Exploring Network Topology Evolution Through Evolutionary Computations

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

Alice C. Parker
University of Southern California
Electrical Engineering - Systems
3740 McClintock Ave., LA, CA 90089 USA
parker@eve.usc.edu

ABSTRACT
We present an evolutionary methodology that explores the evolution of network topology when a uniform growth of the network traffic is considered. The network redesign problem is formulated as an optimization problem, subject to a set of design and performance constraints, while minimizing the redesign cost by maintaining as many as possible of the network devices that constitute the original topology. The experimental results for a 3-level network redesign problem (consisting of 65 client nodes) demonstrate the value of the search technique within the genetic algorithms in finding good solutions with respect to redesign cost and time.

Categories and Subject Descriptors
C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *network topology*.

General Terms: Algorithm, Performance, Design, Experimentation.

Keywords: Network topology, redesign, network technology, optimization, and evolutionary computation.

1. INTRODUCTION
Networks are continually being refined and upgraded as needs change and technology evolves. In specific, the network traffic has been growing, partially due to streaming multimedia applications. Network traffic represents both the client-to-client and client-server communications. Therefore, a static traffic matrix, which was used to design a network topology, might not be valid to represent the current network workload a few months after the initial design.

The **network topology redesign** problem is similar to network topology design, which includes determining network topologies along with the choice of network technology, such as ATM, Ethernet, and Router, that enable all clients to communicate and access servers efficiently, while minimizing the network hardware cost. However, the network redesign software should maintain as many as possible of the original network devices as long as the redesign criteria, for example the uniform traffic growth, are satisfied. Furthermore, the cost of replacing old network devices with new network devices should be minimized. Maintaining the original devices creates a legacy problem that may complicate the network design problem, which is already known as an NP-complete problem [1].

We encoded a network as a tree data structure rather than the classical genetic encoding schemes such as a binary string, which requires a very long binary vector to list all inputs and design decisions. The tree structure eliminates the processing time to encode and decode the Boolean information within the network chromosome. However, using an object-oriented tree forces us to devise novel ways to implement the evolutionary mutation and crossover operators.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

2. RELATED WORK
Weiss and Zeyer [5] describe a case-based reasoning system for the redesign of local area networks (LAN). Their technique takes as inputs a redesign problem and a description of the LAN. Then, a case is selected on the basis of the information included in the redesign problem, where a case contains a corresponding solution (list of actions) for the described redesign problem. The work of Weiss and Zeyer was based on Klein's dissertation [2], where he presented a framework for conflict resolution in a cooperative design. With a number of design actions (whether the actions are applied in serial or in parallel), the possibility of having conflict or creating a new conflict is great. Klein proposed a conflict resolution expert system to (1) detect the occurrence of conflict, and (2) recommend a solution for the detected conflict. This work can resolve any conflict that may occur due to the removal of a bottleneck during the network redesign process. There is no explicit redesign objective in mind when the redesign process was initiated, and hence no optimization.

An early effort by Ravikumar, Pandit, and Mishra [4] describes applying a number of classical techniques, such as the Kernighan-Lin (KL) algorithm for partitioning and the Steiner tree for routing, for the LAN redesign problem. The KL algorithm was used to find the best possible location to add/move clients between the segments of a LAN; A Steiner tree was used to connect clients within the segment, obtaining a minimal cable cost.

3. Network Topology model
We have modeled a network as a 3-level hierarchy of tasks (backbone, site and group), which correspond to three network levels (backbone network, intermediate network, and local area network). The top level task is referred to as the backbone task, which is performed at a number of physical sites, each of which performs a site task (second level task). A site task consists of a number of distinct group tasks (third level task), where each group task comprises a number of distinct client tasks (workstations). The group network (local area network, LAN), the lowest level of network, contains a set of clients. Clustering the clients into LANs and clustering LANs 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 clients need to communicate among themselves efficiently. A typical network problem can be described:

1. A client traffic matrix (CTM) represents the average client-to-client traffic requirements, and
2. A client location table (CLT) represents the physical location of each client within the network.

In this paper, we focus only on the client-to-client load. We have increased CTM load uniformly by 10%, 15%, 20% and 25% and studied the effects on the network topology.

4. EXPERIMENTAL RESULTS

Here we focused on the outcomes of the redesign process, which was implemented in C++ (13,500 lines of codes) on a SUN Blade 150. We used a hypothetical 3-level network topology consisting of four sites, eight groups and 65 clients. The lowest network topology cost, which satisfied all redesign and performance constraints, found by the genetic algorithm, was considered the recommended solution to the problem. Also a proportionate selection scheme [3] was used with the following parameters: population size (PS) = 250, number of generations (NG) = 5000, mutation rate (MR) = 0.05, and crossover rate (CR) = 0.80.

We ran the redesign process with four traffic incremental factors (TIF): 1.1, 1.15, 1.2, and 1.25 which represent uniform traffic increments on the original CTM by 10%, 15%, 20%, and 25% respectively for seven different threshold network delays (TND). TND is 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 average network delay is estimated by summing all delays generated by all network devices within the 3-level network topology, which is based on a network of M/M/1 queues [1]. Figure 1 shows a plot representing the trade-off in the network cost versus network performance while increasing traffic.

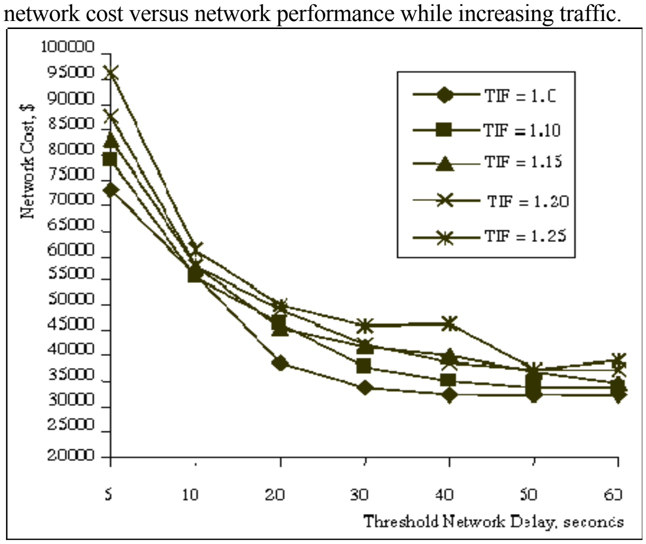

Figure 1. Network redesign cost versus performance.

Each point in the plot represents the network hardware cost of 13 network devices, where 8, 4, and 1 are allocated to groups, sites and backbone respectively. The run-time took less than 5 minutes. The network cost depended on the design decisions made by the genetic algorithm. The largest upward shift occured during TND = 5 seconds, and the cost went from $72,843 to $96,458 when the added traffic load increases from zero to 25%. Figure 2 illustrates the redesign cost versus the saving cost, when TND = 30 seconds and TIF increased by 10%, 15%, 20%, and 25% respectively. Using the original CTM, the cost of 3-level network topology came to $29,789 for network devices and $3,760 for wiring. For the network redesigns, we compared the cost with the original CTM design. The parameters *redesign cost* and *saving cost* represent the total cost of added new network devices and the total saving from keeping old network devices respectively.

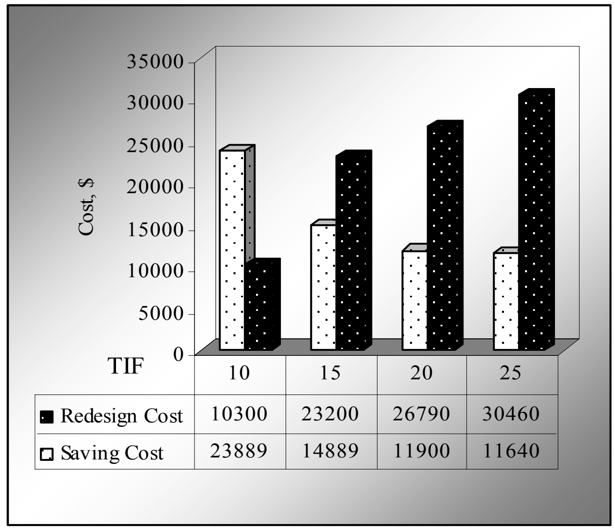

Figure 2. Redesign cost vs. saving cost when TND = 30.0 second.

The saving costs decreased from $23,889 to $11,640 as TIF increased from 10% to 25%. At the same time, the redesign costs increased from $10,300 to $30,460 as TIF increased from 10% to 25%. The wiring costs stayed the same. In all the network topology redesigns, a Router dominated 100% at the backbone task and 94% at the site tasks, which eliminated the usage of gateways, especially when connecting two heterogeneous network devices such as Ethernet to ATM or vice versa.

5. FUTURE WORK

We will improve the redesign process by considering various redesign factors, such as relocating clients, incrementing traffic in non-uniform, and splitting/merging group tasks or site tasks.

REFERENCES

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

[2] Klein, M. *Conflict Resolution in Cooperative Design*. PhD Thesis, University of Illinois at Urbana-Champaign, 1990.

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

[4] Ravikumar, C., Pandit, D. and Mishra, A. Performance-driven Design and Redesign of High-speed Local Area Networks. *In the 1998 Proceedings of 5th International Conference on High Performance Computing.*

[5] Weiss, M. and Zeyer, F. Redesign of Local Area Networks Using Similarity-based Adaptation. *In the 1994 Proceedings of the 10th Conference on Artificial Intelligent for Applications.*