Research

# Sensitivity Analysis of Server Placement on Enterprise Network Topology Through Soft Computing

Maytham Safar, Sami Habib
Kuwait University
Computer Engineering Department
P. O. Box 5969, Safat 13060, Kuwait
{maytham, shabib}@eng.kuniv.edu.kw

Journal of Digital
Information Management

***ABSTRACT:*** *We analyze the consequences when both the server placement problem and network topology problem are solved concurrently through soft computing. Both problems are formulated as a combined optimization problem, subject to a set of design and performance constraints while minimizing the enterprise network cost. We have coded the combined optimization problem within a soft computing methodology, which is based on a probabilistic genetic program for automatically searching the design space for good network topologies. The experimental results for synthesizing and optimizing 3-level enterprise network (65 user nodes) for an edge-server placement has demonstrated the effectiveness of our methodology in finding good solutions with a static workload in less than five minutes.*

**Categories and Subject Descriptors**
**C.2.1[Network Architecture and Design];**Network Topology: **[C2.4 [Distributed Systems];**Client/Server

**General Terms**
Enterprise networks, Soft computing, Web server placement

**Keywords:** Edge-server placement, network topology, analysis, optimization, soft computing, clustering, web structure analysis

Received 10 March 2006; Revised and accepted 27 Sep. 2006

## 1. Introduction

The Internet has been experiencing a growth in terms of increasing numbers of users, servers, capacity of the networks, and the available information. The phenomenal growth of the amount of available information requires to store and access large databases. Because of the nature of the Internet as a distributed information system, heterogeneity, frequent changes, large size, and non-uniformity of information access, the growth of Internet in terms of increasing resources can't be sustained in keeping up with users' demands.

To improve the performance for end users, network operators, and content providers, it is vital to place the information closer to the users. This would speed up web access, reduce latency perceived by the users, network traffic, and server load, and improve response time to the users [18][19]. However, since the Web is huge in size, and is a distributed system rather than a centralized system, delivering the requested information from remote servers to users in time, while placing the least demands on the servers and networks is a challenging optimization (scheduling) problem. The solution lays in sharing the limited Internet's resources (servers and networks) among the users wisely, and developing new web retrieval techniques.

For example, effective and efficient server placement within an enterprise network requires a sensitivity analysis to identify how well the network and server hardware devices are integrated. We believe that the sensitivity analysis plays a vital role in the success of enterprises by maintaining their entire database at easy reach for both the employees and clients, as demonstrated by an early sensitivity analysis [17]. Designing an enterprise network comprises of solving the network topology problem while considering the server placement. The database will be stored near the placed servers. If the database consists of data-intensive applications like multimedia, then such enterprise network is not only characterized by massive bandwidth and data storage requirements, but how well the network and server hardware devices are integrated. This situation poses a design challenge since enterprise network designers should design a robust and low network topology cost, which meets both the users' communications and file accesses. Therefore, we have combined server placement and network topology problems as a single optimization problem.

The network topology problem includes determining network topologies along with the network technologies, such as ATM switch, Ethernet hub, and IP router, that enable all users to communicate and access servers efficiently, while minimizing the enterprise network cost. The server placement problem is to determine the number, locations, storage and process capacities of servers, while minimizing their placement costs and satisfying all users' requests.

Network topology and server placement are interdependent problems, which should be combined as a single optimization problem to reach optimal solutions. However, the resulting optimization problem is an intractable. Therefore, a soft computing approach offers to exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness and low solution cost [24]. We have coded the combined optimization problem in a probabilistic soft computing program, which is based on the genetic algorithms. In this paper, we have simulated the program with several server placement situations and analyzed the consequences on the enterprise network topologies. Such simulated analysis helps us in comparing and justifying a capital investment in network devices for high-speed networks.

The rest of paper is organized into six sections. The related work on network topology and server placement have been discussed in section 2. In section 3, we classified the database system based on the server placement into centralized and distributed. The enterprise network model is presented in Section 4. We described the soft computing methodology in Section 5. The experimental results are presented in Section 6. Section 7 concludes the work.

## 2. Related work

The automatic planning and integration of a complete enterprise design problem (including both network topology and server placement) has not been reported in the research literature to our knowledge. Most research and commercial tools are focused on network or server analysis, because the automatic network and server placement design problem has quickly become a very complex problem, due to the large design space, and tools have not caught up.

Network tools and methodologies, which are described by the following papers [3][5][7][8][13][14][15], are either designing a virtual network topology using pre-existing physical network topology, limiting the number of network levels to one or two, limiting the problem to network synthesis and not considering server placement, using one specific network technology or using rigid design techniques. Customer support tools are available for specific product lines, for example.

The teleprocessing network problem described in the literature [6][10] is to find an optimal topological network design problem for two classified sets of nodes. The first set contains users and the second set contains servers. In contrast to our problem and approach, the locations of the servers in the teleprocessing network problem is fixed at one specific location. Also, we consider two types of data management systems depending on the application requirements: a centralized server or a distributed server.

There are a number of recent published papers [2][4][11][20], which have examined and proposed solutions to various versions of the server placement problem within given networks. A paper by a group of researchers from IBM Watson Research Center [2] proposes a tractable model to analyze the theoretical effectiveness of network with many small servers versus a network with a few more powerful servers. Cronin *et al.* [4] formalizes the server placement problem as the mirror placement problem of Internet content and the objective is to improve the performance of the network. A paper by a group of researchers from Bell Laboratories [11] reformulates the server placement problem as the cache location problem and the objective is to minimize the overall traffic in the network and reduce the average delay to the users. A paper by Shi and Turner [20] formulates the server placement problem in an overlay network as the set cover problem.

## 3. The database architecture

The server is an important part of the database system, which can be classified based on the server placement. Therefore, a database system can be either centralized or distributed as shown in Figure 1.

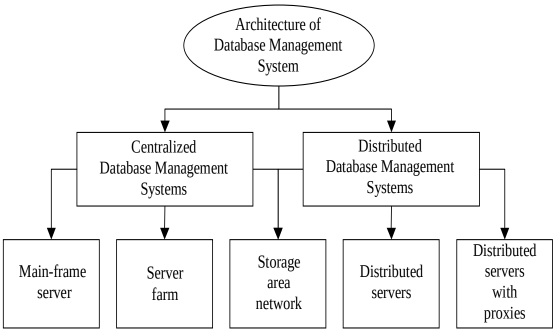

Figure 1. The possible database system architectures

A centralized database system comprises a single *mainframe server* or a *server farm*, where one location within the network is selected to house either approach. On the other hand, a distributed database system can be comprised of a number of *distributed servers* or a number of *proxy servers attached to servers*. The main difference between the distributed database system approaches is the content of servers and proxies. A server's or proxy's content refers to the number and type of files that are stored within a server/proxy's storage space. In the distributed servers approach, the content of a server may not be a subset of other servers. In the distributed servers with proxies, the content of each proxy is a subset of its binding server(s) and also a proxy's content is directly related to its users' requests.

A storage area network (SAN) is an alternative architecture combining the advantages of both centralized and distributed database architectures [16]. According to [22], a server is not connected to any one storage device, and all storage devices are potentially available to all servers; moreover, a dedicated network makes connections between servers and storage devices.

We have considered the edge-server, which should be placed away from the enterprise core network (backbone) and placed at the edge of the network, such as at the local area networks. Such approach provides better load balancing, since the server load is distributed away from the core network.

## 4. Network model and application

We have modeled an enterprise network application as a hierarchy of 3-level tasks [9]. The first task is referred to as *backbone task*, which is performed at a number of physical sites, each of which performs a *site task* (second task). A site task consists of a number of distinct *group tasks* (third task), where each group task comprises a number of distinct *user nodes* (workstations). These three tasks (backbone, site and group) correspond to the three network levels. The group network (local area network, LAN), the lowest level of network contains a set of users (workstations). Clustering the workstations into groups (LANs) and clustering groups into sites are assumed to represent the structure of the network application ideally. In order to perform all the collaborative group and site tasks within an acceptable time, all users need to communicate among themselves, and share servers for storing/retrieving files efficiently.

An animation production studio is described here as an example of a 3-level enterprise network problem. The content growth for such enterprise has been exacerbated by the concurrent growth in the sizes of *data sets*. The film *Toy Story 2* has 122,699 frames of up to 4 gigabytes per frame [21]. This data reflects the finished film, which means that an enormous quantity of data is created within all production tasks to develop the finished film (for now, we are considering a worst-case estimation of traffic flow between tasks). Animation network is expected to have certain characteristics, such as high communication bandwidth, large storage space, and low delay bounds. According to Weinberg [23], digital media production has rapidly become a highly distributed collaborative activity involving teams of people and digital resources in different locations. A typical animation network consists of four collaborative site tasks, such as live-action, audio, background and special effect, and drawing, where ten group tasks are divided among the four site tasks as shown in Figure 2. For this example, there are 65 user nodes (workstations) used within the ten group tasks.

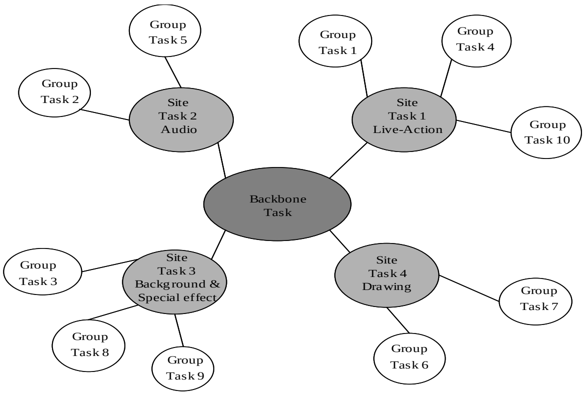

Figure 2. A typical animation network (4 site tasks and 10 group tasks)

The animation enterprise network and other network applications can be described by one matrix and two tables: *user traffic matrix* (UTM) represents the average user-to-user traffic requirements, *user location table* (ULT) represents the physical location of each user within the network, and *file request table* (FRT) represents the access rate of each file by all users. In this paper, we have analyzed the effect of placing server at the edge of the network. Thus, we have constrained the server placement to the group tasks only, where each group task can have at most one server.

## 5. Overview of the soft computing

The structure of soft computing consists of four major procedures as shown in Figure 3. The first procedure, *server placement designs*, creates a population of designs by selecting and placing servers and their hard disks at group task either at random or at a specific location. The second procedure, *initial network designs*, selects and integrates network devices to create the initial 3-level network topologies. If a network topology satisfies all users' communications and files access requirements, then the program proceeds into the optimization loop. Otherwise the network design is considered as invalid and it must be redesigned to satisfy all users' communications and files accesses. The third procedure, *network designs evaluation*, evaluates the cost and estimates the *average network delay* (AND) of each member of the population. The fourth procedure, *network designs optimization*, selects some of the fittest networks and modifies the rest. The third and fourth procedures represent the optimization process by executing as many times as possible to achieve good designs, then the program terminates.

### 5.1 An overview of genetic algorithm

The basic structure of GA, as shown in Figure 4, is a powerful search technique that is used to solve many combinatorial problems [12]. The genetic algorithm starts with an initial population P (t=0) of solutions encoded as chromosomes. An initial population is most often generated randomly but a heuristic technique can be used to create the initial population. Each chromosome is made of a sequence of genes and every gene controls the inheritance of specific attributes of the solution's characteristics. A fitness function measures the quality of the chromosome, and in our case the chromosome's fitness represents the total covered area by all placed sensors. A fit chromosome suggests a better solution. In the evolution process relatively fit chromosomes reproduce new chromosomes and inferior chromosomes die. This process continues until a chromosome with desirable fitness is found. These selected chromosomes, known as parents, are used to reproduce the next generation of chromosomes, known as offspring. The evolution process involves selection operation, and two genetic operations namely, crossover and mutation.

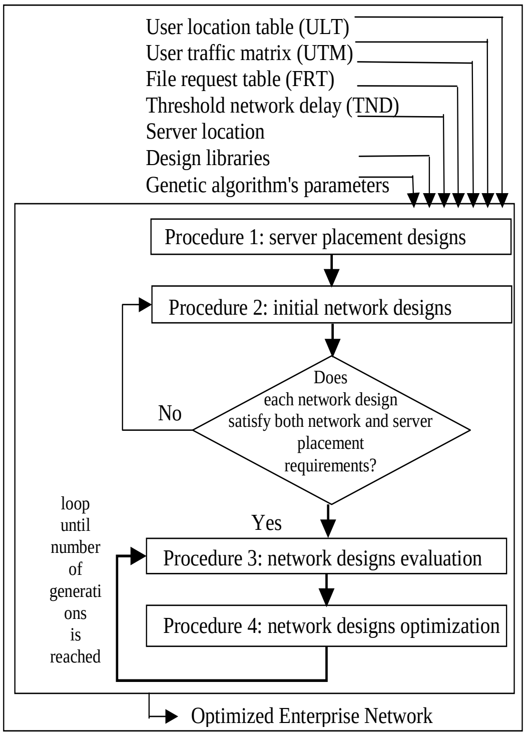

Figure 3. An overview of the soft computing program

The inputs to the soft computing program are application inputs and tool inputs. The application inputs: user location table (ULT), user traffic matrix (UTM), file request table (FRT), and threshold network delay (TND) describe the application tasks. The ULT, UTM, and FRT vary from application to application. The threshold network delay (TND) is a real value given by the designer to insure that the average network delay (AND) of a synthesized 3-level network never exceeds the TND. The value AND is estimated by summing all the delays generated by all network devices with the 3-level network topology. Such a performance method (network of M/M/1 queues) is known for its simplicity and quickness [1] and it is embedded within our program.

```
Genetic Algorithm:

begin
    t =0;
    initialize chromosomes P (t);
    evaluate chromosomes P (t);
    while (termination conditions are unsatisfied)
    begin
        t = t + 1;
        select P (t) from P (t-1);
        mutate some of P (t);
        crossover some of P (t);
        evaluate chromosomes P (t);
    end
```

Figure 4. The basic structure of Genetic Algorithm

| Site Tasks (ST) | Group Tasks (GT) | User Nodes (UN) |
|---|---|---|
| 1 | 1 | 1-5 |
| | 4 | 11-15 |
| | 10 | 6-10 |
| 2 | 2 | 16-20 |
| | 5 | 21-30 |
| 3 | 3 | 31-37 |
| | 8 | 44-50 |
| | 9 | 38-43 |
| 4 | 6 | 51-57 |
| | 7 | 58-65 |

Table 1. Users and groups clustering information.

| Task | Local Traffic (Mbps) | Outgoing Traffic (Mbps) | Incoming Traffic(Mbps) |
|---|---|---|---|
| Site task 1 | 15.0 | 55.625 | 2.5 |
| Site task 2 | 0.0 | 15.0 | 21.25 |
| Site task 3 | 8.4 | 15.0 | 33.75 |
| Site task 4 | 4.2 | 7.5 | 35.625 |
| Group task 1 | 6.0 | 11.25 | 7.5 |
| Group task 2 | 2.0 | 2.5 | 2.5 |
| Group task 3 | 4.2 | 12.6 | 7.35 |
| Group task 4 | 8.0 | 48.125 | 2.5 |
| Group task 5 | 9.0 | 12.5 | 18.75 |
| Group task 6 | 4.2 | 7.7 | 16.625 |
| Group task 7 | 5.6 | 4.0 | 23.2 |
| Group task 8 | 4.2 | 2.1 | 20.475 |
| Group task 9 | 1.2 | 8.7 | 14.325 |
| Group task 10 | 6.0 | 11.25 | 7.5 |

Table 2. Traffic flow for site and group tasks

| File Identification | File Type | File Size (Mbytes) | Group Task | User Node | Total Requests |
|---|---|---|---|---|---|
| 1 | Image | 15 | 1 | 1-5 | 40 |
| | | | 4 | 11-15 | 15 |
| | | | 10 | 6-10 | 15 |
| 2 | Image | 260 | 1 | 1-5 | 25 |
| | | | 10 | 6-10 | 25 |
| 3 | Video | 905 | 2 | 16-20 | 35 |
| 4 | Text | 4 | 3 | 31-35 | 45 |
| 5 | Text | 3 | 3 | 31-37 | 56 |
| | | | 9 | 38-40 | 24 |
| 6 | Text | 5 | 4 | 11-15 | 50 |
| | | | 5 | 26-30 | 15 |
| | | | 6 | 51-57 | 35 |
| | | | 7 | 58-65 | 40 |
| | | | 8 | 45-50 | 25 |
| 7 | Text | 3 | 5 | 21-25 | 100 |
| 8 | Video | 1685 | 6 | 51-57 | 28 |
| | | | 7 | 58-65 | 32 |
| 9 | Audio | 745 | 6 | 51-57 | 35 |
| | | | 7 | 58-65 | 40 |
| 10 | Image | 235 | 7 | 60-65 | 35 |
| 11 | Video | 975 | 8 | 44-50 | 56 |
| | | | 9 | 41-43 | 24 |
| 12 | Audio | 660 | 8 | 44-50 | 28 |
| | | | 9 | 41-43 | 12 |
| 13 | Image | 528 | 8 | 44-50 | 42 |
| | | | 9 | 41-43 | 18 |
| 14 | Image | 428 | 8 | 44-50 | 35 |
| | | | 9 | 41-43 | 15 |
| 15 | Audio | 750 | 8 | 44-50 | 35 |
| | | | 9 | 41-43 | 15 |

Table 3. A file request table (FRT) for an animation studio.

The tool inputs are the design libraries and the genetic algorithm's parameters. The design libraries consist of network libraries and data management libraries. The network libraries contain attribute information about all network device types (ATM switch, Ethernet hub, IP router, gateway) such as cost, capacity, number of ports, and type of wire. The data management libraries contain attribute information about all database device types (server and hard disk) such as cost, process and storage capacities. The genetic algorithm's parameters refer to the population size (PS), number of generations (NG), crossover rate (CR), and mutation rate (MR).

The output of the current soft computing program is a population of optimized 3-level network topologies that meet all network and server placement design and performance constraints, and have acceptable costs.

| Group Task (GT) | Request Traffic (in bps) | Reply Traffic (in bps) |
|---|---|---|
| 1 | 9.25 | $2.068 \times 10^6$ |
| 2 | 9.24 | $9.357 \times 10^6$ |
| 3 | 14.36 | $0.101 \times 10^6$ |
| 4 | 7.11 | $0.072 \times 10^6$ |
| 5 | 16.36 | $0.109 \times 10^6$ |
| 6 | 13.94 | $21.388 \times 10^6$ |
| 7 | 20.91 | $26.839 \times 10^6$ |

Table 4. Server-group traffic flow

## 6. Experimental results

We implemented a probabilistic soft computing algorithm in C++ (16,500 lines of code) on a SUN Blade 100. The program selects network devices, integrates them into 3-level network topologies, and optimizes topologies for the users' communications and server loads. Here we presented the results for the experiment based on a hypothetical enterprise animation production studio, which is used as an example of a 3-level network topology problem. Such a studio contains four site tasks, ten group tasks and 65 user nodes. Table 1 provides detailed information about the clustering of users and groups. The traffic flow is summarized by three parameters for each task: local traffic, outgoing traffic and incoming traffic, all of which are calculated from the user traffic matrix (UTM). The *local traffic* represents all the traffic flow within a task. The *outgoing traffic* represents all the traffic flow from a task to all other tasks. The *incoming traffic* represents all the traffic flow coming into a task from all other tasks. Table 2 shows traffic flow given for the experiments and it is measured in megabits per second (Mbps).

The traffic flow within the backbone task can be summarized by one parameter (backbone local traffic, BLT) or site

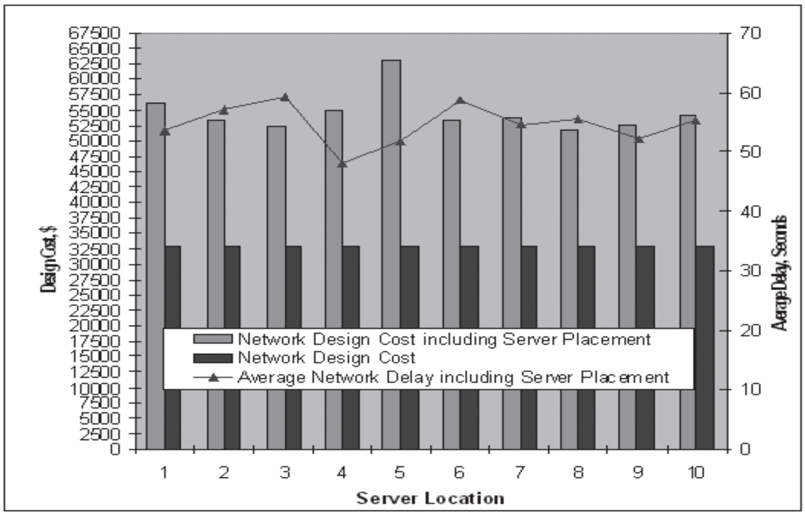

Figure 5. Network design cost versus location of the placed servers within the network (TND = 60 seconds)

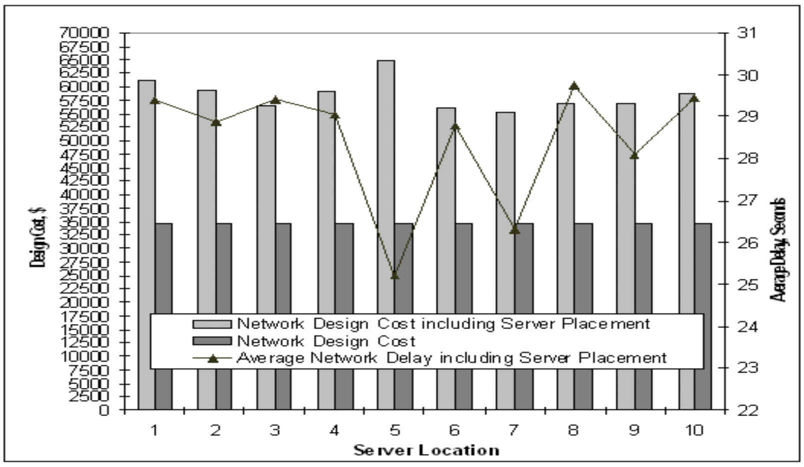

Figure 6. Network design cost versus location of the placed servers within the network (TND = 30 seconds).

trafficmatrix (STM) depending on which topology is selected.

For a local star topology, the backbone local traffic (BLT = 93.125 Mbps) represents all the traffic flow between all sites. Otherwise for a wide tree topology, the traffic flow between site to site is computed also from UTM (here we limit our experiments to a local star backbone topology).

The input information regarding the server placement is given by Table 3. The first three columns of Table 3 identify the file identification, the file media-type (such as text, still image, audio or video), and the file size in megabytes respectively. The fourth and fifth columns identify the group task and user nodes that are requesting the retrieval of such a file. The last column indicates the total number of requests that are made by the users. This file request table (FRT) presents a possible pattern of requests by the 65 users that execute 995 file retrieval requests (an average of 15.3 requests per user).

| Tasks | **All Design Decisions made by the tool when TND = 60 seconds** | | |
|---|---|---|---|
| | **Network topology without server placement** | **Server is placed in group task 5** | **Server is placed in group task 8** |
| Backbone | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 150Kpps, 5 ports, $4,400 | IP Router, 150Kpps, 5 ports, $4,400 |
| Site 1 | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 100Kpps, 10 ports, $4,400 | IP Router, 100Kpps, 5 ports, $2,800 |
| Group 1 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 |
| Group 4 | Ethernet, 100Mbps, 6 ports, $2,940 | Ethernet, 100Mbps, 6 ports, $2,940 | ATM, 75Mbps, 10 ports, $3,500 |
| Group 10 | ATM, 45Mbps, 10 ports, $2000 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 |
| Site 2 | IP Router, 50Kpps, 5 ports, $1,200 | **IP Router, 150Kpps, 5 ports, $4,400** | IP Router, 50Kpps, 5 ports, $1,200 |
| Group 2 | Ethernet, 10Mbps, 8 ports, $91 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 25Mbps, 10 ports, $1,500 |
| Group 5 | ATM, 45Mbps, 15 ports, $3,500 | **Ethernet, 1000Mbps, 12 ports, $9,800 (server)** | ATM, 75Mbps, 15 ports, $4,700 |
| Site 3 | IP Router, 50Kpps, 5 ports, $1,200 | IP Router, 100Kpps, 5 ports, $2,800 | **IP Router, 100Kpps, 5 ports, $2,800** |
| Group 3 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 75Mbps, 10 ports, $3,500 | ATM, 45Mbps, 10 ports, $2,000 |
| Group 8 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 100Mbps, 10 ports, $4,250 | **Ethernet, 1000Mbps, 10 ports, $8,400 (server)** |
| Group 9 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 |
| Site 4 | IP Router, 50Kpps, 5 ports, $1,200 | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 100Kpps, 5 ports, $2,800 |
| Group 6 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 75Mbps, 10 ports, $3,500 | ATM, 75Mbps, 10 ports, $3,500 |
| Group | ATM, 45Mbps, 10 ports, $2,000 | ATM, 75Mbps, 10 ports, $3,500 | ATM, 75Mbps, 10 ports, $3,500 |
| | Design Summary:<br>Network cost = $29,731.00<br>Wiring Cost = $3,535.00<br>Bridging Cost = $0.00<br>AND = 41.52 secondsHF = 0.53 | Design Summary:<br>Network cost = $54,290.00<br>Wiring Cost = $8,917.00<br>Bridging Cost = $0.00<br>AND = 51.87 secondsHF = 0.53 | Design Summary:<br>Network cost = $47,100.00<br>Wiring Cost = $4,599.00<br>Bridging Cost = $0.00<br>AND = 55.56 secondsHF = 0.6 |

Table 5. Three network designs produced by the soft computing methodology when TND = 60 seconds

| Tasks | **All Design Decisions made by the tool when TND = 30 seconds** | | |
|---|---|---|---|
| | **Network topology without server placement** | **Server is placed in group task 5** | **Server is placed in group task 7** |
| Backbone | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 200Kpps, 5 ports, $6,800 | IP Router, 150Kpps, 5 ports, $4,400 |
| Site 1 | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 100Kpps, 5 ports, $2,800 |
| Group 1 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 |
| Group 4 | Ethernet, 100Mbps, 6 ports, $2,940 | Ethernet, 100Mbps, 6 ports, $2,940 | Ethernet, 100Mbps, 6 ports, $2,940 |
| Group 10 | ATM, 45Mbps, 10 ports, $2000 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 |
| Site 2 | IP Router, 50Kpps, 5 ports, $1,200 | **IP Router, 150Kpps, 5 ports, $4,400** | IP Router, 50Kpps, 5 ports, $1,200 |
| Group 2 | Ethernet, 10Mbps, 6 ports, $49 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 25Mbps, 10 ports, $1,500 |
| Group 5 | ATM, 75Mbps, 15 ports, $4,700 | **Ethernet, 1000Mbps, 12 ports, $9,800 (server)** | ATM, 75Mbps, 15 ports, $4,700 |
| Site 3 | IP Router, 50Kpps, 5 ports, $1,200 | IP Router, 100Kpps, 5 ports, $2,800 | IP Router, 100Kpps, 5 ports, $2,800 |
| Group 3 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 45Mbps, 10 ports, $2,000 |
| Group 8 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 100Mbps, 10 ports, $4,250 | ATM, 100Mbps, 10 ports, $4,250 |
| Group 9 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 75Mbps, 10 ports, $3,500 | ATM, 75Mbps, 10 ports, $3,500 |
| Site 4 | IP Router, 50Kpps, 5 ports, $1,200 | IP Router, 100Kpps, 5 ports, $2,800 | **IP Router, 150Kpps, 5 ports, $4,400** |
| Group 6 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 75Mbps, 10 ports, $3,500 | ATM, 75Mbps, 10 ports, $3,500 |
| Group 7 | ATM, 45Mbps, 10 ports, $2,000 | ATM, 100Mbps, 10 ports, $4,250 | **Ethernet, 1000Mbps, 10 ports, $8,400 (server)** |
| | Design Summary:<br>Network cost = $30,889.00<br>Wiring Cost = $3,535.00<br>Bridging Cost = $0.00<br>AND = 24.00 seconds<br>HF = 0.53 | Design Summary:<br>Network cost = $55,840.00<br>Wiring Cost = $8,917.00<br>Bridging Cost = $0.00<br>AND = 25.19 seconds<br>HF = 0.53 | Design Summary:<br>Network cost = $50,390.00<br>Wiring Cost = $4,934.00<br>Bridging Cost = $0.00<br>AND = 26.31 seconds<br>HF = 0.53 |

Table 6. Three network designs produced by the soft computing methodology when TND = 30 seconds

The traffic generated from requesting and retrieving files in Table 3 is listed in Table 4. This table shows the server access traffic flow at a group task (column 1) when the users within a group are requesting file retrieval from servers. The second column represents the traffic flow generated by sending all users' requests within the group task to servers. The third column represents the traffic flow generated by servers to reply to all users' requests. This traffic is in addition to the traffic between users shown in Table 2.

In the experiments, the lowest enterprise network design cost, which satisfied all design and performance constraints, found by the soft computing program, is considered as the recommended solution to the problem. Also a proportionate selection scheme [12] was used with the following parameters: population size (PS) = 500, number of generations (NG) = 5000, mutation rate (MR) = 0.05, and crossover rate (CR) = 0.80. The proportionate selection scheme is a simple selection method, which compares each member's fitness function with the average fitness function of the entire population. In our formulation, the fitness function represents the total network cost – summing the cost of all network devices, protocol translators, and wiring. If a member's fitness function is less than or equal to the average fitness function of the entire population, then this member is kept for the next generation. Otherwise, the member is selected for redesign. The convergence criterion used in our experiments is to terminate the program when the number of generations reaches the limit specified.

We ran the program with two different values of threshold network delay (TND) 60.0 and 30.0 seconds. Figure 5 shows a plot representing the trade-off in the network design cost when TND = 60.0 seconds and the server is assigned at a specific group location (from 1 to 10). Each point in the plot represents the network design cost for ten local area networks (LAN), four site networks and one backbone network.The network design cost depends on the design decisions made by our program. For each server placement, the program generates a design with a different cost ranging from $51,699.60 to $63,207.00. Thus, the soft computing program provides us with different network topologies, where there is 22.3% trade-off in design cost. The average network delay (AND) for ten designed networks is in the range of 47.97 to 59.23 seconds. On the other hand, designing the 3-level network topology without considering server load comes up to be $33,266.00 with AND = 41.52 seconds.

Figure 6 shows a plot representing the trade-off in the network design cost with a tight TND = 30.0 seconds and the server is assigned at a specific group location (from 1 to 10). For each server placement, the program generates a design with a different cost ranging from $55,324.50 to $64,757.00. Thus, our soft computing program provides different network topologies, where there is 15.05% trade-off in design cost. The average network delay (AND) for ten designed networks is in the range of 25.19 to 29.75.

Tables 5-6 illustrate all design decisions made by soft computing for the two extreme designs (lowest design cost and highest design cost) when TND is assigned to 60.0 and 30.0 seconds respectively, and the lowest network design cost without considering the server placement. The network *homogeneity factor* (HF) is an output parameter ranging (0.0,1.0); HF = 1.0 indicates that all the allocated network hardware components are based on the same technology. Otherwise, HF indicates the ratio of the maximum number of allocated network components of the same technology to the total allocated network components in the network. From Tables 5 and 6, we have observed the usage and domination of GigaEthernet hub at the location of the placed server, since it has the bandwidth capacity to handle the user-to-user traffic and server load requirements. Also, we have observed the usage and domination of IP router at the backbone and site levels, since it provides a protocol translation without additional cost, especially when it is connecting heterogeneous network technology, such as Ethernet and ATM.

## 7. Conclusion and future direction

We briefly described our experience with a soft computing program, which is based on the genetic algorithms to synthesize and optimize 3-level enterprise networks by considering a server placement at the edge of the network with a static load. Our methodology demonstrates how effective in planning and integrating 3-level network topologies under five minutes on SUN Blade 100. The outcomes of the soft computing method can help or guide the network designer/planner to design and integrate many different data management systems and examine the effect on the network topologies. We will continue to improve the capability of our method by considering various database architectures.

## References

[1] Bersekas, D., Gallager, R. (1992). Data Networks. Prentice Hall.

[2] Calo, S., Verma, D., Giles, J., Agrawal, D (2002). On the Effectiveness on Content Distribution Networks. Proceeding of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems.

[3] Cosares, S., Deutsch, D., Saniee, I., Wasem, O (1995). Sonet Toolkit: A Decision Support System for Designing Robust and Cost-effective Fiber-Optic Networks. Interfaces, 25 (1) 20-40.

[4] Cronin, E., Jamin, S., Jin, C., Kurc, A., Raz, D., Shavitt, Y (2002). Constrained Mirror Placement on the Internet. *IEEE Journal on Selected Areas in Communications*, 20 (7)

[5] Doshi, B., Dravida, S., Harshavardhana, P (1995). Overview of INDT – A New Tool for Next Generation Network Design. Proceedings of the IEEE Globecom.

[6] Esau, L., Williams, K (1966). On Teleprocessing System Design. *IBM System Journal*, 5, 142-147.

[7] Gerla, M., Kleinrock, L (1977). On the Topological Design of Distributed Computer Networks. IEEE Trans. on Communications, 25 (1) 48-60.

[8] Gersht, A., Weihmayer, R (1990). Joint Optimization of Data Network Design and Facility Selection. IEEE Journal on Selected Areas in Communications, 8 (9) 1667-1681.

[9] Habib, S., Parker, A., Lee, D. (2002). Automated Design of Hierarchical Intranets. Computer Communications, 25 (11-12) 1066-1075.

[10] Kershenbaum, A., Boorstyn, R (1983). Centralized Teleprocessing Network Design. Network, 13, 279-193.

[11] Krishnan, P., Raz, D., Shavitt, Y. (2000). The Cache Location Problem. *IEEE/ACM Transactions on Networking*, 8 (5).

[12] Michalewicz, Z (1994). Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer-Verlag.

[13] Mitra, D., Morrison, J. & Ramakrishnan, K. (1998). VPN Designer: A Tool for Design of Multiservice Virtual Private Networks. Bell Labs Technical Journal, 3(4), 15-31.

[14] Palmer, C., Kershenbaum, A. (1995). An Approach to a Problem in Network Design using Genetic Algorithms. Networks, 26 (10) 151-163.

[15] Pierre, S.,Legault, G (1998). A Genetic Algorithm for Design Distributed Computer Network Topologies. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, 28, 249-258.

[16] Phillips, B (1998). Have Storage Area Networks Come of Ages? IEEE Computer, 10-12.

[17] Safar, M., Shahabi, C., Tan. C. (2000). Resiliency and robustness of alternative shape-based image retrieval techniques. Proceedings of IEEE international database engineering and applications symposium (IDEAS).

[18] Safar, M (2002). Classification of Web Caching Systems. Proceedings of International Conference on WWW/Internet.

[19] Safar, M (2004). Design Issues of Web Caching Systems. Proceedings of International Applied Computing Conference.

[20] Shi, S. Turner, J. (2002). Placing Servers in Overlays Networks. Proceeding of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems.

[21] Slaton, J. Toys Will Be Toys. http://www.wired.com/news/culture/0,1284,32591,00.html

[22] Thornburgh, R., Schoenborn, B. (2001). Storage Area Networks: Designing and Implementing A Mass Storage System. New Jersey: Prentice-Hall.

[23] Weinberg, R (1995). Producing Content Producers. *IEEE Communications Magazine,* 33 (8) 70-73.

[24] Zadeh, L (1994). Soft Computing and Fuzzy Logic. IEEE Software, 48-56.

Maytham Safar is currently an Assistant Professor at the Computer Engineering Department at Kuwait University. He received his Ph.D. degree in Computer Science from the University of Southern California in 2000. Dr. Safar's current research interests include Peer-to-Peer Networks, Spatial Databases, Multidimensional Databases, and Geographic Information Systems. He served on many conferences as a reviewer and/or a scientific program committee member. He also served as a member on the editorial board or a reviewer for many journals such as IEEE TMJ, ACM Computing Reviews, IAJIT, MTAP, IEEE TPAMI, CAI, and ACM MSJ.

Sami J. Habib received the B.S. degree in computer engineering in 1993 from Iowa State University. Then, he pursed a graduate study at the University of Southern California, where he earned the M.S. and Ph.D. degrees in computer engineering in 1995 and 2001 respectively. Currently, he is an assistant professor in the Computer Engineering Departtment at Kuwait University. His current work focuses on developing computer aided design methodologies and performance analysis techniques for designing/redesigning distributed systems, especially the network topology and data management system. Dr. Habib has served as a reviewer for a number of international conferences and journals.