Video-on-Demand Server Selection and Placement

Frederic Thouin and Mark Coates

McGill University
Department Electrical and Computer Engineering
3480 University, Montreal, Quebec, Canada H3A 2A7
{frederic.thouin,mark.coates}mail.mcgill.ca

Abstract. Large-scale Video-on-Demand (VoD) systems with high storage and high bandwidth requirements need a substantial amount of resources to store, distribute and transport all of the content and deliver it to the clients. We define an extension to the *VoD equipment allocation problem* as determining the number and model of VoD servers to install at each potential replica location to minimize deployment costs for a given set of distributed demand and available VoD server models. We propose three novel heuristics that generate near-optimal solutions and show that the number of replica sites for networks where the load is unevenly distributed is low ($35 - 45\%$ of potential locations), but that the hit ratios at deployed replicas are high ($> 85\%$).

1 Introduction

As the number of available titles and usage of video-on-demand services is expected to grow dramatically in the next years, many providers are planning the deployment of large-scale video-on-demand (VoD) systems. These systems require significant resources (bandwidth and storage) to store the videos, distribute them to caches, and deliver them to clients. An important and complicated task part of the network planning phase is resource allocation. It consists of determining the location and number of resources to deploy such that user demand is satisfied, cost is minimized, and any quality of experience (QoE) constraints (delay, packet loss, frame loss, or packet jitter) are respected. The main challenge is to build sufficiently accurate models for all of the factors involved: the available infrastructure, the network topology, the peak/average usage of the system, the popularity of each title, and bandwidth and storage requirements.

In the case of a distributed video-on-demand network deployment, the resources to consider are the equipment required at the origin and proxy video servers and for the actual transport between each location. We assume an existing topology with a high bandwidth capacity and focus on the equipment required at each location to store and stream the content. A video server consists of storage devices to cache the desired content and streaming devices to deliver the videos to the users. In [1], we defined the *VoD equipment allocation problem* that consists of determining the number of streaming and storage devices at each location in the topology such that the demand is satisfied and the deployment cost is minimized. We showed that the nature of the equipment installed at each location has a major impact on the design and on whether it is beneficial to even cache content. A natural extension of the problem thus involves identifying the best type of equipment to install at each location when many models are available and there is flexibility for variation from site to site. Therefore, in this paper, we address the problem of determining not only the number, but also the model of the VoD servers at each potential replica location.

This paper is organized as follows. In Sect. 2, we formulate the VoD equipment allocation problem such that the solution includes both the number and model of the VoD servers. In Sect. 4, we present two simple algorithms (Full Search and Centralized or Fully Distributed Heuristic) and three novel heuristics (Greedy Search, Integer Relaxation Heuristic and Improved Greedy Search) to solve the problem. In Sect. 5, we show and discuss the results of simulation experiments performed on randomly generated topologies. Finally, in Sect. 6, we present our conclusions and suggest future extensions to our work.

1.1 Related Work and Contribution

Researchers have tackled the problem of generating cost-efficient VoD network designs using different optimization techniques: placement of replica servers, video objects or allocation of available resources to minimize cost. Solving the replica placement [2, 3] or video placement [4] problems independently of the resource allocation problem usually leads to suboptimal solutions because the location of the replicas has a direct impact on the amount of resources required. Laoutaris et al. defined the *storage capacity allocation problem* as determining the location of each object from a set to achieve minimal cost whilst enforcing a capacity constraint [5]. Although they determine the actual storage requirements at each node with their solution, the authors do not explicitly determine the equipment required. In [6], Wauters et al. define an Integer Linear Programming (ILP) model built on viewing behavior, grooming strategies, statistical multiplexing and Erlang modeling to specify the equipment required for transport (the number of ports, multiplexers and switch ports) at each of the candidate network nodes [6]. Thouin et al. defined the *VoD equipment allocation problem* in [1] as the task of determining the number of VoD servers (which include both a storage and streaming device) to deploy at each potential location in a network topology such that the total demand is satisfied and the deployment cost is minimized. Solving the VoD equipment allocation problem determines the location of the replicas, the amount of storage available to cache content, the streaming capacity available to serve clients and the explicit specification of the equipment installed at each location. However, our approach in [1] assumed that a fixed, single and predetermined type of VoD server was available at each location. This constraint rarely holds in practice and enforcing it leads to suboptimal designs if the nature of the equipment is not a good fit to the streaming (user demand) and storage (library size) requirements. This optimization problem also has some similarities with the classical facility location problem which has been studied thoroughly (many algorithms and exact heuristics have already been developed to solve it). However, the presence of an origin server that gathers traffic from all other locations and the non-linearity in some constraints make our problem substantially different even from the generalized form of the facility location problem proposed in recent work [7] and thus unsolvable using available heuristics.

In this paper, we re-formulate the VoD equipment allocation problem to determine both the number and model of the servers to install at each location. Instead of fixing the streaming and storage capacity per VoD server at each site (the approach used in [1]), we require the pre-specification of a set of available VoD servers and select the model at each location that minimizes total network cost. This leads to the faster generation of lower-cost solutions because the network designer does not need to manually try all models for each potential site. To solve the problem, we develop a network cost model solely in terms of the numbers and models of servers and propose three novel heuristics: the Integer Relaxation Heuristic and two greedy-search based algorithms (GS and IGS).

2 Problem Statement

We consider a metropolitan area network with one origin server and N potential replica locations such as the one depicted in Fig. 1. Each cluster of clients has worst-case demand M_i (peak usage demand) and is assigned to a potential replica location with hit ratio h_i that represents an estimate of the fraction of M_i served at that replica, the other portion is served directly by the origin server.

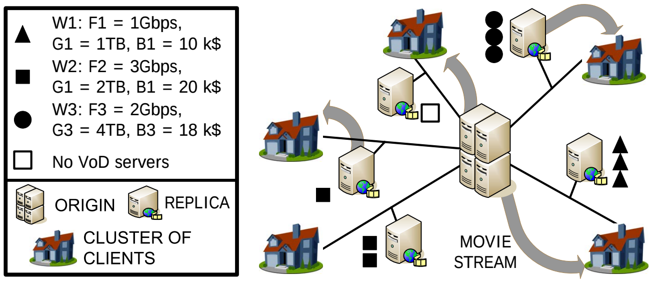

▲ W1: F1 = 1Gbps, G1 = 1TB, B1 = 10 k$
■ W2: F2 = 3Gbps, G1 = 2TB, B1 = 20 k$
● W3: F3 = 2Gbps, G3 = 4TB, B3 = 18 k$
□ No VoD servers
ORIGIN
REPLICA
CLUSTER OF CLIENTS
MOVIE STREAM

Fig. 1. Video-on-Demand equipment allocation problem. Logical connectivity between origin, $N = 5$ potential replica server locations and clients. Clients' requests (shown as movie stream arrows) are served by replicas (if content is available) or by origin. Key shows the specifications of $W = 3$ different VoD server models. We show the number and type of VoD servers installed at each potential location. The optimal solution can include locations with no equipment (empty square).

We address the *VoD equipment allocation problem* of determining not only the number, but also the model of the VoD servers at each potential replica location. To solve this problem, we require the specification of a set of available VoD server models $\mathcal{W} = \{w_j : j = 1, \ldots, W\}$ where w_j is a VoD server with streaming capacity F_j Gbps, storage capacity G_j TB and unit cost B_j k\$. We define the sets $\mathcal{N} = \{n_i : i = 1, \ldots, N\}$ and $\mathcal{V} = \{v_o, v_i \in \mathcal{W} : i = 1, \ldots, N\}$ where n_i is the number and v_i is the model of the servers installed at location i. The optimization problem is expressed as follows:

$$\{\mathcal{N}^*, \mathcal{V}^*\} = \arg \min_{\mathcal{N}, \mathcal{V}} C_{\mathrm{TOT}_{\mathcal{V}}}(\mathcal{N}) \tag{1}$$

where $C_{\mathrm{TOT}_{\mathcal{V}}}(\mathcal{N})$ is the total cost of the network C_{TOT} for a fixed set $\mathcal{V}$.

$$n_o \cdot G_o \geq Y \cdot \text{file size} \tag{2}$$

$$n_i \cdot F_i \geq h_i \cdot M_i, \quad \forall i \in \{1 \ldots N\} \tag{3}$$

$$n_o \cdot F_o \geq \sum_{i=1}^{N} (1 - h_i) M_i \tag{4}$$

$$\widehat{H}(n_i \cdot G_i) \geq h_i, \quad \forall i \in \{1 \ldots N\} \tag{5}$$

The first constraint states that the storage capacity at the origin must be large enough to host the entire initial library where Y is the number of files in the library. The constraints in (3) and (4) ensure that the streaming capacity at each replica and the origin is large enough. For each replica sites, the streaming capacity required is at least the fraction of the user demand coming from its associated cluster of clients. For the origin, the total demand is equal to the sum of all residuals fractions of the demand that are not handled by the replicas. In (5), we introduce $\widehat{H}$, an estimate of the hit ratio as a function of the storage capacity (we give more details about the form of $\widehat{H}$ in Sect. 3). This constraint states that the amount of storage at every location should be large enough such that the estimated hit ratio is greater than the actual hit ratio h_i; that a fraction of the requests equal to h_i are for files stored at the replica.

3 Network Cost Model

In order to perform a direct optimization, we derive an expression for the deployment cost solely in terms of $\mathcal{N}$ and $\mathcal{V}$. We express the total cost C_{TOT} as the sum of the cost of infrastructure, C_T, and the cost of transport, C_S:

$$C_T = f_1(n_o, v_o) + \sum_{i=1}^{N} f_1(n_i, v_i) \qquad C_S = \sum_{i=1}^{N} f_2(h_i, M_i)$$

$$C_{\mathrm{TOT}} = f_1(n_o, v_o) + \sum_{i=1}^{N} f_1(n_i, v_i) + f_2(h_i, M_i) \tag{6}$$

The cost of infrastructure at each potential location and the origin includes a start-up cost for installation and software (A_i) and increases linearly with the number of VoD servers installed (n_i). Note that f_1 is also a function of v_i which defines B_i, F_i and G_i.

$$f_1(n_i, v_i) = A_i + B_i n_i \tag{7}$$

The cost of transport for each location is inspired from the model proposed by Mason et al. is divided in two components: transport from the replica to the clients ($C_{S_{RC_i}}$) and from the origin to the replica ($C_{S_{OR_i}}$) [8]. $C_{S_{RC_i}}$ includes the cost of network interfaces (C_{IF}) and fiber (C_f). The number of network interfaces (n_{RC_i}) required is proportional to the demand M_i and the fiber capacity (c). For transport from the origin to the replica location, we add the cost of DWDM with w_{max}-ports multiplexers (C_{DWDM}) and line amplifiers (C_{LA}). The number of network interfaces (n_{OR_i}) required is a function of the demand M_i and the hit ratio h_i: the amount of traffic on this link is equal to the fraction of the demand un-served by the replica. For more details on the cost functions f_1 and f_2, the reader is referred to [1].

$$\begin{aligned} C_{S_{RC_i}} &= n_{RC_i} \cdot (2 \cdot C_{IF} + d_{RC_i} \cdot C_f) \\ C_{S_{OR_i}} &= n_{OR_i}(2 \cdot C_{IF}) + \frac{n_{OR_i}}{w_{max}} \left[2C_{DWDM} + d_{OR_i} \cdot C_f + \left(\frac{d_{OR_i}}{max_{amp}} \right) \cdot C_{LA} \right] \end{aligned}$$

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

$$f_2(h_i, M_i) = C_{S_{OR_i}} + C_{S_{RC_i}} \tag{8}$$

To derive an expression for C_{TOT} solely in terms of n_i, we resolve the hit ratio h_i and number of servers at the origin n_o as functions of n_i. To estimate the hit ratio $\widehat{H}$, we use (9), a function of the cache size ratio X_i (number of files in the cache / number of total files in the library), the library size Y and the number of files added to the library every week Z. We designed the function and determined best-fit constants K_1 to K_8 using a discrete-time simulator based on the file access model proposed by Gummadi et al. in [9] (refer to [1] for more details).

$$\widehat{H} = A(Y, Z) + B(Y, Z) \cdot \log(X) \tag{9}$$

$$A = K_1 + K_2 Z + K_3 \log(Y) + K_4 Z \log(Y) \qquad B = K_5 + K_6 Z + K_7 Y + K_8 Z Y$$

The hit ratio at a location is limited by either the streaming or the storage capacity represented by constraints shown in (3) and (5). We isolate h_i in both expressions and define $f_3(n_i, v_i)$ as the minimum (worst-case) hit ratio:

$$f_3(n_i, v_i) = \min \left[\frac{n_i \cdot F_i}{M_i}, \widehat{H} \left(\frac{n_i \cdot G_i}{Y \cdot \text{file size}}, Y, Z \right) \right] \tag{10}$$

The number of servers required at the origin, n_o, is also constrained by either streaming or storage (shown in (4) and (2)). In (11), we define n_o as $f_4(\mathcal{N}, \mathcal{V})$ by substituting h_i with the expression in (10).

$$f_4(\mathcal{N}, \mathcal{V}) = \max \left[\frac{\sum_{i=1}^{N}(1-h_i)\cdot M_i}{F_o}, \frac{Y \cdot \text{file size}}{G_o} \right] \tag{11}$$

We replace the equations for n_o and h_i in (6):

$$C_{\text{TOT}} = f_1(f_4(\mathcal{N}, \mathcal{V})) + \sum_{i=1}^{N} f_1(n_i) + f_2(f_3(n_i, v_i)) \tag{12}$$

4 Description of Heuristics

4.1 Full Search (FS)

The Full Search is a very straightforward approach that consists of trying all the possible points in the solution space. We reduce this space by calculating the maximum number of servers it is worth installing at a given location using (13). We define $\mathbf{ub} = \{ub_i : i = 1, \ldots, N\}$ where ub_i is the number of servers required to store the entire library and handle 100% of the requests ($h_i = 1.0$).

$$ub_i = \max\left(\frac{M_i}{F_i}, \frac{Y \cdot \text{file size}}{G_i}\right) \tag{13}$$

For a given $\mathcal{V}$, the boundaries of the solution space are $\mathcal{N} = \mathbf{0}$ to $\mathbf{ub}$ where $\mathbf{0} = \{n_i = 0 : i = 1, \ldots, N\}$. To complete the full search, all the possible combinations of $\mathcal{V}$ must also be tried. Although this procedure is guaranteed to find the optimal solution, it is very computationally expensive and the amount of time to search the entire space grows exponentially with the size of the network.

4.2 Central or Fully Distributed Heuristic (CoFDH)

The Central or Fully Distributed Heuristic simply calculates the cost of a centralized design ($\forall i : n_i = 0$) and a fully distributed design ($\forall i : n_i = ub_i$) for each available VoD server model in $\mathcal{W}$ and picks the cheapest design. This heuristic is straight-forward and highly suboptimal, but it provides an upper-bound that can be used as a comparison base for other approaches.

4.3 Greedy Search (GS)

We define a search topology in the discrete solution space where each solution is connected to its neighbouring solutions. In this case, a neighbour consists of adding one server at one of the locations. Greedy Search is a searching heuristic that explores all neighbouring nodes and selects the one that yields the best solution at every iteration without considering the subsequent steps [10].

4.4 Improved Greedy Search (IGS)

The Improved Greedy Search is divided into two steps inspired by GS. The difference is that a neighbour solution is obtained by adding or removing ub_i servers at location i. The motivation behind this is that the hit ratio at replica location is often very high which leads to n_i close to ub_i. During the first step of the heuristic, we iteratively add servers in a greedy-fashion starting from a centralized design by setting $n_i = ub_i$ at the location that achieves the lowest cost. We complete the first step and determine an initial integer solution by repeating this procedure for each VoD server model.

The second step is an exploration procedure in the neighbourhood of the initial solution. In a greedy-type approach, we add or remove, at iteration k, one server to the initial solution at the location that minimizes the cost C_k. We stop the search when $C_j \geq C_{j-1}$ $\forall j \in k - I + 1 \ldots k$ or when $C_j \geq C_{IGS}$ $\forall j \in k - 2I + 1 \ldots k$ (minimum cost has not decreased for 2I iterations).

4.5 Integer Relaxation Heuristic (IRH)

The first step of the Integer Relaxation Heuristic is to find a non-integer solution for each server model using a constrained nonlinear optimization algorithm based on sequential quadratic programming [11]. Then, we calculate the cost associated with each replica ($C_{T_i} + C_{S_{OR_i}}$) and determine the model that minimizes this cost for each location. We complete the initial solution by determining the best server model to install at the origin. In the second step, we perform two different searches to find a near-optimal integer solution: we iteratively (i) set $n_i = 0$ at each location to make sure it is profitable to setup a replica and then (ii) try to remove or add up to two servers at each location until we find a local minimum.

5 Simulation Experiments

Our simulation results were obtained by applying our heuristics to different networks (simulations were executed on a AMD Athlon 3000+ with 1 GB of OCZ Premier Series 400 MHz Dual Channel memory), each defined by the constant variables in Table 1 and choosing values for the other network parameters from uniform distributions with the ranges specified in Table 2 (these values were obtained from discussions with industrial partners [12]).

5.1 Complexity

Table 3 presents the size of the solution space for different network topologies (generated using the values displayed in Table 1 and Table 2) which consist of all possible number of servers ($n_i = 0$ to $n_i = ub_i$) and server models at each location. From other experiments, we measured that the machines we used for simulations explore 4000 solutions per second on average, which allows us to estimate to time it would take to explore the entire solution space in order

Table 1. Values of constants.

Variable	Value
C_{IF}	10 k$
C_{DWDM}	25 k$
C_{LA}	10 k$
C_f	0.006 k$/km
w_{max}	16
c	10 Gbps
max_{amp}	75 km
bit rate	3.75 Mbps
duration	5400 s
file size	2.53 GB

Table 2. Range of the variables.

Variable	Min	Max
d_{OR} (km)	0	50
d_{RC} (km)	0	5
Y (files)	1000	10000
Z (files/week)	0	100
priceGbps (k$/Gbps)	0	4
priceTB (k$/TB)	0	3
A (k$)	6	36
F (Gbps)	1	5
G (TB)	1	11
M (Gbps)	1	20

to determine the global optimal solution. From our estimates, it is clear that performing a full search is infeasible as even the smallest problems ($N = 10, K = 1$) can take up to thousands of days to solve depending on the demand and the specifications of the equipment. This shows that heuristics are essential to solve the VoD Equipment Allocation Problem.

Table 3. Number of possible solutions for topologies of N locations with K possible VoD server models and estimate of time taken to find the global optimal solution based on an observed average rate of 4000 solutions per second. Values obtained from 50 different topologies for each pair of (N,K).

N	K	Number of solutions			Estimated time (days)		
		min	median	max	min	median	max
10	1	7.8×10^{5}	4.5×10^{8}	9.5×10^{13}	2.2×10^{-3}	1.3×10^{0}	2.8×10^{5}
25	1	5.6×10^{14}	7.1×10^{20}	2.4×10^{35}	1.6×10^{6}	2.1×10^{12}	6.9×10^{26}
50	1	2.0×10^{28}	1.4×10^{40}	1.2×10^{68}	5.9×10^{19}	4.1×10^{31}	3.5×10^{59}
100	1	1.9×10^{60}	5.0×10^{79}	1.3×10^{130}	5.6×10^{51}	1.4×10^{71}	3.7×10^{121}
15	2	9.1×10^{13}	2.0×10^{18}	6.7×10^{23}	2.6×10^{5}	5.9×10^{9}	1.9×10^{15}
10	3	5.4×10^{10}	1.0×10^{14}	4.3×10^{17}	1.6×10^{2}	2.9×10^{5}	1.3×10^{9}

5.2 Performance

In this section, we evaluate the performance of our heuristics. In our first set of tests, we analyse small networks to allow comparison with the full search, which cannot produce a solution for larger networks within a reasonable time frame. In Fig. 2, we show the performance of our heuristics by dividing the cost of the solution by the optimal solution provided by the full search. For these small networks, Integer Relaxation Heuristic and Improved Greedy Search perform within 4% of the optimal solution. For all values of N and W, both IRH and IGS perform better than the Greedy Search.

In Fig. 3, we show values of the ratio between the cost of Integer Relaxation Heuristic, Improved Greedy Search and Greedy Search and the cost of CoFDH.

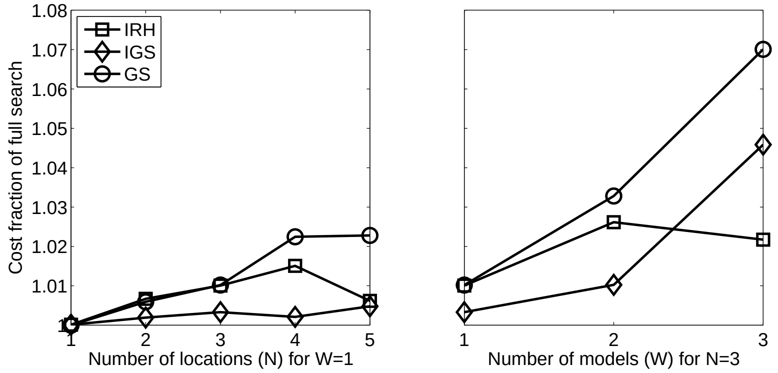


Fig. 2. Cost ratio between heuristics and the full search averaged over 30 runs for networks with the number of locations $N \in \{1, \ldots, 5\}$ and the number server model $W = 1$ (LEFT) and another series with $N = 3$ and $W \in \{1, 2, 3\}$ (RIGHT).

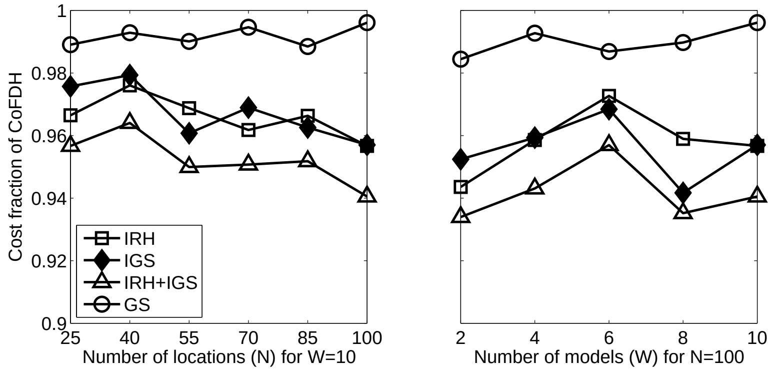


Fig. 3. Cost ratio between the heuristics solution and CoFDH averaged over 25 runs. IRH+IGS is the average of the minimum value between IRH and IGS for all runs.

Whereas Greedy Search is actually very close to the cost produced by CoFDH, the other two heuristics generate solutions that cost 2-5% less. It is not clear from those plots whether Integer Relaxation Heuristic or Improved Greedy Search performs better. By combining both (choosing the best solution of the two), we obtain a better heuristic (IRH+IGS), which achieves a 4-6% cost reduction compared to CoFDH. In the left panel, we notice the downward trend of the cost fraction as the number of locations in the network increases because more modifications to the CoFDH design can be made to reduce cost. For the same set of tests, we also observed the computational time in seconds of each heuristic. The Integer Relaxation Heuristic was the slowest of the tested heuristics because of constrained optimization using SQP, but it still converges in a reasonable amount of time (< 250 seconds for $N = 100$ and $W = 10$). Since the computation time of Improved Greedy Search is so low (< 10 seconds), we can combine IRH and IGS and obtain a solution in a timely fashion.

5.3 Analysis

Finally, we focus on the networks with six server models (similar behaviour was observed for other values of W) to analyze the ratio of locations with replicas and average demand at replica locations. The left panel of Fig. 4 shows that for networks of any size, where demand is not uniformly distributed among all locations (the demand at each location is different), the percentage of locations where a replica will be deployed is below 40% for both heuristics. Although a case where the demand load is evenly shared among all the locations is more plausible, this result indicates that it is not always advantageous to cache content. If the demand is too low then it can be more cost-effective to assume the entire load from a group of clients directly at the origin. An impact of this low percentage is the number of servers installed at the origin. Because the fraction of locations where replicas are installed remains approximately constant for any value of N, the total number of sites for which the origin must assume the demand grows as the network becomes larger.

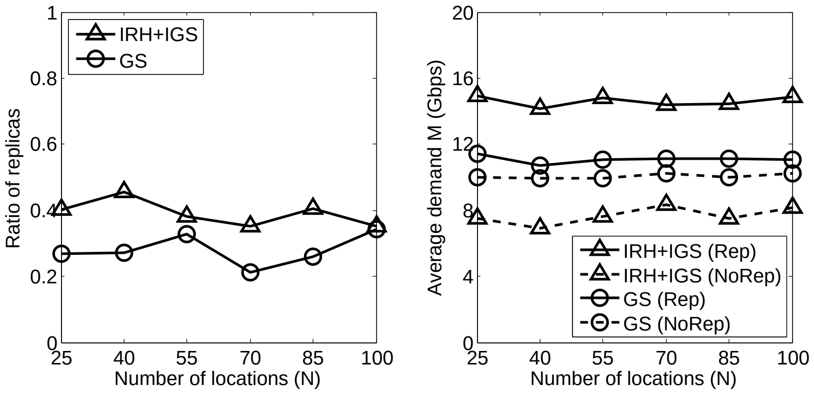
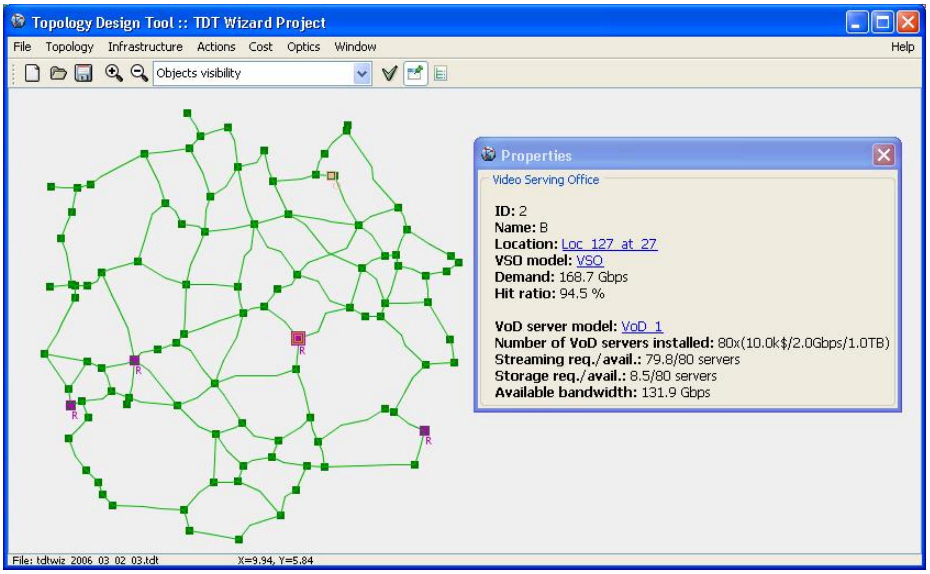


Fig. 4. LEFT: Ratio between the number of replicas (location with cached content) and potential locations. RIGHT: Average load on the locations where replicas are installed (Rep) and where no replicas are installed (NoRep). The values shown are averages of 25 runs with $W = 6$.

In the right panel of Fig. 4, we depict the difference between the average demand at replica locations and that at sites where no caching is performed. Whereas there is only a marginal difference in the GS case, the average demand at replica sites in the IRH+IGS solutions is almost twice the average demand of the other locations. The solutions generated by combining Integer Relaxation Heuristic and Improved Greedy Search have a much lower total cost than the GS solutions, indicating that it is more cost-efficient to install replicas at locations where demand is high and transport the entire load of locations with low demand to the origin.

6 Concluding Remarks

In this paper, we defined an extension of the *VoD equipment allocation problem* described in [1]. Instead of considering fixed and pre-determined streaming and storage capacity at each location, we require the specification of a set of available VoD servers models. The optimization problem consists of choosing the number and type of VoD servers to install at each potential location in the network such that cost is minimized. For most topologies, we showed that it is infeasible to obtain the global optimal solution. We described three heuristics to find a near-optimal solution including two greedy-type approaches (GS and IGS) and an integer relaxation method (IRH) that we implemented in an interactive design tool shown in Fig. 5. We combined IRH and IGS by choosing the best of the two to obtain a better solution while maintaining low computational time.



ID: 2
Name: B
Location: Loc 127 at 27
VSO model: VSO
Demand: 168.7 Gbps
Hit ratio: 94.5 %

VoD server model: VoD 1
Number of VoD servers installed: 80x(10.0k$/2.0Gbps/1.0TB)
Streaming req./avail.: 79.8/80 servers
Storage req./avail.: 8.5/80 servers
Available bandwidth: 131.9 Gbps

File: tdtwiz 2006 03 02 03.tdt X=9.94, Y=5.84

Fig. 5. Screenshot of the design tool that implements our heuristic (IRH+IGS) to solve the VoD equipment allocation problem. A sample topology of potential replica locations and the properties window of a selected replica location is shown in the figure.

We saw that a way to obtain a cost-efficient solution is to use equipment (VoD server model) that satisfies the streaming and storage requirements of most of the locations in the topology. Alternatively, the network designer could strive to divide the demand evenly among all locations such that it is optimal to deploy replicas at most locations using the same model of equipment. A sensible extension to the resource allocation problem we addressed in this paper is the problem of jointly designing the VoD network and the logical topology: choosing a topology that allows an allocation of resources which minimizes the deployment cost of the network. We also observed that limiting ourselves to a single VoD server model per location, solutions produced with our heuristics often deploy the same model at all replica sites. Our preliminary results show that relaxing this constraint allows finer tuning and the possibility to match more closely the demand in storage and streaming at each location and leads to cost savings.

Other future research avenues include the scenario where the service provider owns network equipment or infrastructures prior to the deployment. However, even if there is no installation cost, there are still fees incurred by the usage and maintenance of the equipment and the resources, which have to be considered when generating solutions for this scenario. Also, we focused on large-scale deployments, but there is also the issue of scalability of such deployments. As the library reaches tens of thousands of movies, the access model we assumed changes as a larger portion of requests are located in the heavy tail of the popularity distribution ('long-tail' of content).

References

[1] F. Thouin, M. Coates, and D. Goodwill, "Video-on-demand equipment allocation," in *Proc. IEEE Int. Conf. Network Computing Applications (NCA)*, Cambridge, MA, July 2006.

[2] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Trans. Networking*, vol. 8, pp. 568–582, Oct. 2000.

[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*, vol. 6, pp. 356–365, April 2004.

[4] W. Tang, E. Wong, S. Chan, and K. Ko, "Optimal video placement scheme for batching vod services," *IEEE Trans. on Broad.*, vol. 50, pp. 16–25, Mar. 2004.

[5] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, "On the optimization of storage capacity allocation for content distribution," *Computer Networks Journal*, vol. 47, pp. 409–428, Feb. 2005.

[6] T. Wauters, D. Colle, M. Pickavet, B. Dhoedt, and P. Demeester, "Optical network design for video on demand services," in *Proc. Conf. Optical Network Design and Modelling*, Milan, Italy, Feb. 2005.

[7] L.-Y. Wu, X.-S. Zhang, and J.-L. Zhang, "Capacitated facility location problem with general setup cost," *Comp. Oper. Research*, vol. 33, pp. 1226–1241, May 2006.

[8] L. Mason, A. Vinokurov, N. Zhao, and D. Plant, "Topological design and dimensioning of agile all photonic networks," *Computer Networks, Special issue on Optical Networking*, vol. 50, pp. 268–287, Feb. 2006.

[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. OS Principles (SOSP)*, Bolton Landing, NY, Oct. 2003.

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms.* Cambridge, MA: The MIT Press, 1990.

[11] R. Fletcher, *Practical Methods of Optimization.* New York, NY: John Wiley and Sons, 1987.

[12] D. Goodwill, private communication, Nortel Networks, 2005.