

2009

On The Study of Establishing a Responsive Infrastructure for a Massively Multiplayer On-Line Game

Robert Aboolian

California State University San Marcos, raboolia@csusm.edu

Yi Sun

California State University San Marcos, ysun@csusm.edu

Jack Leu

California State University San Marcos, leu@csusm.edu

Follow this and additional works at: <http://aisel.aisnet.org/amcis2009>

Recommended Citation

Aboolian, Robert; Sun, Yi; and Leu, Jack, "On The Study of Establishing a Responsive Infrastructure for a Massively Multiplayer On-Line Game" (2009). *AMCIS 2009 Proceedings*. 762.

<http://aisel.aisnet.org/amcis2009/762>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

On The Study of Establishing a Responsive Infrastructure for a Massively Multiplayer On-Line Game

Robert Aboolian, Yi Sun, and Jack Leu

Department of Department of Information Systems & Operations Management, California State University San Marcos, San Marcos, CA 92096-0001, USA {raboolia@csusm.edu, ysun@csusm.edu, Leu@csusm.edu}

A massively multiplayer online game (MMOG) often requires a game publisher to deploy dozens or hundreds of n -tiered servers to support millions of concurrent players around the world. Planning such a massive network infrastructure, particularly in an environment where uncertain demand and limited server capacity could cause congestions in a host site and the network, poses a great challenge. A slow response time stemming from an ill-designed infrastructure could render an otherwise technically superior MMOG noncompetitive in the marketplace. In this study, we focus on three critical issues related to establishing an MMOG server infrastructure: selecting host facilities on a broadband provider's backbone network nodes, assigning client clusters represented by the Point of Presences (PoPs) to these MMOG facilities, and determining the required capacity for each host site. The problem is first formulated as a non-linear integer program based on an M/M/1 queuing system in each host facility. We then develop an exact solution approach obtained from solving a minimum cost set-covering problem. The efficiency of the solution approach is also reported.

Key words: Online Game, Congested facility location models, Non-linear integer program, Set-covering problem.

1. Introduction

Massively multiplayer online games (MMOGs) have become one of the most vibrant sectors in the video game industry because of their appeal to the younger generation. MMOGs refer to genres of online role-play videogames in which gamers can freely create or assume a character in a persistent and dynamic virtual community. The global market for these games was estimated to be \$2.7 billion in revenue in 2006 (Staehlin, 2003), and a successful game often serves a large group of players with a major economic stake. For example, it was estimated that World of Warcraft, one of the most popular MMOGs, had 5.5 million users and a revenue of \$300 million in 2005 (Helm, 2006). In order to support millions of players around the world, an MMOG publisher needs to create a massive client-server infrastructure with dozens to hundreds of copies of the application deployed globally.

In addition to game contents, the success of an MMOG also hinges on its playability, often measured by server throughput and network response time. Throughput is largely dictated by the capacity of game servers. MMOGs typically employ an n -tiered server architecture, with the front-tier managing security and load balance, the mid-tier handling game simulations, and the database tier keeping track of information about game objects and maneuvers (Dolbier, 2007a 2007b, 2007c; Van der Steen, 1997). To determine the server capacity for each tier, a game distributor must be able to estimate the number of concurrent players per geography (Dolbier, 2007a). This implies that the service zone of a server must be either known a priori or determined concurrently with server capacities. Network response time, on the other hand, largely depends on the distance between a player and the server (Johansson, 2000). While it is difficult to boost the propagation speed of network signals, an MMOG publisher can strategically locate game servers with adequate service capacity on a network to maintain a certain level of service quality.

To alleviate the last-mile bandwidth constraint, it is highly recommended that an MMOG server be hosted within a broadband provider's facility or in the close proximity (Megler, 2004). Thus, one of MMOG key research questions is how to strategically locate game servers with appropriate capacities on broadband network nodes so that the game distributor's cost can be minimized while meeting the service quality requirement. In this paper, the problem is first formulated as a non-linear integer program based on an $M/M/1$ queuing system in each host facility. We then develop an exact solution approach obtained from solving a minimum cost set-covering problem. We believe that we are among the first to study the optimal service design for MMOGs. Although the model and the algorithm are

developed specifically for MMOGs' service design problem, we expect them to be applicable, with modifications, to many applications with similar structures.

The plan for the paper is as follows. A literature review is provided in the next section. In Section 3, we introduce notation and formulations for the MMOG deployment problem. In Section 4, we develop an exact solution approach, which involves solving a minimum cost set-covering problem. Results for computational experiments and sensitivity analyses are presented in Sections 5 and 6, respectively. Finally, the strengths, the limitations, and future extensions of this study are discussed in the Section 7.

2. Literature Review

Deploying an MMOG involves significant economic tradeoffs in terms of costs associated with opening and operating server facilities and maintaining a certain level of service quality. For an action-packed MMOG, game access time, defined as the time from a client machine sends out a game request till it receives a response from the server, is regarded as the foremost important quality measure as it correlates strongly with user satisfactions (Armitage, 2001; Dick, Wellnitz, & Wolf, 2005; Henderson 2001; Henderson & Bhatti, 2002; Henderson, 2002). Game access time has two major components, network response time and server response time, which have been at the center of MMOG deployment consideration (Dolbier, 2007a, 2007b, 2007c).

In fact, the study of network response time dated back to the research in distributed database systems. Johansson (2000) examined the makeup of network response time and concluded that only network latency, defined as the time needed to propagate a signal between the sending and receiving nodes once the signal has been sent onto the network, would become the limiting factor. On the contrary, other factors such as the time needed to load information to the medium and the delay due to network access contentions were of immaterial in a high-speed networking environment. His study further showed that ignoring network latency could underestimate the response time by more than 80 percent in some case. In this study, we follow this research result and use network latency to measure network response time.

The conventional wisdom believes that network latency depends not only on the distance between the sending and receiving nodes but also on the protocols and topologies. However, physical distance has been shown to be the most relevant measurement for latency in recent studies. For example, Huffaker et al. (2002) examined the correlation between

latency and four popular Internet distance metrics: IP path length, autonomous system path length, great circle geographic distance, and round trip time. They concluded that metrics based on physical (geographic) characteristics correlated better with latency than those based on logical topologies. This finding was also supported by the research on the geographic distribution of online game servers and players (Dick, Wellnitz, & Wolf, 2005; Feng & Feng, 2003). Based on these results, this study uses distance to approximate network latency and server locations to control the amount of network induced game latency.

The second component of game access time is server response time, which includes the time waiting for accessing servers (queuing time) and being served by a server (processing time). Queuing time has been incorporated into many service system design problems employing queuing models to determine the appropriate server capacity so as to keep waiting time or service quality at an acceptable level (Berman & Drezner, 2002; Marianov & Serra, 1998; Wang, Batta, & Rump, 2002). However, we submit that using queuing time as a surrogate measurement for service quality is too limiting and does not reflect the entire delay experienced by an MMOG player. Therefore, this study suggests the more encompassing game access time, which is defined as the sum of network latency and server response time, be used to measure the service quality.

There are two popular types of MMOG architecture: *the zoned architecture*, in which a server manages the game state for the players in its dedicated zone, and *the seamless architecture*, in which all servers collaborate such that each server manages only a small piece of the game world (Van der Steen, 1997). In this study, we consider only the zoned MMOG, in which a server cannot alleviate congestions by redirecting service requests to a proxy server because the information about a user's game state is captive to the zone. Therefore, the problem for this study is to determine the location and the capacity of each game server as well as to assign clients to the servers, so as to balance the cost of opening and operating game facilities while keeping the service quality (measured by game access time) at a certain level. We call this the MMOG deployment problem hereafter.

While not much research has been devoted to the MMOG deployment problem, there is a rich body of Operations Research literature dedicated to the design of immobile service facilities. For example, Aboolian et al (2008a), Berman & Drezner (2007), and Wang et al. (2002) took a customer's perspective and focused on minimizing the total travel and waiting cost; Wang et al. (2002) and Marianov and Serra (1998) addressed the need of service providers with an emphasis on minimizing the total facility cost while holding a certain level of service quality; and Aboolian et al. (2008b, Amiri, 1997, Castillo et al. (2002), and Elhedhli

(2006) held a more balanced perspective known as the Socially Optimal Service System Design and tackled the cost of service capacity and the quality of services simultaneously. In this paper, we also approach the MMOG deployment problem from a provider's perspective. These problems are commonly modeled as a nonlinear MIP problem. However, we are able to reduce the MMOG nonlinear MIP problem to a tractable set-covering IP problem due to the unique definition of service quality.

3. Model Formulation

Let $M = \{1, 2, \dots, m\}$ be the set of m candidate host facility locations. We assume that the demand for service is concentrated at n Point of Presences (PoPs) or demand nodes $N = \{1, 2, \dots, n\}$, with node i generating an independent Poisson stream of service requests at a mean arrival rate for service request of λ_i per unit of time. Poisson arrivals are commonly used in modeling the performance of traditional Web applications and online games (Ye & Cheng, 2006). We will use $S \subset M$ to denote the set of facilities selected as the host sites.

We assume that each MMOG facility hosts a single server with a scalable capacity. While a server with a higher capacity may allow several physical Ethernet interfaces, these interfaces are typically aggregated into one virtual interface through a process known as *Channel Bonding*. Therefore, without the loss of generosity, a scaled-up server could be considered as a single server with an improved service rate. Define μ_j to be the service capacity at facility $j \in M$. In other words, facility $j \in M$ is assumed to serve the requests at a mean rate $\mu_j > 0$. Note that μ_j here is a decision variable, which can also be regarded as the mean service rate with which a service request is fulfilled. Also note that $1/\mu_j$ is the average processing time for a service request at facility j .

Define γ_j to be the mean arrival rate of service requests for the facility located at $j \in M$. Also, define H_j to be the set of all customer nodes served by the facility located at site j . Then, $\gamma_j = \sum_{i \in H_j} \lambda_i$. Assuming an exponential probability distribution for the service time, an MMOG host facility at $j \in M$ can be modeled as an $M/M/1$ queuing system with service rate μ_j and arrival rate γ_j . Define $w_j(\gamma_j, \mu_j)$ to be the average response time, defined as the time from a data packet arriving at a facility till a return packet ready to be sent, which includes queuing delay and processing time. In other words, $w_j(\gamma_j, \mu_j)$

represents how quickly a server can respond to a game request and can be calculated as follows:

$$w_j(\gamma_j, \mu_j) = \frac{1}{\mu_j - \gamma_j} \quad \forall j \in S. \quad (1)$$

Let t_{ij} be the network latency from an MMOG host facility located at $j \in S$ to clients located at $i \in N$ and define the average access time to be the average time a client machine takes to receive a game response from the server. Given the above definitions, $t_{ij} + w_j(\gamma_j, \mu_j)$ for $j \in S$ becomes the average game access time for clients located at $i \in H_j$.

To maintain a certain service satisfaction level, we assume that each host facility needs to ensure that the average game access time does not exceed a certain amount, denoted as φ ; therefore,

$$t_{ij} + w_j(\gamma_j, \mu_j) \leq \varphi \quad \forall j \in S, i \in H_j. \quad (2)$$

As mentioned before, μ_j is a decision variable representing the server capacity in facility located at $j \in M$. Let x_j be a binary decision variable, which will take a value of one if the decision is to open an MMOG host facility at candidate site $j \in M$ and zero otherwise. Define f_j to be the installation cost (e.g., infrastructure cost) for opening a host facility at $j \in M$, and c to be the cost for each unit of server capacity. We assume that the game publisher adopts a type of shared-memory MIMD (Multiple Instruction stream, Multiple Data stream) machines, which allows more CPUs to be added as needed. We further assume that these independent CPUs are connected through a bus network; therefore, the cost for each added CPU unit can be considered identical (Van der Steen, 1997).

In this paper, each customer is assumed to be served by a single facility. Let y_{ij} be a binary decision variable that takes the value of one if customers at $i \in N$ are to be served by the facility located at $j \in M$ and zero otherwise. Then, γ_j , the arrival rate for the server at $j \in M$, can be obtained by

$$\gamma_j = \sum_{i \in N} \lambda_i y_{ij} \quad \forall j \in M. \quad (3)$$

With the definitions and the discussions provided thus far, the MMOG infrastructure problem can be formulated as the following optimization model:

$$\min \sum_{j \in M} f_j x_j + c \sum_{j \in M} \mu_j \quad (4) \quad \text{Problem } P1$$

S.t.

$$y_{ij} \leq x_j \quad \forall i \in N, j \in M, \quad (5-1)$$

$$\sum_{j \in M} y_{ij} = 1 \quad \forall i \in N, \quad (5-2)$$

$$\mu_j \geq \sum_{k \in N} \lambda_k y_{kj} + \varepsilon x_j \quad \forall j \in M, \quad (5-3)$$

$$(t_{ij} + w_j(\gamma_j, \mu_j)) y_{ij} \leq \varphi \quad \forall i \in N, j \in M, \quad (5-4)$$

$$w_j(\gamma_j, \mu_j) = \frac{x_j}{\mu_j - \sum_{i \in N} \lambda_i y_{ij} + 1 - x_j} \quad \forall j \in M, \quad (5-5)$$

$$\mu_j \geq 0 \quad \forall j \in M, \quad x_j \in \{0,1\} \quad \forall j \in M, \quad y_{ij} \in \{0,1\} \quad \forall i \in N, j \in M.$$

Equation (4), the objective function, minimizes the total fixed facility and variable capacity cost. Constraints (5-1) assure that if a facility at a given location is not opened ($x_j = 0$) then no customer is allocated to it ($y_{ij} = 0$). Constraints (5-2) guarantee that each client on the network will be served by one and only one MMOG host facility. Constraints (5-3) prevent an unlimited response time (here $\varepsilon = 10^{-6}$ clients per unit of time). Constraints (5-4) affirm that the average game access time in each facility will not exceed a certain threshold. Constraints (5-5) make sure that the average time to service completion in each host facility will equal to $\frac{1}{\mu_j - \gamma_j}$ if the host facility is opened ($x_j = 1$) and will equal to zero otherwise (note that when $x_j = 0$, $\mu_j = \sum_{i \in N} \lambda_i y_{ij} = 0$ as well because of the objective function and the constraints in (5-1)). This is a nonlinear integer program, which generally is hard to solve.

In the next section, we develop a solution approach to solve Problem *P1* optimally.

4. Solution Approach for Problem *P1*

Before we present the exact solution methodology for Problem *P1*, consider the following result.

Lemma 1: For $j \in S$, define $\hat{t}_j = \max_{i \in H_j} \{t_{ij}\}$ to be the maximum network latency from

facility $j \in S$ to client nodes in H_j . Also, denote $e_j = [\varphi - \hat{t}_j]^{-1}$ and define $\Lambda = \sum_{i \in N} \lambda_i$ to

be the total arrival rate on the network. Then

- a) The server response time (short for the response time hereafter) at facility $j \in S$, $w_j(\gamma_j, \mu_j) = \varphi - \hat{t}_j$, and e_j can be defined to be the mean rate for service completion (including delay and processing times) at facility $j \in S$ and can be expressed as $e_j = [w_j(\gamma_j, \mu_j)]^{-1}$; and
- b) The total fixed facility and variable capacity cost $\sum_{j \in S} f_j + c \sum_{j \in S} \mu_j$ can be rewritten as $\sum_{j \in S} f_j + c \sum_{j \in S} e_j + c \Lambda$.

Proof:

(a) From (2), for $j \in S$, we have $w_j(\gamma_j, \mu_j) \leq \varphi - t_{ij} \quad \forall i \in H_j$. Therefore, $w_j(\gamma_j, \mu_j)$

$$= \min_{i \in H_j} \{\varphi - t_{ij}\} = \varphi - \max_{i \in H_j} \{t_{ij}\} = \varphi - \hat{t}_j. \text{ Thus, } e_j = [\varphi - \hat{t}_j]^{-1} = [w_j(\gamma_j, \mu_j)]^{-1}.$$

(b) From (1) and the result of part (a), we conclude $e_j = \mu_j - \gamma_j$ or $\mu_j = \gamma_j + e_j$ for

$$j \in S. \text{ Then } \sum_{j \in S} \mu_j = \sum_{j \in S} \gamma_j + \sum_{j \in S} e_j = \sum_{j \in S} \sum_{i \in H_j} \lambda_i + \sum_{j \in S} e_j = \Lambda + \sum_{j \in S} e_j.$$

Therefore, $\sum_{j \in S} f_j + c \sum_{j \in S} \mu_j = \sum_{j \in S} f_j + c \sum_{j \in S} e_j + c \Lambda$, which concludes the proof.

Lemma 1 shows that the total cost can be rewritten as the function of fixed facility and variable response time (instead of capacity) costs and that minimizing this function will automatically minimize the total fixed facility and variable capacity cost. This also means that once the response times in all facilities are decided, how the clients are assigned will not affect the objective function provided that the assignment scheme does not violate the established response time at each facility.

Given the above argument, we will provide a new formulation for selecting facility locations and establishing the response time for each of those selected facilities. Then, with the optimal solution to this new problem, we will find a feasible client assignment and determine the server capacity for each facility accordingly.

Define $N_j = \{i \mid t_{ij} < \varphi\}$ to be all the client nodes with a network latency to facility $j \in S$ lower than φ . Define $z_{ij} \in N_j$ to be a binary decision variable, which takes a value of one if the maximum response time at MMOG host facility $j \in M$ equals $\varphi - t_{ij}$, and a value of zero otherwise. For the simplicity of presentation and the correctness of the definition of z_{ij} , we assume that no two client nodes will have the same network latency for

accessing facility $j \in M$. This assumption is realistic given that latency is measured by network distance whose representation accuracy can always be increased for the discriminating purpose. Since $\sum_{i \in N_j} z_{ij}$ equals one if a facility is located at $j \in M$ and equals zero otherwise, we have:

$$\sum_{i \in N_j} z_{ij} \leq 1. \quad (6)$$

Given Lemma 1, the total fixed facility and variable capacity cost can be rewritten as $\sum_{j \in M} \sum_{i \in N_j} f_j z_{ij} + c \sum_{j \in M} \sum_{i \in N_j} [\varphi - t_{ij}]^{-1} z_{ij} + c\Lambda$. If we denote $a_{ij} = f_j + c[\varphi - t_{ij}]^{-1}$, the objective function can be expressed as $\sum_{j \in M} \sum_{i \in N_j} a_{ij} z_{ij} + c\Lambda$. (7)

Now, consider the following definition and results regarding the coverage conditions for a client node.

Definition 1 (Cover): the MMOG host facility located at site $j \in S$ is said to cover (can provide services to) clients located at $i \in N$ if

$$t_{ij} + w_j(\gamma_j, \mu_j) \leq \varphi. \quad (8)$$

Lemma 2: For $j \in M$ and $i \in N$, define $K_j^i = \{k \mid t_{ij} \leq t_{kj}, k \in N_j\}$ to be all the client nodes with a network latency to facility $j \in S$ lower than φ but higher than or equal to that of node i . Then,

- a) If $\sum_{k \in K_j^i} z_{kj} = 1$, then facility j covers clients at node i ; and
- b) The coverage condition for clients at node $i \in N$ is $\sum_{j \in M} \sum_{k \in N - K_j^i} z_{kj} \geq 1$.

Proof:

(a) Since $\sum_{k \in K_j^i} z_{kj} = 1$, then $\exists k \in K_j^i$ such that $z_{kj} = 1$. Now, by the definition of z_{kj} and K_j^i , we have $w_j(\gamma_j, \mu_j) = \varphi - t_{kj} \leq \varphi - t_{ij}$ or $t_{ij} + w_j(\gamma_j, \mu_j) \leq \varphi$. Therefore, we can conclude, by Definition 1, that facility j covers clients at node i .

(b) Follows directly from the result in part (a).

With the above definitions and results, the new problem can be formulated as the following optimization model:

$$\min \sum_{j \in M} \sum_{i \in N_j} a_{ij} z_{ij} + c\Lambda \quad (9) \quad \text{Problem P2}$$

s.t.

$$\sum_{j \in M} \sum_{k \in K_j^i} z_{kj} \geq 1 \quad \forall i \in N, \quad (10-1)$$

$$\sum_{i \in N_j} z_{ij} \leq 1 \quad \forall j \in M, \quad (10-2)$$

$$z_{ij} \in \{0,1\} \quad \forall j \in M, i \in N_j. \quad (10-3)$$

It is easy to verify that objective function (9) and constraint (10-1) ensure that (10-2) will always hold; therefore, (10-2) becomes redundant. To prove this for any $j \in M$, assume that two distinct client nodes $p, q \in N_j$ such that $z_{pj} = z_{qj} = 1$ ($\sum_{i \in N_j} z_{ij} > 1$) and $t_{pj} > t_{qj}$. Then, we can conclude that $K_j^p \subset K_j^q$, which means that a client node that is supposed to be covered by $z_{pj} = z_{qj} = 1$ at a cost of $a_{pj} + a_{qj}$ can be covered by $z_{pj} = 1$ at a lower cost of a_{pj} . Also, $c\Lambda$ in (9) has no effect on the solution of problem P2. Thus, problem P2 can be written as the following minimum cost set covering problem:

$$\begin{aligned} \min \sum_{j \in M} \sum_{i \in N_j} a_{ij} z_{ij} \quad & \text{Problem P2} \\ \text{s.t. (10-1) and (10-3).} \end{aligned}$$

Although set covering problems are NP hard, there are plenty of efficient solution approaches available in the OR literature.

After finding the optimal MMOG host facility locations and the response time in each facility through solving problem P2, we need to find a feasible allocation scheme to assign clients to these facilities without violating their respective response times.

Define z_{ij}^* to be the optimal solution to problem P2, $S^a = \left\{ j \mid \sum_{i \in N_j} z_{ij}^* = 1 \right\}$ to be the optimal set of sites to host MMOG facilities obtained from problem P2, and $S_i^a = \left\{ j \mid \sum_{k \in K_j^i} z_{kj}^* = 1 \right\}$ the set of optimal facilities covering client node i . Please note that constraints (10-1) ensure $S_i^a \neq \{ \}$ for all $i \in N$. To find a feasible client allocation, we can arbitrarily assign client node i to one of the facilities in S_i^a . Next, we show that this client allocation scheme would not violate the optimal response time at facility $j \in S^a$. Given the definition of S_i^a , K_j^i , and \hat{t}_j , if $j \in S_i^a$, then $t_{ij} \leq \hat{t}_j$. Therefore, $[\varphi - t_{ij}]^{-1} \leq [\varphi - \hat{t}_j]^{-1} = w_j(\gamma_j, \mu_j)$, which means that allocating client node $i \in N$ to any facility at $j \in S_i^a$ would not increase the response time of that facility. In order to have a

distinct allocation scheme, we propose that client node i be assigned to the closest facility in S_i^a for all $i \in N$. Define H_j^a to be all the client nodes allocated to facility $j \in S^a$; therefore, $H_j^a = \{i \mid t_{ij} \leq t_{ik} \ \forall k \in S^a, i \in N_j\}$.

After obtaining the feasible client allocation, we can determine the capacity required at each facility. More specifically, the required capacity at facility $j \in S^a$ can be expressed as

$$\mu_j^a = \sum_{i \in H_j^a} \lambda_i + \sum_{i \in H_j^a} \left[\varphi - \hat{t}_j \right]^{-1}. \quad (11)$$

For convenience, we define $C^a = \{\mu_j^a \mid j \in S^a\}$ to be the set of required capacities for all facilities in S^a .

Note that we may have different feasible client allocations to Problem P2, which in turn may result in a different capacity cost in some facilities, but, given Lemma 1, the overall capacity cost for any feasible allocations would always equal to $c \sum_{j \in S^a} \left[\varphi - \hat{t}_j \right]^{-1} + c\Lambda$.

To summarize the above arguments on how to find a solution for the original problem, we present the following algorithm:

Algorithm 1

Step 0: For $j \in M$, $i \in N_j$, set $a_{ij} = f_j + c \left[\varphi - t_{ij} \right]^{-1}$ and $K_j^i = \{k \mid t_{ij} \leq t_{kj}, k \in N_j\}$.

Step 1: Solve set-covering problem P2 and find $z_{ij}^* \ \forall j \in M, i \in N$.

Step 2: Find $S^a = \left\{ j \mid \sum_{i \in N_j} z_{ij}^* = 1 \right\}$, and $S_i^a = \left\{ j \mid \sum_{k \in K_j^i} z_{kj}^* = 1 \right\}$.

Step 3: For $j \in S^a$, find $H_j^a = \{i \mid t_{ij} \leq t_{ik} \ \forall k \in S^a, i \in N_j\}$.

Step 4: Find C^a , the required capacities for facilities in S^a , using (11).

Step 5: Set $\mathbf{Z}_{P1}^a = \sum_{j \in S^a} f_j + c \sum_{j \in S^a} \left[\varphi - \hat{t}_j \right]^{-1} + c \sum_{i \in N} \lambda_i$.

Step 6: Stop. S^a , C^a , H_j^a , and \mathbf{Z}_{P1}^a are the solutions to Algorithm 1.

Next, we prove that the solution for Algorithm 1 is an optimal solution for the original problem P1. The exactness of the Algorithm 1 is based on the following result.

Theorem 1: Define \mathbf{Z}_{P1}^* to be the optimal objective function value of problem P1. Also, define \mathbf{Z}_{P1}^a to be the objective value obtained by Algorithm 1. Then $\mathbf{Z}_{P1}^* = \mathbf{Z}_{P1}^a$.

Proof: Define S^* , and H_j^* to be an optimal set of facility locations and an optimal set of client allocations for the original problem, respectively. Also define

$i^j = \arg \max_{i \in H_j^*} \{t_{ij}\}$ and $z_{kj} = \begin{cases} 1 & \text{if } k = i^j, \text{ and } j \in S^* \\ 0 & \text{otherwise} \end{cases}$. By definition, for every

$i \in H_j^*$, we have $i^j \in K_j^i$ and $\sum_{k \in K_j^i} z_{kj} = 1$. Therefore, (10-1) in problem $P2$ holds

for S^* , H_j^* . In other words, the optimal solution for problem $P1$ is a feasible solution for problem $P2$. According to Lemma 1, we have

$\mathbf{Z}_{P1}^* = \sum_{j \in S^*} f_j + c \sum_{j \in S^*} [\varphi - \hat{t}_j]^{-1} + c \sum_{i \in N} \lambda_i$. Now, by the definition of \mathbf{Z}_{P1}^a in

Algorithm 1 and the optimality conditions in problem $P2$, we have

$\sum_{j \in S^*} f_j + c \sum_{j \in S^*} [\varphi - \hat{t}_j]^{-1} \leq \sum_{j \in S^a} f_j + c \sum_{j \in S^a} [\varphi - \hat{t}_j]^{-1}$. Thus, $\mathbf{Z}_{P1}^* = \mathbf{Z}_{P1}^a$ and the proof is complete.

In the next two sections, we conducted a series of experiments to evaluate the efficiency of the exact solution approach presented here and examine its behavior with respect to changes in parameters.

5. Experiment and Results

We conducted a computational experiment to assess the efficiency of the proposed solution approach (Algorithm 1). The algorithm was coded in C++, with the exception of Step 1, in which the CPLEX IP Solver Version 10.0 was invoked to solve Problem $P2$. The program was run on an Intel 2.0 GHz computer with 2 GB RAM using a set of simulated cases generated according to the settings of the following three main factors:

- I. The number of candidate host facility locations (M) is set at four levels: $M=25, 50, 75$, and 100.
- II. The number of demand nodes (N) is set at four levels: $N=100, 150, 200$, and 250.
- III. The maximum game access time, φ , is set at three levels: low (15), medium (30), and high (45).

A pilot study was conducted first to help determine the levels of the first two factors so that the optimal solutions could be obtained within a reasonable amount of time. The three levels of the maximum game access time were chosen based on the result of some studies showing that even a delay of 50 ms – 75 ms could become noticeable (Beigbeder, Coughlan, Lusher, Plunkett, Agu, & Claypool, 2004, Dick, Wellnitz, & Wolf, 2005).

We also set other parameters in the following fashion and deferred the investigation of their impact to the next section devoted to sensitivity analyses.

- Network latency, t_{ij} , was randomly generated from a uniform distribution on (0, 600). The upper bound of the interval was a rough estimate of the latency halfway across the globe on a frame relay based network during the peak usage period.
- Service request arrival rate $\lambda_i \forall i \in N$ was randomly generated from a uniform distribution on [1,000, 10,000]. We assumed that a server could support up to 600 concurrent users (Dolbier, 2007a, 2007b, 2007c; Smed, Kaukoranta, & Hakonen, 2001) and that it was desirable to keep the maximum game access time at 60 ms. Hence, we set the upper bound to 10,000 service requests per second.
- Unit server cost was set to \$1.00 per request annually. We estimated that a server costs range from \$5,000 to \$10,000 per year. With a maximum of 10,000 service requests per second, the annualized unit server cost for one request per second would be between \$.50 and \$1.00. We, however, fixed the unit server cost at \$1 for this experiment and then investigated the impact of its variations later because the cost of a server should be able to be estimated rather accurately.
- Facility fixed cost $f_j \forall j \in M$ was randomly generated from a uniform distribution on [25,000, 100,000]. The interval of facility fixed cost was chosen to suggest a diverse range of facility costs among candidate facility locations.

This experiment represented a $4 \times 4 \times 3$ factorial design. Each experiment combination was replicated 10 times for a total 480 test cases. Our objective in this experiment was to measure how the three main factors affect the computational speed of Algorithm 1, the number of selected facility locations, the overall cost, and the client's expected latency. An analysis of variance (ANOVA) was carried out for each performance measurement to identify significant main and interaction effects.

Table 2 showed the average CPU times for each combination of ten test cases. The average CPU times required ranged from a fraction of a second for smaller test cases to nearly half an hour for the largest case. It is easy to understand the rise in computational times with respect to the increase in the number of candidate facility locations (M) and the number of demand nodes (N). However, the impact of maximum game access time, ϕ , is much more profound and warrants a further investigation.

The ANOVA result in Table 3a showed that all main and interaction effects were statistically significant. It also revealed that, among all significant effects, ϕ had the

strongest explanatory power (had the largest mean square errors and F-value) in accounting for the variations in CPU times. As shown in Tables 3b-3d, a similar conclusion about the effect of φ could be applied to the other three performance measures. In Figure 1, we further explored how different levels of maximum game access time affect Algorithm 1's computational speed. More specifically, we devised a statistic called the CPU ratio defined as $\text{CPU ratio} = \text{CPU}_{\text{factor level}} / \text{CPU}_{\text{base case factor level}}$, where $\varphi = 30$ is the base case for every M and N combination. Figure 1 revealed that when φ was set at 15 ms, the gain in computation speed was less than 30%. However, the computational time for φ at 45 ms skyrocketed to an average of 155 times higher than that for φ at the base level. The exponential increase in computational time could be largely attributed to the rapid increase in the number of binary variables required to solve the set-covering problem in Step 1.

In addition to the computational speed, φ also affected many aspects of the MMOG deployment. To illustrate this, we used statistics similar to that used in Figure 1 in that the performance measure at $\varphi = 30$ was used as a base level for performance comparisons. Figure 2, which showed the relationship between the different levels of φ and the number of locations selected, revealed that increasing the maximum game access time would result in fewer server locations. This was because a higher level of φ would allow servers to have slower service rates and/or permit a game request to travel a longer distance to reach its designated server. In either case, φ would have an impact on the degree of network congestions. In addition, given the assumption of a constant server cost per request in this experiment, changes in the number of locations would affect the total fixed facility cost, and, therefore, the overall cost as shown in Figure 3. As depicted in Figure 4, another consequence of varying φ was that a longer maximum game access time would result in a longer expected latency for the clients, thus a lower service quality. These experiment results suggested an important managerial implication. That is, the proposed approach allows the management to strike a balance between the infrastructure cost and the quality of service through adjusting the maximum game access time.

6. Sensitivity Studies and Results

The proposed model in (9) through (10-3) has a few parameters that might be critical to its performance. In the last experiment, we investigated the effects of parameters that mainly change the number of constraints and the number of variables of the set-covering

problem. In this section, we conducted three sensitivity analyses, each of which focused on the effect of one of the parameters in the objective function: service request arrival rate (λ_i), fixed facility cost (f_j), and annualized unit server cost (c). Our objective was twofold: (1) to validate the findings in Experiment I and (2) to offer additional insights into the pros and the cons of the proposed model.

Unlike in Experiment I where λ_i and f_j were assumed to be uniformly distributed and c was fixed at \$1, in the sensitivity studies, they were set to following three levels:

- $\lambda_i = 1,000, 5,000, \text{ and } 9,000$;
- $f_j = 25,000, 50,000, \text{ and } 75,000$;
- $c = \$0.5, \$1.0, \text{ and } \$1.5$.

Since the effects of M , N , and φ were known through the previous experiment, we generated only a subset of test cases used in Experiment I based on the following settings:

- $M = 75 \text{ and } 100$;
- $N = 100, 150, 200, \text{ and } 250$;
- $\varphi = 30$;

Therefore, each sensitivity study was a $2 \times 4 \times 1 \times 3$ factorial design. We also replicated each experiment combination 10 times for a total of 240 experiment runs per analysis. Other parameter settings unless aforementioned were kept the same as those in Experiment I. However, performance evaluations were only based on the ratios of computational speed, the number of locations selected, expected latency, and overall cost to cancel out effects due to confounding factors so that any performance differences could be attributed solely to the intended parameter changes. The results were shown in Figures 5-8 where parameter setting level 2 was always used as a base level for calculating the ratios, and the following conclusions could be made:

- We could infer from Figure 5 that the differences in computation speed due to changes in λ_i , f_j , and c were either nil or not statistically significant. This is because the total arrival rate was only a constant in the objective function and changes in f_j and c affected only the search path not the solution space.
- In the absence of budget and capacity constraints, the number of selected facility locations was not at all affected by the changes in the objective parameters. Instead, any changes in these parameters were only reflected in the overall cost. This could be

verified by examining Figures 6 and 7. Not including these additional constraints, however, is not a weakness of the proposed model. First, these constraints would drastically increase the complexity of the set-covering problem and would possibly render it intractable even for a mid-sized MMOG deployment problem. Second, another difficulty for including budget and capacity constraints in addition to the quality constraints in (5-4) is that all of them might have to be dealt with explicitly as these constraints make the conversion to the set-covering problem difficult, if not impossible. Rather, the proposed model affords a manager to balance cost and quality of service via the maximum access time parameter as discussed in Experiment I. In effect, the proposed model allows this complex problem to be decomposed into several set-covering problems with different maximum access times.

- As shown in Figure 8, the effect of changes in the objective parameters on client's expected latency was negligible. Because these parameters did not affect the location selections and the proposed algorithm always assigned a client to his/her nearest server, the negligible latency difference was due to the existence of alternative solutions in location selections. This revealed an important trait of the proposed model -- less accurate cost and demand estimates would not appreciatively affect the decision of server locations.

In all, the sensitivity studies not only affirmed the validity of the findings in Experiment I, but also revealed a few inherited advantages in the proposed model. In addition, these experiments showed that the proposed algorithm was capable of obtaining an optimal solution to a decent sized MMOG deployment problem and the solution should be able to withstand the test of empirical data because the model depended only on the assumption of exponential distribution of service time and Poisson stream for service request arrival rate.

7. Concluding Remarks and Future Research

The MMOG industry has become one of most vibrant e-commerce segments due to its appeal to the younger generation globally. As the competition intensifies, a game publisher must mitigate the adverse effect of network latency. In this study, we proposed a non-linear mathematical model for deploying an MMOG system on the Internet. The proposed model was subsequently converted to a set-covering problem, and an exact algorithm was developed. We also proved that the algorithm was able to obtain the optimal solution to the

original problem. An experiment was then carried out to evaluate the effectiveness of the algorithm based on four performance measurements. Important conclusions from the experiment included: (1) the algorithm was capable of solving a good sized problem within a reasonable amount time; and (2) maximum access time, which could directly affect the degree of network congestions, could be used for a manager to balance the infrastructure cost and the quality of service. The findings of the experiment were validated via three sensitivity analyses, which also shed lights on some interesting properties of the proposed model.

While we presented a novel approach to the MMOG deployment problem, many issues have yet to be addressed. First, this study did not consider deploying a game in a competitive environment, in which the importance of pricing, latency, and server capability would be heightened and a good model must have the provision for a manager to use them as a competitive weapon. Second, while the proposed algorithm was shown to be effective for a mid-sized problem, a heuristic algorithm must be developed in order to deal with a large-sized problem. Third, to maintain its tractability, the proposed model did not include constraints for budget, capacity, and so forth. The tradeoffs for the inclusion of such constraints should be examined. It also only considered the minimization of the cost. The tradeoffs for other objective functions and the inclusion of budgetary and capacitated constraints should be examined. Fourth, while our experiments showed that a manager could explore the setting of maximum access time to strike a balance between infrastructure cost and quality of service, it is possible to develop a profit maximization model for obtaining the optimal maximum access time in lieu of the brutal force approach suggested in this study. Fifth, this study focused only on the zoned MMOGs. An investigation into the deployment problem concerning the seamless MMOGs would enhance the contribution to the online game industry. Sixth, to study the efficacy of the heuristics and the characteristics of the model, we used simulated data. Despite our best effort to generate reasonable and representative data, we acknowledge that the study might benefit from using empirical data. Lastly, the proposed model and algorithm form a general optimization methodology. We will explore their applicability to the design of other service systems.

References

- Aboolian, R., O. Berman, Z. Drezner. 2008. The Multiple Server Center Location Problem. *Annals of Operations Research*. DOI: 10.1007/s10479-008-0341-2, in press.
- Aboolian, R., O. Berman Z. Drezner. 2008. Location-Allocation of Service Units on a Congested Network, *IIE Transactions* **40** 1-12.
- Amiri, A. 1997. Solution procedures for the service system design problem, *Computers & Operations Research* **24** 49–60.
- Armitage, G. 2001. Sensitivity of Quake3 Players to Network Latency. *Third International Workshop on Networked Group Communication*, London, UK.
- Beigbeder, T., R. Coughlan, C. Lusher, J. Plunkett, E. Agu, M. Claypool. 2004. The Effects of Loss and Latency on User Performance in Unreal Tournament 2003. *Workshop on Network and System Support for Games*, Portland, OR, USA.
- Berman, O., Z. Drezner. 2007. The Multiple Server Location Problem, *Journal of the Operational Research Society* **58** 91-99.
- Berman, O., Z. Drezner. 2002. Facility location problems with stochastic demands and congestion. H.W. Hamacher, Z. Drezner, eds. *Facility Location: Applications and Theory*, Springer-Verlag, New York, NY, USA.
- Castillo, I., A. Ingolfsson, T. Sim. 2002. Socially Optimal Location of Facilities with Fixed Servers, Stochastic Demand and Congestion. *Management Science* working paper 02-4, University of Alberta School of Business, Edmonton, Canada.
- Cheshire, S. 1996. Latency and the Quest for Interactivity. White paper commissioned by Volpe Welty Asset Management, L.L.C., for the Synchronous Person-to-Person Interactive Computing Environments Meeting.
- Dick, M., O. Wellnitz, L. Wolf. 2005. Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games. *Fourth Workshop on Network and System Support for Games*, Hawthorne, NY, USA.
- Dijkstra, E. 1959. *A Note on Two Problems in Connexion with Graphs*. *Numerische Mathematik* **1** 269-271.
- Dolbier, G. 2007a. Massively multiplayer online games, Part 1: A performance-based approach to sizing infrastructure. *IBM developerWorks*. <http://www.ibm.com/developerworks/web/library/wa-mmogame1>.

- Dolbier, G. 2007b. Massively multiplayer online games, Part 2: Understand the economic factors of hosting a game. *IBM developerWorks*. <http://www.ibm.com/developerworks/web/library/wa-mmogame2>.
- Dolbier, G. 2007c. Massively multiplayer online games, Part 3: Resolve potential issues with hosting MMOGs. *IBM developerWorks*. <http://www.ibm.com/developerworks/web/library/wa-mmogame3>.
- Feldman, E., F.A. Lehrer T.L. 1966. Ray, Warehouse Location under Continuous Economies of Scale. *Management Science* **12** 670-684.
- Elhedhli, S. 2006. Service System Design with Immobile Servers, Stochastic Demand, and Congestion. *Manufacturing & Services Operation Management* **8** 92-97.
- Feng, W., W. Feng. 2003. On the Geographic Distribution of On-line Game Servers and Players. *Second Workshop on Network and System Support for Games*, Redwood City, CA, USA.
- Garey, M.R., D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP Completeness*. W. H. Freeman and Company, New York, NY, USA.
- Helm, B. 2006. James Cameron's Game Theory. *Business Week*, February 13, 40.
- Henderson, T. 2001. Latency and User-behaviour on a Multiplayer Game Server. *Third International Workshop on Networked Group Communication*, London, UK.
- Henderson, T., S. Bhatti. 2001. Modeling User Behavior in Networked Games. *Ninth ACM International Conference on Multimedia*, Ottawa, Ontario, Canada.
- Henderson, T. 2002. Observations on Game Server Discovery Mechanisms. *First Workshop on Network and System Support for Games*, Braunschweig, Germany.
- Huffaker, B., M. Fomenkov, D. Plummer, D. Moore, K. claffy. 2002. Distance Metrics in the Internet. *IEEE International Telecommunications Symposium*, Natal, Brazil.
- Johansson, J.M. 2000. On the Impact of Network Latency on Distributed Systems Design. *Information Technology Management* **1** 183-194.
- Jouni, S., K. Timo, H. Harri. 2002. A Review on Networking and Multiplayer Computer Games, Technical Report 454, Turku Centre for Computer Science, Turku, Finland.
- Kuehn, A.A., M.J. Hamburger. 1963. A Heuristic Program for Locating Warehouses. *Management Science* **9** 643-666.
- MacKenzie, I.S., S. Ware. 1993. Lag as a Determinant of Human Performance in Interactive Systems. *ACM Conference on Human Factors in Computing Systems*, Amsterdam, The Netherlands.

- Marianov, V., D. Serra. 1998. Probabilistic maximal covering location–allocation for congested system. *Journal of Regional Science* **38** (3) 401–424.
- Megler, V. 2004. Online Game Infrastructures. *IBM developerWorks*. <http://www-128.ibm.com/developerworks/ibm/library/wa-games1>.
- Schäfer, C., T. Enderes, H. Ritter, M. Zitterbart. 2003. Subjective Quality Assessment for Multiplayer Real-Time Games. First workshop on Network and System Support for Games, Bruanschweig, Germany.
- Smed, J., T. Kaukoranta, H. Hakonen. 2001. Aspects of Networking in Multiplayer Computer Games. International Conference on Application and Development of Computer Games in the 21st Century, Hong Kong, China.
- Staehlin, C. 2003. Making MMOGs: Redefining ‘Product Development’. Austin Game Conference, Austin, TX, USA.
- Van der Steen, A.J. 1997. *Overview of Recent Supercomputers*. 7th Edition, NCF, The Hague. <http://www.phys.uu.nl/~steen/web00/overview00.html>.
- Wang, Q., R. Batta, C.M. Rump. 2002. Algorithms for a Facility Location Problem with Stochastic Customer Demand and Immobile Servers, Recent Developments in the Theory and Applications of Location Models Part II, *Annals of Operations Research* **111** 17-34. O. Berman and D. Krass (eds.), Kluwer Academic Publishers.
- Ye, M., L. Cheng. 2006. System-performance modeling for massively multiplayer online role-playing games. *Online Game Technology* **45**, 1. <http://www.research.ibm.com/journal/sj/451/ye.html>.

Table 1: Summary of Notation

Sets	
M	set of candidate facility locations $M = \{1, 2, \dots, m\}$,
N	set of demand points $N = \{1, 2, \dots, n\}$,
S	set of facilities selected as the host sites,
H_j	set of all customer nodes served by the facility located at site j ,
N_j	set of the client nodes with a latency to facility $j \in S$ lower than maximum game access time,
S^a	set of sites to host MMOG facilities obtained from Algorithm 1,
S_i^a	set of optimal facilities covering client node $i \in N$, obtained from Algorithm 1.
Parameters	
λ_i	mean arrival rate of service request per unit of time at demand point $i \in N$,
γ_j	mean arrival rate of service requests for the facility located at $j \in M$,
t_{ij}	network latency from the MMOG host facility $j \in S$ to demand node $i \in N$,
\hat{t}_j	maximum latency from facility $j \in S$ to client nodes in H_j ,
φ	maximum access time required for each host facility.
f_j	fixed installation cost for opening a host facility at site $j \in M$,
a_{ij}	coefficients of z_{ij} in the objective function in Problem P2, $a_{ij} = f_j + c[\varphi - t_{ij}]^{-1}$,
c	cost for each unit of server capacity.
Decision Variable	
x_j	a binary variable to indicate whether a new facility at $j \in M$ is opened,
μ_j	service capacity at facility $j \in M$,
z_{ij}	a binary variable to indicate whether the maximum response time at facility $j \in M$ equals $\varphi - t_{ij}$,
y_{ij}	a binary variable to indicate whether customers at $i \in N$ are served by the facility at $j \in M$.
Computed Values	
Z_{P1}^*	optimal objective function value of problem P1,
Z_{P1}^a	objective function value obtained from Algorithm 1,
Λ	total arrival rate on the network,
e_j	mean rate for service completion (including delay and processing times) at facility $j \in S$,
μ_j^a	capacity at facility $j \in S^a$ obtained from Algorithm 1,
$w_j(\gamma_j, \mu_j)$	average response time at MMOG host facility $j \in S$.

Table 2: Solution Speeds of Algorithm 1

M	N	φ	Average CPU Time (in seconds)	M	N	φ	Average CPU Time (in seconds)
25	100	15	0.0295	75	100	15	0.0438
25	100	30	0.0295	75	100	30	0.0674
25	100	45	0.0295	75	100	45	0.2988
25	150	15	0.0431	75	150	15	0.0610
25	150	30	0.0325	75	150	30	0.0781
25	150	45	0.0386	75	150	45	0.6038
25	200	15	0.0436	75	200	15	0.0780
25	200	30	0.0451	75	200	30	0.3457
25	200	45	0.0468	75	200	45	2.5235
25	250	15	0.0470	75	250	15	0.1015
25	250	30	0.0486	75	250	30	0.1189
25	250	45	0.0563	75	250	45	25.8682
50	100	15	0.0386	100	100	15	0.0534
50	100	30	0.0373	100	100	30	0.1705
50	100	45	0.0436	100	100	45	2.4085
50	150	15	0.0470	100	150	15	0.0716
50	150	30	0.0518	100	150	30	0.4797
50	150	45	0.0642	100	150	45	13.3569
50	200	15	0.0630	100	200	15	0.1000
50	200	30	0.0626	100	200	30	0.8130
50	200	45	0.0938	100	200	45	368.7879
50	250	15	0.0780	100	250	15	0.1334
50	250	30	0.0828	100	250	30	1.0198
50	250	45	0.0970	100	250	45	1775.1040

Table 3a: ANOVA for Solution Speed

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	31875985.750(a)	47	678212.463	18.011	.000
Intercept	1002184.527	1	1002184.527	26.614	.000
Candidate Location (CL)	2894722.282	3	964907.427	25.624	.000
Client Node (CN)	1819471.843	3	606490.614	16.106	.000
Maximum Access Time (MAT)	1998648.728	2	999324.364	26.538	.000
CL * CN	5253896.367	9	583766.263	15.503	.000
CL * MAT	5776647.070	6	962774.512	25.568	.000
CN * MAT	3665836.184	6	610972.697	16.225	.000
CL * CN * MAT	10529056.157	18	584947.564	15.534	.000
Error	16267399.353	432	37656.017		
Total	49145992.198	480			
Corrected Total	48143385.104	479			

a R Squared = .662 (Adjusted R Squared = .625)

Table 3b: ANOVA for Overall Cost (in Thousands)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	196378108.217(a)	47	4178257.622	497.276	.000
Intercept	2292052282.903	1	2292052282.903	272789.165	.000
Candidate Location (CL)	17399837.214	3	5799945.738	690.282	.000
Client Node (CN)	87637875.528	3	29212625.176	3476.748	.000
Maximum Access Time (MAT)	60974206.678	2	30487103.339	3628.430	.000
CL * CN	3321852.301	9	369094.700	43.928	.000
CL * MAT	25059246.935	6	4176541.156	497.072	.000
CN * MAT	700380.383	6	116730.064	13.893	.000
CL * CN * MAT	1466230.612	18	81457.256	9.695	.000
Error	3629787.084	432	8402.285		
Total	2493068080.597	480			
Corrected Total	200007895.301	479			

a R Squared = .982 (Adjusted R Squared = .980)

Table 3c: ANOVA for the Number of Selected Facility Locations

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	68701.955(a)	47	1461.744	558.511	.000
Intercept	524228.793	1	524228.793	200300.332	.000
Candidate Location (CL)	11981.087	3	3993.696	1525.934	.000
Client Node (CN)	6698.308	3	2232.769	853.109	.000
Maximum Access Time (MAT)	33997.962	2	16998.981	6495.068	.000
CL * CN	1791.732	9	199.081	76.066	.000
CL * MAT	13214.501	6	2202.417	841.512	.000
CN * MAT	402.104	6	67.017	25.606	.000
CL * CN * MAT	738.823	18	41.046	15.683	.000
Error	1130.636	432	2.617		
Total	594270.000	480			
Corrected Total	69832.592	479			

a R Squared = .984 (Adjusted R Squared = .982)

Table 3d: ANOVA for Expected Latency

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	8967.304(a)	47	190.794	450.123	.000
Intercept	77317.351	1	77317.351	182408.178	.000
Candidate Location (CL)	300.459	3	100.153	236.282	.000
Client Node (CN)	87.731	3	29.244	68.992	.000
Maximum Access Time (MAT)	8485.903	2	4242.951	10010.030	.000
CL * CN	13.779	9	1.531	3.612	.000
CL * MAT	45.549	6	7.591	17.910	.000
CN * MAT	23.173	6	3.862	9.112	.000
CL * CN * MAT	6.002	18	.333	.787	.717
Error	183.112	432	.424		
Total	86589.488	480			
Corrected Total	9150.415	479			

a R Squared = .980 (Adjusted R Squared = .978)

Figure 1: The Impact of Maximum Latency on the Solution Speed

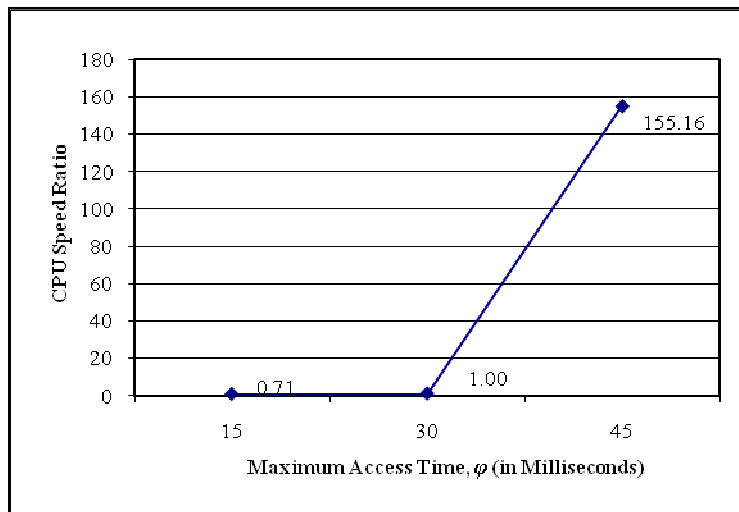


Figure 2: The Impact of Maximum Latency on the Number of Selected Facility Locations

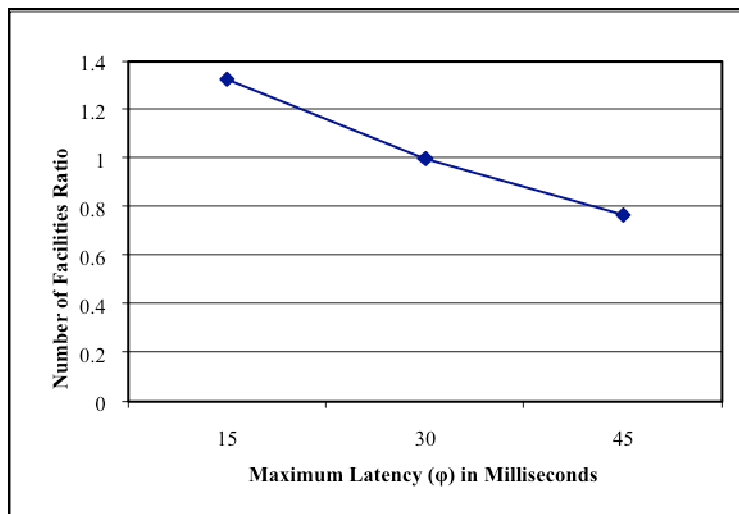


Figure 3: The Impact of Maximum Latency on the Overall Cost

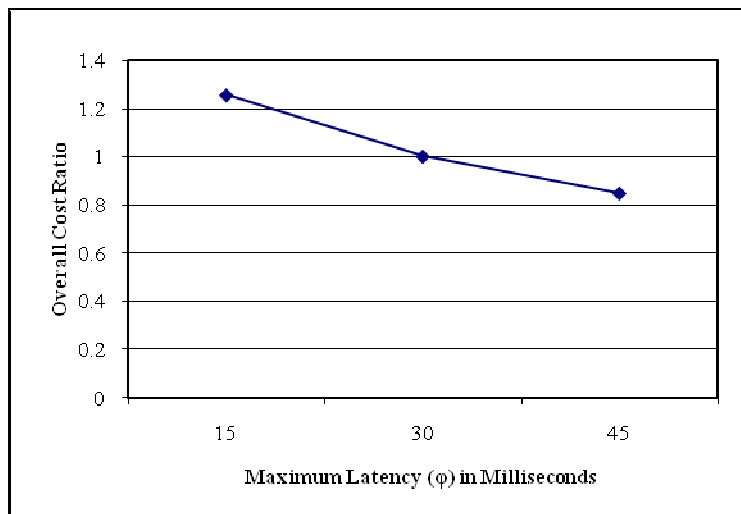


Figure 4: The Impact of Maximum Latency on the Expected Latency

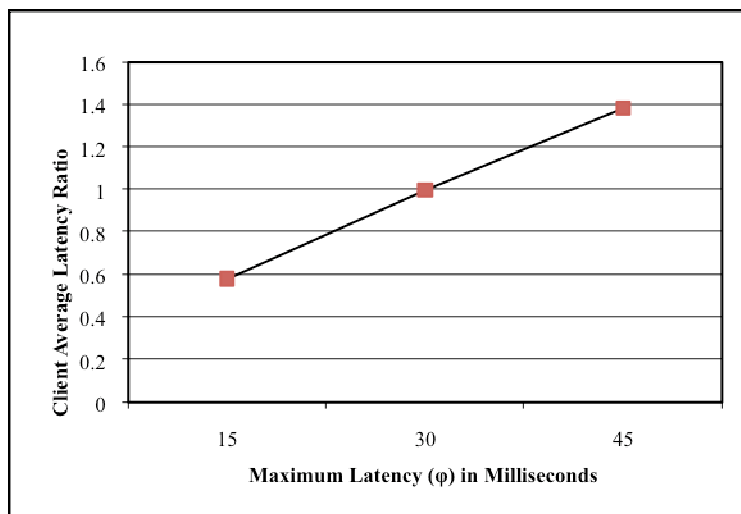


Figure 5: The Impact of Changes in Unit Server Cost, Fixed Facility Cost, and Service Request Arrival Rate on the Solution Speed

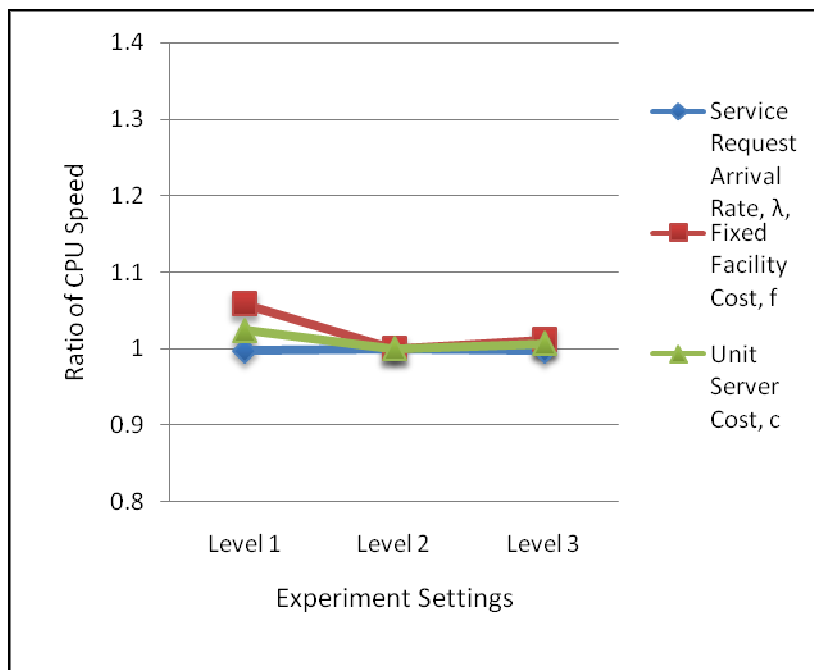


Figure 6: The Impact of Changes in Unit Server Cost, Fixed Facility Cost, and Service Request Arrival Rate on the Number of Selected Facility Locations

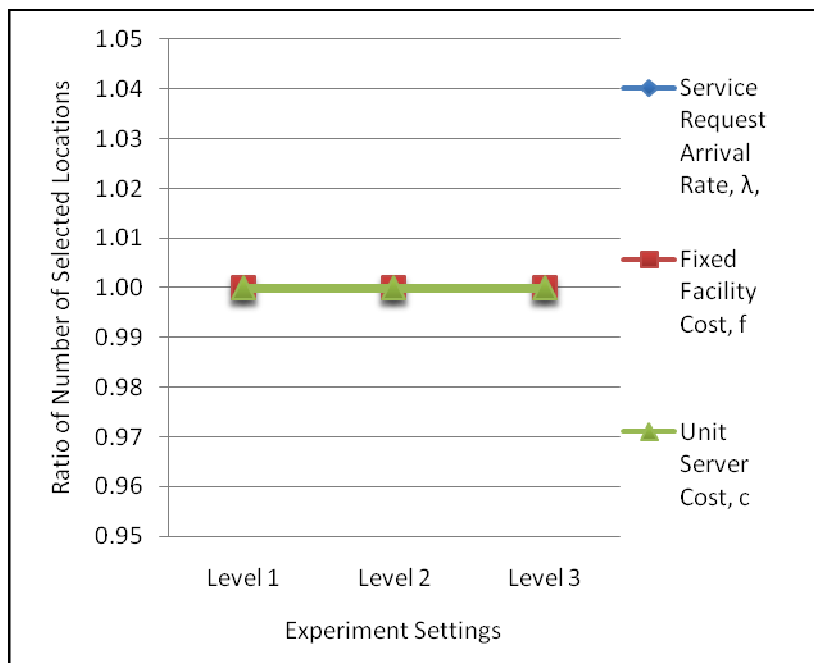


Figure 7: The Impact of Changes in Unit Server Cost, Fixed Facility Cost, and Service Request Arrival Rate on the Overall Cost

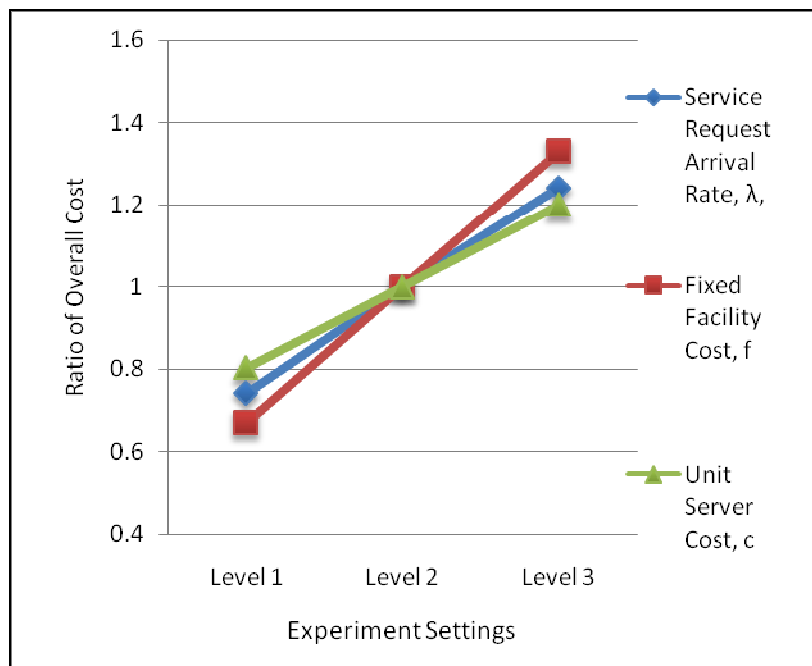


Figure 8: The Impact of Changes in Unit Server Cost, Fixed Facility Cost, and Service Request Arrival Rate on the Expected Latency

