

Dissemination of Dynamic Multimedia Content in Networked Virtual Environments

Tolga Bektaş, Farzad Safaei, Iradj Ouveysi and Osman Oğuz

Abstract—This paper aims to design an infrastructure for distribution of dynamic multimedia objects in a Networked Virtual Environment (NVE) based on a distributed proxy architecture. Dynamic objects (such as audio, video and images) are generated by the users on-the-fly, are often short lived and must be disseminated to others in their area of interest while the application is running. It is expected that a significant portion of network traffic for future NVE applications will be due to these objects. However, because of the upstream bandwidth constraints of clients, a peer-to-peer model for dissemination of these objects is unlikely to be scalable. We develop a distributed server infrastructure for this purpose and produce a mathematical model for optimal establishment of proxy servers, assignment of clients to proxies, and replication pattern of objects among the proxies. The formulation is non-linear (quadratic) in both the objective function and one set of constraints. We provide a technique to linearize the model, which provides the opportunity to solve reasonable size problems for benchmarking. In addition, we provide a fast and efficient heuristic that can be used to obtain near-optimal solutions to the problem in real-time. Finally the paper concludes with computational results showing the performance of the linearization procedure and the heuristic algorithm on randomly generated Internet topologies. It is shown that our heuristic algorithm produces solutions that are within 20% of the optimal.

Index Terms—networked virtual environment, integer programming, heuristic algorithm.

I. INTRODUCTION

THERE has been a significant increase in popularity of Networked Virtual Environments (NVE) in recent years. It has been estimated that by 2009 more than 230 million people will be playing multiplayer network games [1]. Other collaborative applications for work, education and entertainment are also likely to become widespread.

NVEs represent a possible approach for utilizing the power of Internet for enhancing group communication and interaction, where several participants (perhaps hundreds or thousands) can enter a common virtual environment for the purpose of sharing an experience, collaboration, recreation and so forth. Within the virtual environment, the participants are typically represented by their *avatars*. Each participant can control his/her avatar using a suitable interface (keyboard and

mouse, game controller, gestures, etc.). Through this control and depending on the design and purpose of the application, participant's avatar may move within the virtual environment, interact with objects or other avatars and perform a variety of tasks to achieve its goals. The interactivity, therefore, is not confined to a web-style point-and-click and involves a more substantive and immersive kind - i.e., virtual presence and participation.

In the current generation of NVE's, the visual and audio scenes associated with the virtual environment are commonly comprised of static content only. The images of avatars, the environment (walls, rooms, mountains, trees), and other objects are created offline either by application designers or, in some cases, the users themselves and disseminated beforehand. Likewise, the audio objects such as the sound of closing a door, an explosion, waves or wind are pre-recorded and distributed with the application software. (In some cases, the sound may also be created synthetically using a suitable model for the physics of objects and collision.) Hence, the visual and audio scenes are created by the end clients based on information that is available locally and the real-time exchange of information across the network is primarily limited to the *state information* exchange.

The state information is used to communicate changes in the state of the virtual environment due to actions of participants (moving, rotating, shooting, etc). The aim is to create a consistent perception for all. Significant research effort in recent years has been devoted to design of suitable models for traffic characterization and processing of state information so that the NVE can be deployed cost effectively, accommodate a large number of players and scale with respect to geographical spread of participants (see [2] and references therein).

We are interested in situations where a significant portion of multimedia content within the virtual environment is *dynamic*. By 'dynamic' we mean that these objects are created on-the-fly, are often short lived and must be disseminated to others while the application is running. To a limited extent this is already happening. Most NVE's support some form of communication mechanism such as text chat or a single channel voice 'party line'. This information is often broadcast to everyone or a subset of participants. More advanced systems for immersive voice communication have been described in [3] and commonly require a separate server infrastructure for creation and delivery of personalized audio scenes for each avatar. In the future, we expect to see more dynamic objects including video and images in these environments. These could include images of participants, artifacts created by the users and shared, presentation slides or video clips for education

T. Bektaş and O. Oğuz are with the Department of Industrial Engineering of Bilkent University, 06800, Ankara, Turkey (e-mail: tolgab@bilkent.edu.tr, ooguz@bilkent.edu.tr)

Farzad Safaei is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, NSW 2522, Australia (email: farzad@uow.edu.au).

I. Ouveysi is with ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, University of Melbourne, Australia (email: iradjouveysi@yahoo.co.uk).



or collaboration, and gesture and haptics information. As the multimedia communication aspect of NVE's become richer and more immersive, having a common delivery platform for distribution of dynamic multimedia objects, as opposed to designing separate infrastructure for each, becomes more critical.

A conceptually simple model for dissemination of dynamic multimedia objects is to use a peer-to-peer model. In this case, the source of the dynamic object will have to multicast the object to all other clients that might be interested in this object. For example, audio and visual information should be streamed to every other avatar who happen to be within the hearing or visual range. This model has scalability issues especially with respect to access and network bandwidth utilization and latency of dissemination.

This paper aims to develop a suitable model based on a distributed proxy architecture to support dissemination of dynamic multimedia objects associated with the networked virtual environments. In the next Section, we describe the distributed proxy architecture of interest to this paper and identify the characteristics and requirements for effective dissemination of content. In Section 3, a mathematical model is developed to optimize the choice of proxy servers, assignment of clients to proxies, and replication of multimedia objects among the proxies. We will then use a suitable technique to linearize the formulation and develop a heuristic model for fast and efficient solution. The computation results and conclusions are presented next.

II. SYSTEM DESCRIPTION

There is clearly some similarity between dissemination of dynamic multimedia objects amongst the participants of a NVE and content distribution networks (CDN) for delivery of web or streaming content in today's Internet. There are, however, some stark differences especially with respect to temporal and spatial aspects of demand distribution. The following considerations are relevant.

- The lifetime of dynamic multimedia objects may range from immediate (real-time streams like voice and video) to short-lived (images or presentation slides) to semi-permanent (new artifact added to the virtual environment). Consequently, timeliness and bandwidth cost of object replication among the distributed servers is a critical factor in designing the infrastructure.
- Within the virtual environment, a dynamic multimedia object is of relevance only to a subset of participants. For example visual objects are visible within a range and need not be distributed to all users. This 'area of interest' of an object depends on many contextual factors. The characteristics of the 'map' of the environment will impact the visual and audible range of objects (for example, presence of walls, doors, and buildings). The 'rules' of the virtual environment is also of relevance (for example, capabilities of avatars to zoom in on remote objects or obtain high-speed transportation). In addition, the 'area of interest' may differ for different types of media (audio and music, for example, could propagate

through walls but not video). The combination of above factors leads to a rather complex demand model where the user's demand for dynamic objects depends more on their location and behaviour within the virtual environment than their physical distribution around the Internet. So in addition to dynamic content, the demand pattern is also highly dynamic due to often rapid movement of avatars in the virtual world.

- In CDN, the origin servers owned by the content provider contain the original content, which is then distributed for replication among the surrogates. For NVE's, the origins of dynamic objects are users/clients. It is reasonable to assume that the upstream bandwidth of the clients is not very large. Hence the 'cost' of download directly from the origin is very high.

In this paper, we propose to use a distributed proxy architecture for dissemination of dynamic multimedia objects within a NVE. This architecture has already been used for immersive voice communication [4] and our intention here is to extend the concept to the more general case of multimedia communications. We assume that the service provider has access to a set of distributed servers - referred to as proxies - across the Internet. In general, the service provider might be hosting more than one NVE, hence, a given NVE might use only a subset of proxies for the distribution and delivery of its dynamic objects. This is because the number and distribution of clients associated with each NVE has a significant influence on the choice of proxies needed. During different time periods, however, these characteristics might change. For example, the number of participants in an entertainment NVE may increase during night time. In this case, it might be suitable to add new proxies to the service. This addition has certain overhead and configuration cost that we refer to as the proxy "establishment" cost.

Every client must be assigned to an established proxy server. Each proxy, in turn, is responsible for a group of clients and performs the necessary functions of distribution, replication/catching of objects. The combination of these proxies will form an overlay network of distributed servers.

It is reasonable to assume that the network path connecting each client to its assigned proxy is known and cannot be controlled by us. (This usually is an Internet routed path from the user through its ISP Point of Presence and its quality is influenced by the topology of ISP infrastructure and subscription options available.) The service provider, however, can have some control over the communication paths between the proxies. Obviously, each proxy can communicate with every other proxy using an Internet routed path. However, in some cases the service provider could hire better quality communication paths between a subset of proxies. As an example, a Virtual Private Network provided by a carrier could be used to connect some proxies together if available/economical. For our purpose, it is sufficient to assume that there is connectivity between all proxies, but some of the overlay paths may offer better quality and bandwidth resources than others (usually at additional cost).

As stated above, we assume that clients are the *source* of dynamic objects but due to upstream bandwidth limitations,

only one copy is sent to their assigned proxy. The proxy then becomes the ‘origin server’ for this object for the purpose of replication among other proxies for the duration of object’s lifetime. The clients are also the source of demand for other objects. This demand pattern, however, is highly dynamic and heavily dependent on the context of the virtual environment. To simplify the task of estimating demand, we partition the virtual environment into a number of *communication zones* for each type of object (audio, video, etc.). The avatars within the same communication zone would require each others objects. Within the lifetime of an object, there is also a possibility of other avatars moving into its communication zone. As such, we require a demand estimator that provides an estimate of demand from each avatar for a given dynamic object for the time period of interest(which could be in the order of tens of seconds to hours depending on the dynamics of application). The method of partitioning the virtual environment into communications zone and estimation of demand based on roaming behavior of avatars is outside the scope of this paper and will be subject of future publications.

III. INTEGER PROGRAMMING FORMULATION

The problem of interest to this paper can be stated as follows. Given a particular set of dynamic objects created by the participants of a NVE within a time period, and given the estimated demands for these objects from other clients in this time period, our aim is to determine: (i) the subset of the distributed proxies that should be established for dissemination of these objects; (ii) assignment of each client to a proxy; and (iii) the replication pattern of objects among the proxies. Note that we have not included any request routing decision here. We have assumed that on the short time scales of interest to us, the practical approach for a client would be to obtain the required object from its own proxy (if the object is replicated there) or to fetch it directory from the originating client. The latter option, however, is assumed to incur a higher cost and will be subject to upstream bandwidth constraint of the originating client. The formal definitions and formulation of this problem are presented next.

We consider a fully meshed network $G = (V, E)$, where V is the set of nodes and $E = (\{i, j\} : i, j \in V)$ is the edge set representing the overlay paths connecting the nodes together. The node set V is further partitioned into two nonempty, mutually exclusive and exhaustive subsets as $V = I \cup P$, where I is the set of clients, and P is the set of potential nodes on which proxy servers can be installed. The known parameters of the problem are given in Table I.

To construct the integer programming formulation, we further define the following binary decision variables:

$$y_p = \begin{cases} 1, & \text{if a proxy server at node } p \in P \\ & \text{is established} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ip} = \begin{cases} 1, & \text{if client } i \in I \text{ is assigned to} \\ & \text{proxy server } p \in P \\ 0, & \text{otherwise} \end{cases}$$

$$z_{pk} = \begin{cases} 1, & \text{if proxy server } p \in P \text{ holds object } k \in K \\ 0, & \text{otherwise} \end{cases}$$

TABLE I
PARAMETERS OF THE PROBLEM

Notation	Indication
c_{ij}	The unit cost of transferring an object over the path $\{i, j\} \in E$.
$c_{i,j(k)}$	The unit cost of transferring an object from the client who has generated the object k (denoted by $j(k)$) to a client i that requests the object directly. This is used in the case that a content requested by client i is not found in its assigned proxy server, and the client is forced to retrieve it from the originating client.
s_p	The capacity of the potential server at site p .
f_p	The establishment cost of server at site p .
K	The set of objects available in the system for a short time interval of interest.
K_i	The set of objects generated by client i in the system within the short time interval of interest ($K_i \subset K$).
b_k	The storage requirement of object $k \in K$.
β_k	The bandwidth requirement of object $k \in K$.
d_{ik}	The demand for object $k \in K$ by client $i \in I$.
B_i	The upstream bandwidth capacity of client i available for direct peer-to-peer transfer of objects generated in i to other clients.

We now present an integer programming formulation for the problem as follows (hereafter denoted by F):

$$\text{minimize } \sum_{p \in P} f_p y_p + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K} b_k d_{ik} c_{ip} z_{pk} x_{ip} + \quad (1)$$

$$\sum_{i \in I} \sum_{p \in P} \sum_{k \in K} b_k d_{ik} (1 - z_{pk}) c_{i,j(k)} x_{ip}$$

s.t.

$$\sum_{p \in P} x_{ip} = 1, \quad \forall i \in I \quad (2)$$

$$x_{ip} \leq y_p, \quad \forall i \in I, p \in P \quad (3)$$

$$\sum_{k \in K} b_k z_{pk} \leq s_p y_p, \quad \forall p \in P \quad (4)$$

$$\sum_{i \in I, i \neq l} \sum_{p \in P} \sum_{k \in K_l} \beta_k x_{ip} (1 - z_{pk}) \leq B_l \quad \forall l \in I \quad (5)$$

$$y_p \in \{0, 1\}, \quad \forall p \in P \quad (6)$$

$$x_{ip} \in \{0, 1\}, \quad \forall i \in I, p \in P \quad (7)$$

$$z_{pk} \in \{0, 1\}, \quad \forall p \in P, k \in K \quad (8)$$

The objective function is given in (2). Here, the first summation represents the total cost of establishing proxy servers for this NVE during this time period. The second summation denotes the total cost of delivery of objects to the clients. The first part of this summation is for the case when client i receives content k that is located in its own proxy p (reflected by the cost $b_k d_{ik} c_{ip} z_{pk} x_{ip}$ summed over all the proxies, clients and objects). In the case when the requested object is not located in its proxy server, an additional cost incurs to further request the object from the generating client. This is reflected in the second part of the summation by $b_k d_{ik} (1 - z_{pk}) c_{i,j(k)} x_{ip}$, over all proxies, clients and objects. Similar cost functions have also been suggested in [5], [6], [7], and [8]. Constraint (3) implies that a client can be assigned to a node only if a proxy server is established on that node. Constraint (4) implies that the total size of objects held in each proxy

server is constrained by the available capacity. Constraint (5) is related to the bandwidth capacity between the clients. More specifically, this constraint represents the upstream capacity limit of clients to transfer objects directly to other clients who could not fetch it from their own proxy. Finally, constraints (6)–(8) denote the integrality of the decision variables.

Formulation F belongs to the class of \mathcal{NP} -Hard problems (see Appendix I). There are several ways to solve the F presented above. One strategy would be to use techniques specifically devised for quadratic optimization problems. Another way is to linearize the formulation that will enable one to solve the resulting linearized formulation via a commercial optimization package. This is presented in the following section.

IV. MODEL LINEARIZATION

It is clearly seen that the objective function (2) of F contains a quadratic term due to the multiplication of the x_{ip} and z_{pk} variables. We propose a transformation procedure to linearize the model using a continuous variable and two sets of constraints. The following is the basis of our linearization.

Proposition 1: The following constraints are sufficient to linearize the preceding model,

$$\varphi_{ipk} \leq x_{ip}, \quad \forall i \in I, p \in P, k \in K \quad (9)$$

$$\varphi_{ipk} \leq z_{pk}, \quad \forall i \in I, p \in P, k \in K \quad (10)$$

where $\varphi_{ipk} = z_{pk}x_{ip}$ and is a continuous variable in $[0, 1]$.

The proof of Proposition 1 is provided in Appendix II. Note that the linearizing variable φ_{ipk} is actually an indicator of whether client i is connected to the proxy server p and the proxy server holds the requested object k or not. Based on the preceding proposition, we can now substitute the quadratic term $x_{ip}z_{pk}$ that is present in both the objective function (2) and constraints (5) by the linearization variable φ_{ipk} . Under the proposed linearization, the integer linear programming formulation of the problem can be given as follows (hereafter denoted by F_L):

$$\begin{aligned} \text{minimize } & \sum_{p \in P} f_p y_p + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K} b_k d_{ik} c_{i,j(k)} x_{ip} \quad (11) \\ & + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K} b_k d_{ik} \varphi_{ipk} (c_{ip} - c_{i,j(k)}) \end{aligned}$$

s.t.

$$\begin{aligned} \sum_{p \in P} x_{ip} &= 1, & \forall i \in I \\ x_{ip} &\leq y_p, & \forall i \in I, p \in P \\ \sum_{k \in K} b_k z_{pk} &\leq s_p y_p, & \forall p \in P \\ \sum_{i \in I, i \neq l} \sum_{p \in P} \sum_{k \in K} \beta_k (x_{ip} - \varphi_{ipk}) &\leq B_l & \forall l \in I \quad (12) \\ y_p &\in \{0, 1\}, & \forall p \in P \\ x_{ip} &\in \{0, 1\}, & \forall i \in I, p \in P \\ z_{pk} &\in \{0, 1\}, & \forall p \in P, k \in K \\ \varphi_{ipk} &\in [0, 1], & \forall i \in I, p \in P, k \in K \end{aligned}$$

The (linearized) formulation F_L can now be solved using any integer programming solver. However, bearing in mind that the number of users and the objects existent in a NVE can be huge, solving the preceding formulation for such a data set during short time periods of interest would be impractical. Therefore, we develop a heuristic procedure in the next section that will be used to solve the problem fast, is scalable, and provides reasonably good solution quality.

V. A HEURISTIC ALGORITHM

Given any ordering $\{1, 2, \dots, |P|\}$ of the set of potential proxy locations, the heuristic begins with establishing the first proxy in the first iteration and continues on establishing the p^{th} proxy of the ordering in the p^{th} iteration, until all the potential proxies are engaged to the system. The ordering chosen here can be such that the proxies are sorted in the nondecreasing order of f_p 's or the nonincreasing order of s_p 's. At every iteration, users are connected to an established proxy that incurs the minimum cost. Then, the objects are replicated based on what we refer to here as the *saving* of an object $k \in K$, calculated as follows:

$$\pi_{pk} = b_k c_{p,j(k)} \sum_{i \in N(p)} d_{ik} \quad (13)$$

where $N(p)$ denotes the set of clients connected to proxy p . A similar measure has also been used by Xuanping et al. [9]. Verbally, the saving of an object is the amount of cost reduced by placing object k on a proxy p . Then, the objects are sorted in the non-increasing order of their savings. Let $\{O_{p1}, O_{p2}, \dots\}$ denote this order. The objects are placed in the available proxy server(s) using this order without violating the capacity constraints. We also use $b(L)$ to denote $\sum_{k \in L} b_k$. The outline of the heuristic is given in the following:

Heuristic procedure

0. Let the best solution be $\bar{c} = +\infty$ and $l = 1$.

1. Repeat the following while $l \leq |P|$:

1.1. Select the first l proxies to be opened,

(denoted by $P \supseteq \bar{P} = \{1, \dots, l\}$).

1.2. for each client $i \in I$

let $x_{ip^*} := 1$ such that

$$\sum_{k \in K} b_k d_{ik} c_{ip^*} = \min_{j \in \bar{P}} (\sum_{k \in K} b_k d_{ik} c_{ip})$$

1.3. for each server $p \in \bar{P}$

Sort the objects and let the ordering be

$\{O_{p1}, O_{p2}, \dots, O_{p|K|}\}$.

$L := \emptyset$.

$t := 1$

while $b(L) \leq s_p$

begin

$k := O_{pt}$

if $b(L \cup \{k\}) \leq s_p$

$z_{pk} := 1$

$L := L \cup \{k\}$

$t \leftarrow t + 1$

end

1.4. Check current solution. If feasible, then

Calculate the cost of the current solution c^l .

If $c^l < \bar{c}$, set $\bar{c} = c^l$.

1.5. $l \leftarrow l + 1$.

Here, set L given in the algorithm is used to record the set of objects located on a proxy. At step 1.4., the solution is checked with respect to constraint (5). If the solution does not satisfy this constraint, it is discarded. On the other hand, if the solution does satisfy this constraint and is better than the current solution with respect to cost, it is set as the incumbent solution.

VI. COMPUTATIONAL RESULTS

In order to assess the computational performance of the proposed model and the heuristic algorithm, we have performed a set of computational experiments. These experiments are based on randomly generated Internet topologies, using an Internet topology generator, available at the web address <http://topology.eecs.umich.edu/inet/>, which mimics the characteristics of the real Internet topology. The instances used for comparison purposes have different number of potential proxy locations, clients and objects (denoted by $|P|$, $|I|$, and $|K|$ respectively). A number of nodes are randomly selected to be potential proxy nodes and client nodes for each topology. The size of each object is chosen from a uniform random variable between 0 and 1. The fixed cost of installing a proxy server is also a uniform random variable between 600 and 1000. The capacity of each proxy server is calculated as 50% of the total size of all objects. The cost of download from the originating client is set to be equal to 5 times the cost of download from its proxy. The demand distribution for the objects have been modelled using a Zipf-like distribution. The Zipf-like distribution assumes that the probability of a request for an object is inversely proportional to its popularity. More specifically, let a number of objects be ranked in order of their popularity where object i in this order is the i^{th} most popular object. Then, given an arrival for a request, the conditional probability that the request is for object i is given by $P_K(i) = \frac{\Omega}{i^\alpha}$, where $\Omega = (\sum_{i=1}^K \frac{1}{i^\alpha})^{-1}$ is a normalization constant and α is an exponent. When $\alpha = 1$, we have the true Zipf-distribution. In [10], it is shown that α varies from 0 to 1 for different access patterns and is usually between 0.64 and 0.83 for web objects. This value was recorded to be $\alpha = 0.733$ for multimedia files in [6]. We have used this specific value in our implementation. The metric used to assess each solution is a *normalized cost* metric, as used in other studies (e.g. [11]), which is defined as *normalized cost = cost of the network output by the procedure/cost of the network without any replicated proxies*. Here, the cost of the network without any replicated proxies is the scheme where all the clients are assumed to retrieve the required objects from the originating clients directly. Note that the smaller the normalized cost, the better the solution found by our procedure.

For the experiments, random problems have been generated with 3 to 5 potential proxy locations, 10 clients and 20 to 80 objects. We solve each problem once using the linearized formulation F_L by employing a state-of-the-art commercial

TABLE II
COMPUTATIONAL ANALYSIS OF THE PROPOSED MODEL AND THE
HEURISTIC ALGORITHM

$ P $	$ I $	$ K $	v_{model}	Time (seconds)	$v_{heuristic}$	dev
3	10	20	0.069922	21	0.081356	16.35
3	10	30	0.067774	52	0.074079	9.30
3	10	40	0.064923	74	0.076072	17.17
3	10	50	0.061548	227	0.072084	17.12
3	10	60	0.06113	128	0.073112	19.60
3	10	70	0.06002	626	0.066189	10.28
3	10	80	0.057473	718	0.068427	19.06
5	10	20	0.036788	121	0.041944	14.02
5	10	30	0.033101	214	0.035707	7.87
5	10	40	0.034180	409	0.037644	10.13
5	10	50	0.031248	503	0.034908	11.71
5	10	60	0.031922	1496	0.036109	13.12
5	10	70	0.030649	2217	0.036493	19.07
5	10	80	0.029493	2770	0.033503	13.60

optimization package CPLEX 9.0 running on a Sun Ultra-SPARC 12x400 MHz with 3 GB RAM, and once with the heuristic algorithm. We report our findings in Table II. In this table, the first three columns correspond to the number of potential proxy locations, number of clients and number of objects, respectively. The next column, v_{model} reports the value of the optimal solution of the F_L whereas the next column (Time (seconds)) presents the corresponding computational solution time. Column $v_{heuristic}$ reports the value of the best solution found by the heuristic algorithm. Finally, the last column indicates how much the solution found by the heuristic algorithm deviates from that of the formulation. This value is calculated as $\frac{v_{heuristic} - v_{model}}{v_{model}} * 100$.

As the figures provided in Table II indicate, it is not practical to use F_L to obtain a solution to the problem, even for small sized instances such as those considered in Table II. This can be seen by observing that the computational time required to solve F_L to optimality increases heavily as the problems grow in size. On the other hand, the greedy heuristic seems to perform fairly well. We do not report on the corresponding time required by the heuristic algorithm, since this is below 1 second for all the instances considered in Table II). Furthermore, based on the data given in the last column of this table, we can state that the heuristic is able to produce fairly good solutions in very short computational times.

VII. CONCLUSIONS

Timely dissemination of dynamic multimedia objects in Networked Virtual Environments to all clients who might be interested in these object is a challenge when the upstream bandwidth of the clients is limited and the cost and delay associated with distribution is important. In this paper, we have developed an integer programming model that can be used on short time intervals for the purpose of optimally locating proxy servers, identifying the replication pattern of objects among the servers and assigning clients to the proxies so as to minimize the total transfer cost of the content. Only when the required object is not found in the proxy, a client will be forced to fetch it from the originating client. Our formulation includes a constraint on the upstream capacity of the clients to reflect the realistic scarcity of access bandwidth.

Since the proposed model includes a quadratic objective function and constraints, we have made use of a linearization technique to convert the formulation to a linear model. This is solved as a benchmark to test the performance of our heuristic algorithm. It is shown that our heuristic algorithm produces results which are close to optimal based on a set of random problems. The efficiency and effectiveness of the algorithm makes it suitable to run online and produce near-optimal solution for short time intervals of interest.

APPENDIX I

PROOF OF THE COMPLEXITY OF THE PROBLEM

Proposition 2: The problem F is \mathcal{NP} -Hard.

Proof: We prove the proposition by restriction (see [12]), where the following instance of the problem is considered. Let $K = \{1\}$ (i.e. there is only a single object), $b_1 = b$, $d_{i1} = d_i$ and $s_p \geq b$, $\forall p \in P$ (i.e. all the proxies have a sufficiently large capacity). Since there are no capacity constraints in this case, constraints (4) and (5) become redundant and $z_{p1} = 1$, $\forall p \in P$. In this case, the resulting problem becomes the Uncapacitated Facility Location Problem (FLP), which is known to be \mathcal{NP} -Hard (see e.g. [13]). ■

APPENDIX II

PROOF OF PROPOSITION 1

Proof: After simplifying the objective function of F as $\sum_{i \in I} \sum_{p \in P} \sum_{k \in K} (b_k d_{ik} c_{i,j(k)} x_{ij} - b_k d_{ik} (c_{ip} - c_{i,j(k)}) \varphi_{ipk})$, where $\varphi_{ipk} = z_{pk} x_{ip}$, the proof relies on the observation that the coefficient of φ_{ipk} in the objective function is $-b_k d_{ik} (c_{ip} - c_{i,j(k)})$, which is always negative. By definition, φ_{ipk} should be 1 if and only if $z_{pk} = 1$ and $x_{ip} = 1$, and 0 for all other cases. Now, assume that $z_{pk} = 1$ and $x_{ip} = 1$ for a specific (i, p, k) triplet. Then, according to constraints (9) and (10), φ_{ipk} is only constrained by the upper bound 1 and the minimizing objective function implies $\varphi_{ipk} = 1$. In all other cases (i.e. $x_{ip} = 1$, $z_{pk} = 0$; or $x_{ip} = 0$, $z_{pk} = 1$; or $x_{ip} = 0$, $z_{pk} = 0$) constraints (9) and (10) together imply $\varphi_{ipk} = 0$. ■

REFERENCES

- [1] D. Intelligence, "The Online Game Market 2004," *California, USA*, August 2004.
- [2] S. F. B. P. Brun, J., "Distributing Network Game Servers for Improved Geographical Scalability," *Telecommunications Journal of Australia*, vol. 55, pp. 23–32, March 2005.
- [3] S. F. D. M. Boustead, P., "DICE: Internet Delivery of Immersive Voice Communication to Crowded Virtual Spaces," *Proceedings of IEEE International Conference on Virtual Reality, VR 2005, Bonn, Germany*, March 2005.
- [4] S. F. Dowlatshahi, M., "A Recursive Overlay Multicast Algorithm for Distribution of Audio Streams in Networked Games," *Proceedings of the IEEE International Conference on Networks (ICON 04), Singapore*, November 2004.
- [5] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 568–582, 2000.
- [6] M. Yang and Z. Fei, "A model for replica placement in content distribution networks for multimedia applications," in *Proceedings of IEEE International Conference on Communications (ICC '03)*, vol. 1, 2003, pp. 557–561.
- [7] A. Datta, K. Dutta, H. Thomas, and D. VanderMeer, "World Wide Wait: a study of Internet scalability and cache-based approaches to alleviate it," *Management Science*, vol. 49, no. 10, pp. 1425–1444, October 2003.

- [8] T. Bektas, O. Oguz, and I. Ouveysi, "Designing cost-effective content distribution networks," *Computers and Operations Research*, 2005, in press.
- [9] Z. Xuanping, W. Weidong, T. Xiaopeng, and Z. Yonghu, *Data Replication at Web Proxies in Content Distribution Network*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2003, vol. 2642, pp. 560–569.
- [10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proceedings of IEEE INFOCOM'99*, vol. 1, New York, March 1999, pp. 126–134.
- [11] J. Xu, B. Li, and D. Lee, "Placement problems for transparent data replication proxy services," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1383–1398, 2002.
- [12] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, California: W. H. Freeman and Company, 1979.
- [13] G. Cornuejols, G. Nemhauser, and L. Wolsey, "The uncapacitated facility location problem," in *Discrete Location Theory*, P. Mirchandani and R. Francis, Eds. John Wiley & Sons, 1990, ch. 3, pp. 119–171.