Simulated Analysis of Server Placement on Network Topology Designs

Sami J. Habib
Kuwait University
Computer Engineering Department
P. O. Box 5969 Safat 13060 Kuwait
shabib@eng.kuniv.edu.kw

***Abstract*— We analyze the consequences when both the server placement problem and network topology problem are solved concurrently through a soft computing. Both problems are formulated as a combined optimization problem, subject to a set of design and performance constraints while minimizing the network design cost. In this paper, we have encoded the combined optimization problem within a capacity-planning tool, called NETCAP, which is a probabilistic soft computing program for automatically searching the design space. The experimental results described here demonstrate the effectiveness of NETCAP in finding good optimized 3-level network topologies (65 user nodes) for server placement with a static load under five minutes.**

I. Introduction

Effective and efficient a server placement within network are a requirement for data-intensive applications like multimedia. Such application environments are not only characterized by massive bandwidth and data storage requirements, but how well the network and server are integrated.

This situation poses a design challenge since network designers/planners must design a robust and low network topology cost, which meets both the users' communications and file accesses. Therefore, we 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, Ethernet, IP router, bridge, T-carrier, VPN and/or SONET, that enable all users to communicate and access servers efficiently, while minimizing the network design 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 must 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 [23]. We encoded the combined optimization problem in a probabilistic soft computing program - NETCAP, which is based on genetic algorithms. In this paper, we simulated the NETCAP with several server placement situations and analyzed the consequences on the network topologies. Such analysis can help a network administrator in justifying a capital investment in upgrading network physical topologies.

This paper is part of an ongoing research project to automated networks planning and integration for multimedia applications, including animation production, telemedicine, and tele-education intranets [10][9][11].

II. An Overview of Data Management Architectures

A data management system can be either centralized or distributed as shown in Figure 1. A centralized data management 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 data management system can be comprised of a number of distributed servers or a number of proxy servers attached to servers. The main different between the distributed data management 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's or 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 proxy servers attached to servers approach, the content of a 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 data management architectures [17]. According to [21], 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. Currently, NETCAP is capable of planning and integrating two types of data management systems: mainframe server, and distributed servers.

0-7803-8735-X/05/$20.00©2005 IEEE

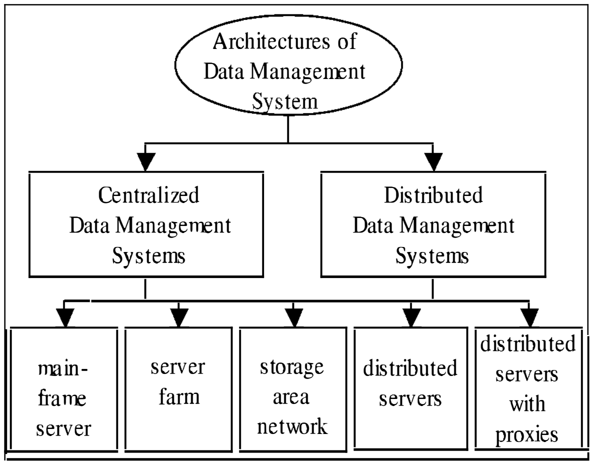

Fig. 1. The possible data management system architectures.

III. Related Work

The automatic planning and integration of a complete network design problem (including both network topology and server placement) has not been reported in the research literature. Most research and commercial tools are focused on network/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 [3][5][7][8][15][16][18] are either designing a logical 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][12] 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 server 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 distributed servers.

There are a number of recent published papers [2][4][13][19], 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 network. A paper by a group of researchers from Bell Laboratories [13] 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 [19] formulates the server placement problem in an overlay network as the set cover problem.

IV. Network Model and Application

We model a network application as a hierarchy of 3-level tasks. The first task refers 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 (backbone network, intermediate network, local area network). Each local area network (LAN) 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 retrieving files efficiently.

An animation production studio is described here as an example of a 3-level network topology design problem. The content growth for such application 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 [20]. 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 [22], 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) are used within the ten group tasks.

The animation 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

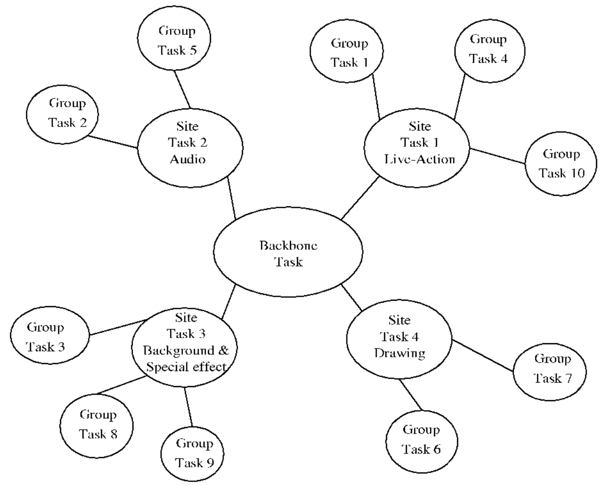

Fig. 2. A typical animation network (4 site tasks and 10 group tasks).

file by all users.

In this paper, we analyzed the effect of placing a 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. All the design and performance constraints are encoded within NETCAP and the reader can refer to the dissertation [10] for the complete formulation of the combined optimization problem.

V. An Overview of the NETCAP

The structure of NETCAP 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 NETCAP 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 and validation*, evaluates the network cost, estimates the average network delay (AND), and validates all design and performance constraints of each member of the population. The fourth procedure, *network designs optimization*, selects some fittest networks and modifies the rest. The third and fourth procedures represent the optimization process by executing as many times to achieve good designs, then NETCAP terminates.

The inputs to NETCAP 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

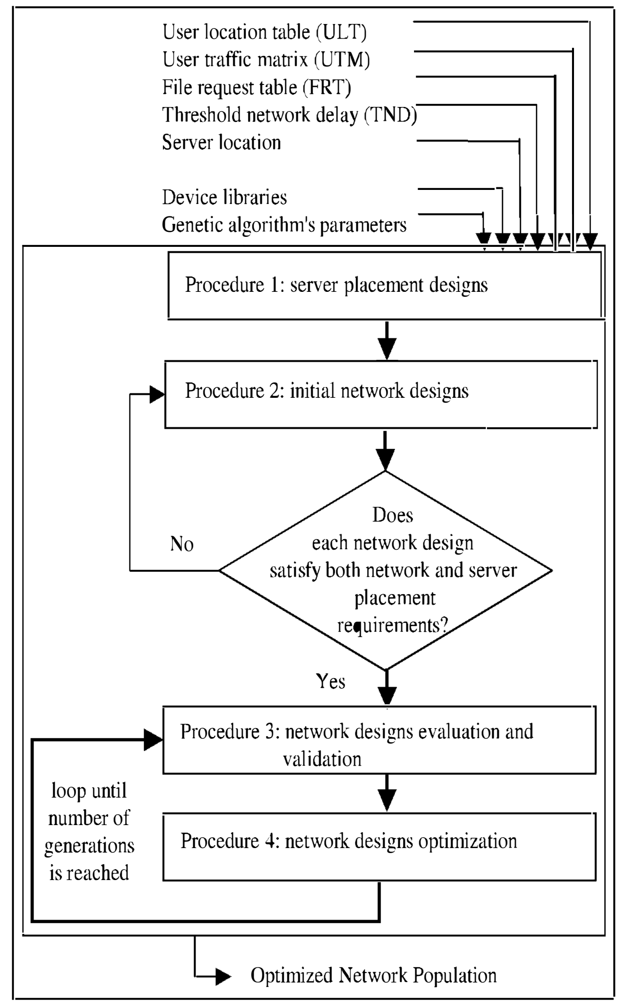

Fig. 3. An overview of the soft computing program NETCAP.

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 NETCAP.

The tool inputs are the design libraries and the genetic algorithm's parameters. The device libraries consist of network libraries and data management libraries. The network libraries contain attribute information about all network device types (ATM, Ethernet, IP router, bridge, T-carrier, SONET, and virtual private network) such 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 NETCAP tool is a population of optimized 3-level network topologies that meet all network and server placement design and performance constraints, and have costs acceptable.

VI. EXPERIMENTS

The NETCAP is a probabilistic soft computing program and was implemented in C++ (16,500 lines of code) on a SUN Blade 100. NETCAP selects network and server devices, integrates them into 3-level network topologies, and optimizes topologies for the users' communications and server loads. Here we presented the results for experiment based on a hypothetical digital 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 I 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 II 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 traffic matrix (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).

TABLE I

USERS AND GROUPS CLUSTERING.

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

The input information regarding the server placement is given by the file request table (FRT), which presents a possible pattern of requests by the 65 users that execute 995 file retrieval requests (an average of 15.3 requests per user). The traffic generated from requesting and retrieving files in FRT is listed in Table III. 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 a server. The second column represents the traffic flow generated by sending all users' requests within the group task to a server. The third column represents the traffic flow generated by a server to reply to all users' requests. This traffic is in addition to the traffic between users shown in Table II.

TABLE II

TRAFFIC FLOW FOR SITE AND GROUP TASKS.

Tasks	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	4.20	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	146.325
group task 10	6.0	11.25	7.5

TABLE III

SERVER-GROUP TRAFFIC FLOW.

Group Tasks (GT)	Request Traffic (in bps)	Reply Traffic (in Mbps)
1	9.25	2.068
2	9.24	9.357
3	14.36	0.101
4	7.11	0.072
5	16.36	0.109
6	13.94	21.388
7	20.91	26.839
8	31.43	39.790
9	15.36	17.059
10	3.56	1.893

In the experiments, the lowest network design cost, which satisfied all design and performance constraints, found by NETCAP, is considered as the recommended solution to the problem. Also a proportionate selection scheme [14] 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 NETCAP when the number of generations reaches the limit specified.

We ran NETCAP with two different values of threshold network delay (TND) 60.0 and 30.0 seconds. Figure 4 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 NETCAP. For each server placement, NETCAP generates a design with different cost ranging from $51,699.60 to $63,207.00. Thus, NETCAP provides us with different network topologies, where there is 22.3% trade-off in design cost. The average network delay (AND) for ten synthesized 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.

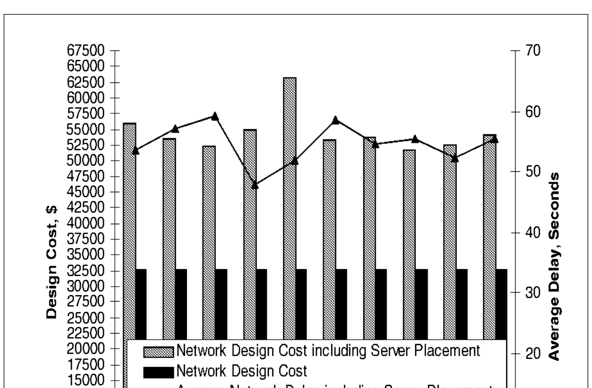

Fig. 4. Network design cost versus server location (TND = 60 seconds).

Figure 5 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, NETCAP generates a design with different cost ranging from $55,324.50 to $64,757.00. Thus, NETCAP provides us with different network topologies, where there is 15.05% trade-off in design cost. The average network delay (AND) for ten synthesized networks is in the range of 25.19 to 29.75 seconds. However, designing the 3-level network topology without considering server load comes up to be $34,424.00 with AND = 27.70 seconds.

Tables IV- V illustrate all design decisions made by NETCAP 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 of NETCAP ranging (0.0,1.0]; HF = 1.0 indicates that all the allocated network hardware devices are based on the same technology. Otherwise, HF indicates the ratio of the maximum number of allocated network devices of the same technology to the total allocated network devices in the network.

At the first glance at the two Tables IV and V, placing the server at a group 5 generated the highest network design cost, but the two networks have some of the lowest ANDs. From the point of view of a capital investment, the best 3-level network topology is when the TND is assigned to 30.0 seconds and the server is placed at a group task 7. Because its network design cost ($55,324.00) and performance (26.31 seconds) are very competitive with respective to the average network design cost ($56,519) and average performance (41.53 seconds) for the top 20 designs

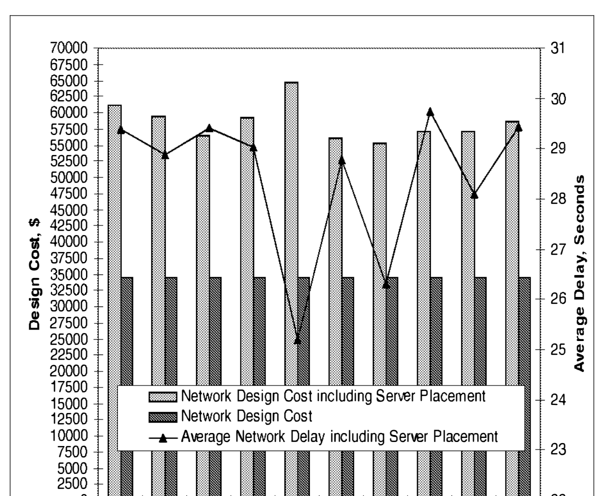

Fig. 5. Network design cost versus server location (TND = 30 seconds).

TABLE IV

THREE NETWORK DESIGNS PRODUCED BY NETCAP WHEN TND = 60 SECONDS.

Tasks	**All Design Decisions made by NETCAP when TND = 60 seconds**		
	Network Topology without Server Placement	**Network Topology with Server at Group 5**	**Network Topology with Server at Group 7**
Backbone	IP Router, 100 Kpps, 5 ports, $2800	IP Router, 150 Kpps, 5 ports, $4400	IP Router, 150 Kpps, 5 ports, $4400
Site 1	IP Router, 100 Kpps, 5 ports, $2800	IP Router, 100 Kpps, 10 ports, $4400	IP Router, 100 Kpps, 5 ports, $2800
Group 1	ATM Switch, 45Mbps 10 ports, $2000	ATM Switch, 45Mbps 10 ports, $2000	ATM Switch, 45Mbps 10 ports, $2000
Group 4	Fast Ethernet Hub, 6 ports, $2940	Fast Ethernet Hub, 6 ports, $2940	ATM Switch, 75Mbps, 10 ports, $3500
Group 10	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 45Mbps, 10 ports, $2000
Site 2	IP Router, 50 Kpps, 5 ports, $1200	**IP Router, 150 Kpps, 5 ports, $4400**	IP Router, 50 Kpps, 5 ports, $1200
Group 2	Ethernet Hub, 10Mbps 8 ports, $91	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 25Mbps 10 ports, $1500
Group 5	ATM Switch, 45Mbps, 15 ports, $3500	**GigaEthernet Hub, 12 ports, $9800 (server)**	ATM Switch, 75Mbps, 15 ports, $4700
Site 3	IP Router, 50 Kpps, 5 ports, $1200	IP Router, 100 Kpps, 5 ports, $2800	**IP Router, 100 Kpps, 5 ports, $2800**
Group 3	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 75Mbps, 10 ports, $3500	ATM Switch, 45Mbps, 10 ports, $2000
Group 8	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 100Mbps, 10 ports, $4250	**GigaEthernet Hub, 10 ports, $8400**
Group 9	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 45Mbps, 10 ports, $2000	ATM Switch, 45Mbps, 10 ports, $2000
Site 4	IP Router, 50 Kpps, 5 ports, $1200	IP Router, 100 Kpps, 5 ports, $2800	IP Router, 100 Kpps, 5 ports, $2800
Group 6	ATM Switch, 45Mbps 10 ports, $2000	ATM Switch, 75Mbps 10 ports, $3500	ATM Switch, 75Mbps 10 ports, $3500
Group 7	ATM Switch, 45Mbps 10 ports, $2000	ATM Switch, 75Mbps 10 ports, $3500	ATM Switch, 75Mbps 10 ports, $3500
Design Summary	Network Cost = $29,731.00 Wiring Cost = $3,535.00 Bridging Cost = $0.00 AND = 41.52 seconds HF = 0.53	Network Cost = $54,290.00 Wiring Cost = $8,917.00 Bridging Cost = $0.00 AND = 51.87 seconds HF = 0.53	Network Cost = $47,100.00 Wiring Cost = $4,599.00 Bridging Cost = $0.00 AND = 55.56 seconds HF = 0.6

TABLE V

THREE NETWORK DESIGNS PRODUCED BY NETCAP WHEN TND = 30 SECONDS.

Tasks	**All Design Decisions made by NETCAP when TND = 30 seconds**		
	Network Topology without Server Placement	**Network Topology with Server at Group 5**	**Network Topology with Server at Group 7**
Backbone	IP Router, 100 Kpps, 5 ports, \$2800	IP Router, 200 Kpps, 5 ports, \$6800	IP Router, 150 Kpps, 5 ports, \$4400
Site 1	IP Router, 100 Kpps, 5 ports, \$2800	IP Router, 100 Kpps, 5 ports, \$2800	IP Router, 100 Kpps, 5 ports, \$2800
Group 1	ATM Switch, 45Mbps 10 ports, \$2000	ATM Switch, 45Mbps 10 ports, \$2000	ATM Switch, 45Mbps 10 ports, \$2000
Group 4	Fast Ethernet Hub, 6 ports, \$2940	Fast Ethernet Hub, 6 ports, \$2940	Fast Ethernet Hub, 6 ports, \$2940
Group 10	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 45Mbps, 10 ports, \$2000
Site 2	IP Router, 50 Kpps, 5 ports, \$1200	**IP Router, 150 Kpps, 5 ports, \$4400**	IP Router, 50 Kpps, 5 ports, \$1200
Group 2	Ethernet Hub, 10Mbps 2 ports, \$49	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 25Mbps 10 ports, \$1500
Group 5	ATM Switch, 75Mbps, 15 ports, \$4700	**GigaEthernet Hub, 12 ports, \$9800 (server)**	ATM Switch, 75Mbps, 15 ports, \$4700
Site 3	IP Router, 50 Kpps, 5 ports, \$1200	IP Router, 100 Kpps, 5 ports, \$2800	**IP Router, 100 Kpps, 5 ports, \$2800**
Group 3	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 45Mbps, 10 ports, \$2000
Group 8	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 100Mbps, 10 ports, \$4250	ATM Switch, 100Mbps, 10 ports, \$4250
Group 9	ATM Switch, 45Mbps, 10 ports, \$2000	ATM Switch, 75Mbps, 10 ports, \$3500	ATM Switch, 75Mbps, 10 ports, \$3500
Site 4	IP Router, 50 Kpps, 5 ports, \$1200	IP Router, 100 Kpps, 5 ports, \$2800	IP Router, 150 Kpps, 5 ports, \$4400
Group 6	ATM Switch, 45Mbps 10 ports, \$2000	ATM Switch, 75Mbps 10 ports, \$3500	ATM Switch, 75Mbps 10 ports, \$3500
Group 7	ATM Switch, 45Mbps 10 ports, \$2000	ATM Switch, 100Mbps 10 ports, \$4250	**GigaEthernet Hub, 10 ports, \$8400 (server)**
Design Summary	Network Cost = \$30,889.00 Wiring Cost = \$3,535.00 Bridging Cost = \$0.00 AND = 24.00 seconds HF = 0.53	Network Cost = \$55,840.00 Wiring Cost = \$8,917.00 Bridging Cost = \$0.00 AND = 25.19 seconds HF = 0.53	Network Cost = \$50,390.00 Wiring Cost = \$4,934.00 Bridging Cost = \$0.00 AND = 26.31 seconds HF = 0.53

showing in Figures 4 and 5. Therefore, the 3-level network topology cost, when the server is placed at group 7, is less than the average cost and within the standard deviation ($3,610). Also, the 3-level network topology performance, when the server is placed at group 7, outperforms the average performance and exceeds the standard deviation (13.67).

From Tables IV and V, we 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 observed the usage and domination of IP router at the backbone and site levels. IP rputer provides a protocol translation without additional cost, especially when it is connecting heterogeneous network technology, such as Ethernet and ATM.

VII. Conclusions and Future Directions

We briefly describe our experience with NETCAP, a probabilistic soft computing that synthesizes and optimizes 3-level network topologies for a server placement with a static load. NETCAP demonstrates how effective in planning and integrating 3-level network topologies under five minutes on SUN Blade 100.

The outcomes of NETCAP 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 NETCAP by considering various data management architecture.

Acknowledgment

The author would like to acknowledge that this work was supported by Kuwait University, Research Grant No. EO 01/02.

References

[1] D. Bersekas and R. Gallager. *Data Networks*. Prentice Hall, Upper Saddle River, NJ, 1992.

[2] S. Calo, D. Verma, J. Giles, and D. Agrawal. On the effectiveness on content distribution networks. In *Proceeding of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, San Deigo, California, USA, 2002.

[3] S. Cosares, D. Deutsch, I. Saniee, and O. Wasem. Sonet toolkit: A decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces*, 25:20–40, January 1995.

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

[5] B. Doshi, S. Dravida, and P. Harshavardhana. Overview of indt - a new tool for next generation network design. In *Proceedings of the 1995 IEEE Globecom*, Singapore, 1995.

[6] L. Esau and K. Williams. On teleprocessing system design. *IBM System Journal*, 5:142–147, 1966.

[7] M. Gerla and L. Kleinrock. On the topological design of distributed computer networks. *IEEE Transactions on Communications*, 25:48–60, January 1977.

[8] A. Gersht and R. Weihmayer. Joint optimization of data network design and facility selection. *IEEE Journal on Selected Areas in Communications*, 8(9):1667–1681, December 1990.

[9] S. Habib. *Synthesis and Optimization of Application-Specific Intranets*. PhD thesis, University of Southern California, 2001.

[10] S. Habib and A. Parker. Computer-aided system integration for data-intensive multimedia applications. In *Proceeding of the ACM Multimedia Conference 2000*, Los Angeles, California, USA, 2000.

[11] S. Habib, A. Parker, and D. Lee. Automated design of hierarchical intranets. *Computer Communications*, 25(11-12):1066–1075, July 2002.

[12] A. Kershenbaum and R. Boorstyn. Centralized teleprocessing network design. *Networks*, 13:279–293, 1983.

[13] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. *IEEE/ACM Transactions on Networking*, 8(5), October 2000.

[14] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, 1994.

[15] D. Mitra, J. Morrison, and K. Ramakrishnan. Vpn designer: A tool for design of multiservice virtual private networks. *Bell Labs Technical Journal*, 3(4):15–31, October-December 1998.

[16] C. Palmer and A. Kershenbaum. An approach to a problem in network design using genetic algorithms. *Networks*, 26:151–163, October 1995.

[17] B. Phillips. Have storage area networks come of ages? *IEEE Computer*, pages 10–12, July 1998.

[18] S. Pierre and G. Legault. A genetic algorithm for design distributed computer network topologies. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 28:249–258, 1998.

[19] S. Shi and J. Turner. Placing servers in overlays networks. In *Proceeding of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, San Deigo, California, USA, 2002.

[20] J. Slaton. Toys will be toys. http://www.wired.com/news/culture/0,1248,32591,00.html.

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

[22] R. Weinberg. Producing content producers. *IEEE Communications Magazine*, 33(8):70–73, August 1995.

[23] L. Zadeh. Soft computing and fuzzy logic. *IEEE Software*, pages 48–56, November 1994.