand increases, the cost differential between the design generated by our tool and a design in which no equipment is installed grows substantially. Below a certain demand ( $\approx 7 - 8Gbps$ ), both designs are of equal cost, which means that if the demand is too low, it is no longer cost-efficient to deploy equipment. The bottom panel of Fig. 6 compares costs of transport and infrastructure of a single location  $i$ , which has a startup cost  $A = 19k\$$  and where VoD servers with  $3Gbps$  and  $2TB$  capacity are available at  $2k\$$ . As long as “Cost of 1 VoD server” is lower than “Cost of transport (NoReplica)”, it is beneficial to cache content at  $i$ . If the equipment installed at the origin and  $i$  is identical and “Cost of transport (NoReplica)” is smaller than “Cost of 1 VoD server”, then it is cheaper not to install any replica and carry the entire demand up to the origin. In the analyzed scenario, equipment is cheaper at  $i$  than at the origin, so there is one point ( $M = 8.3Gbps$ ) where the heuristic indicates that a replica should be installed even though the cost of transport is less than the minimum deployment cost. Therefore, the demand and type of equipment not only have an impact on the fraction of the library to cache, but also determine whether or not caching content is even profitable.

Table 1 displays the values for  $n_i$ ,  $s_i$  and  $t_i$  calculated with our tool for Scenario3. The total network cost for this equipment is 1,810k\$. Looking at the table, we notice significant discrepancies between the values of  $s_{i_1}$  and  $t_{i_1}$ , which signifies that resources are wasted because of

poorly chosen equipment. For example, at location 3, the required number of streaming devices is almost twice the number of storage devices, whereas the streaming capacity of the equipment is half of the storage capacity. We consider the effect of making equipment available at these locations with specifications that better match storage and streaming needs. For example, we change the value of  $F_3$  from 1 to 3 in order to have a closer match for  $s_3$  and  $t_3$ . We repeat for the other three locations where  $n_i \neq 0$  (7, 10 and 11) and adjust the value of  $B$  accordingly; for example,  $B_3$  increases from 9k\$ to 15k\$ to support an extra 2Gbps. All the modifications ( $B_{i_2}$ ,  $F_{i_2}$  and  $G_{i_2}$ ) and new results ( $n_{i_2}$ ,  $s_{i_2}$ ,  $t_{i_2}$ ) are also shown in Table 1. We notice that the four locations where we modified the hardware now have  $s_{i_2} = t_{i_2}$ , which indicate a better usage of resources. Moreover, because of the savings at these locations, it is now beneficial to install more equipment at locations 1 and 14 to achieve a minimal network cost. We also note that even though the value of  $n_5$  has not changed, the streaming and storage requirements have increased from 0 to 1.7. The new total network cost for this setup is 1,580k\$, a 12.5% improvement. It is important to stress that the prices and capacity used in these scenarios are not intended to reflect the real values used in practice. However, this simple example shows the impact of modifying the type of equipment installed at each location on the total deployment cost.

## 6. Conclusion

Network cost is affected not only by where replicas are located, but also what equipment comprises a replica. We developed a design tool to address the *VoD equipment allocation problem*. There are three principal contributions from the development of this tool. We used extensive simulations to train a parametric function that generates accurate estimates of the hit ratio for given cache size, library size and file arrival rate. We constructed a cost function based on the hit ratio  $h_i$ , the demand  $M_i$ , and the number of VoD servers  $n_i$  at each location. We designed a two-step heuristic that relaxes the integer constraint to produce an initial solution and then identifies a near-optimal integer solution in the reduced search space. The tool that implements our cost function, hit ratio function and two-step heuristic is truly interactive because it allows designers to create and change network models to generate optimal designs in an efficient and timely manner.

Our key conclusions are: (i) the nature of the available server equipment has a major impact on the design and cost of a VoD network; and (ii) it is not always beneficial to cache content. It is profitable to install VoD servers (regardless of the library size) if the demand at the given location is significant. On the other hand, even if the library has tens of thousands of assets, if the demand is too low, no amount of caching can reduce the network cost. Accounting for the

**Table 1. Scenario 3. Initial specifications of 14 locations (left). On the right, specifications of the locations after modifying equipment (modified values are highlighted with the surrounding box).**

$M_i$	$A_i$	$B_{i_1}$	$F_{i_1}$	$G_{i_1}$	$n_{i_1}$	$s_{i_1}$	$t_{i_1}$	$B_{i_2}$	$F_{i_2}$	$G_{i_2}$	$n_{i_2}$	$s_{i_2}$	$t_{i_2}$
10	19	12	2	2	0	0	0	12	2	2	3	4.3	4.3
13	8	29	1	9	0	0	0	29	1	9	0	0	0
18	12	9	1	2	18	17.8	11	15	3	2	6	5.5	5.5
5	13	16	4	1	0	0	0	16	4	1	0	0	0
9	15	19	4	2	2	0	0	19	4	2	2	1.7	1.7
2	15	22	4	3	0	0	0	22	4	3	0	0	0
19	10	30	4	6	4	4.5	3.6	22	4	3	4	4.2	4.2
1	15	12	1	3	0	0	0	12	1	3	0	0	0
9	18	24	2	6	0	0	0	24	2	6	0	0	0
19	16	27	3	6	5	6.3	3.6	16	3	2	6	5.8	5.8
19	19	24	3	5	5	6.1	4.4	16	3	2	6	5.6	5.6
10	6	29	2	8	0	0	0	29	2	8	0	0	0
8	13	35	2	10	0	0	0	35	2	10	0	0	0
14	14	22	4	3	1	0	0	22	4	3	4	3.1	3.1

available equipment during the VoD network design is critical as the choice of equipment has a direct impact on the minimum demand that makes caching profitable. Moreover, selecting equipment that jointly matches streaming and storage requirements at each location can result in substantial reductions in network cost; we showed in Scenario3 that only a few equipment changes can have a major impact.

We are extending the design tool to support more than one type of equipment per location, and determine not only the number of servers at each site, but also select which type of equipment yields the best solution. We are modifying the tool to include usage and maintenance costs for the network components subsequent to deployment.

## References

- [1] Video store magazine, March 2000. Ayanstar Comm.
- [2] D. S. A. Dan and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In *Proc. ACM Int. Conf. Multimedia*, San Francisco, CA, Oct. 1994.
- [3] J. M. Almeida, D. L. Eager, M. K. Vernon, and S. Wright. Minimizing delivery cost in scalable streaming content distribution systems. *IEEE Trans. Multimedia*, 6:356–365, April 2004.
- [4] N. Bartolini, F. Presti, and C. Petrioli. Optimal dynamic replica placement in content delivery networks. In *Proc. IEEE Int. Conf. on Networking*, Sydney, Aust., Sept. 2003.
- [5] S. Buchholz and T. Buchholz. Replica placement in adaptive content distribution networks. In *Proc. Symp. Applied Computing*, Nicosia, Cyprus, Mar. 2004.
- [6] C. Anderson. The long tail. *Wired Magazine*, October 2004. <http://www.wired.com/wired/archive/12.10/tail.html>.
- [7] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. Katz. Efficient and adaptive web replication using content clustering. *IEEE J. Selected Areas Comm.*, 21:979–994, Aug. 2003.
- [8] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, 1987.
- [9] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proc. ACM Symp. Operating Systems Principles*, Oct. 2003.
- [10] L. Kontothanassis, R. Sitaraman, J. Wein, D. Hong, R. Kleinberg, B. Mancuso, D. Shaw, and D. Stodolsky. A transport layer for live streaming in a content delivery network. 92:1408–1419, Sept. 2004.
- [11] S. V. Rompaey, K. Spacy, and C. Blondia. Bandwidth versus storage trade-off in a content distribution network and a single server system. In *Proc. Conf. Telecomm.*, Zagreb, Croatia, June 2003.
- [12] F. Schaffa and J.-P. Nussbaumer. On bandwidth and storage tradeoffs in multimedia distribution networks. In *Proc. IEEE Infocom*, Boston, MA, Apr. 1995.
- [13] J. Segarra and V. Cholvi. Distribution of video-on-demand in residential networks. In *Proc. Int. Work. Interactive Dist. Multimedia Syst.*, Lancaster, UK, Sept. 2001.
- [14] A. Vakali and G. Pallis. Content delivery networks: Status and trends. *IEEE Internet Computing* 7, 6:68–74, Nov. 2003.
- [15] M. Yang and Z. Fei. A model for replica placement in content distribution networks for multimedia applications. In *Proc. IEEE Int. Conf. Comm.*, Anchorage, AK, May 2003.