Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

Application of Neural Networks and Machine Learning in Network Design

Hany I. Fahmy, George Develekos, and Christos Douligeris
Dept. of Electrical and Computer Engineering
University of Miami
Coral Gables, FL, 33124-0640
email:{fahmy,george,christos}@ece.miami.edu
Tel (305) 284-3597 Fax (305) 284-4044

Abstract

Communication networks design is becoming increasingly complex, involving making networks more usable, affordable, and reliable. To help reduce this complexity, we have proposed, **END**, an Expert Network Designer for configuring, modeling, simulating, and evaluating large structured computer networks, employing artificial intelligence, knowledge representation and network simulation tools. In this paper, we present a neural network/knowledge acquisition machine-learning approach to improve END's efficiency in solving the network design problem and to extend its scope to acquire new networking technologies, learn new network design techniques, and update the specifications of existing technologies.

1. Introduction

Efficient and optimal network design is necessary to make communication networks usable and affordable. Network design, involves the selection of various circuit parameters, and the selection and interconnecting of various devices to accomplish an organization's operational objectives. A network's configuration can greatly affect its performance and cost. It is, therefore, vital that the best combination of equipment, circuit, and placement of network connections for end-user nodes be made to satisfy an organization's objectives. These objectives may include a multitude of factors other than the prices of the terminals and the networks, such as reliability, response, availability, and seviceability.

Having expert systems handling the network design problem relies on getting and formalizing information not only from theoretical knowledge but from design expertise as well. Unfortunately, the exclusive use of rule-based expert systems to design and optimize computer networks presents several scientific limitations due

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

to two main reasons:

First, such systems may reflect the personal opinions of the network designers whose knowledge has been used in building the systems. Such opinions may be biased or oriented towards specific technologies or products which may not be truly the best. In END, Fahmy&Douligeris [1&2,] solved such problem by utilizing network simulation tools. Once the network environment is obtained from the user, END initially employs the formalized network design experience (expert systems) to eliminate the infeasible solutions for such an environment, recommends suitable ones, explains why such solutions were chosen, and further ranks and evaluates such solutions using artificial intelligence optimization techniques. Then, it uses simulation and performance evaluation to correctly rank the different feasible solutions.

Second, such systems will need a full revision of the design facts/rules from time to time as products evolve and improve. Therefore, learning subsystems should be utilized to give END the capability to acquire the required knowledge to learn about the new technologies, new design techniques, and the updates in the system pre-existing technologies.

Solving the first limitation by the usage of simulation packages in the process of network design and evaluation brought us to another very important motivation behind implementing END's learning subsystem, i.e. the time taken by the network simulation package to simulate the actual network operation and measure its performance. Learning and predicting simulation results, in the case that the expert system recommends a network solution with specifications similar to previously simulated cases, can save time, Rumelhart&Hinton[3].

This paper presents a neural network/knowledge acquisition learning approach to improve the time-efficiency of END's network design problem solver and allow END to learn new emerging network technologies, modern network design techniques, and updated specifications of the existing technologies.

After presenting a general overview of END in section II, section III describes the major research choices and the architecture of the learning subsystem. Section IV presents an interactive learning example to demonstrate the functionality of the learning subsystem, while, in section V we draw conclusions.

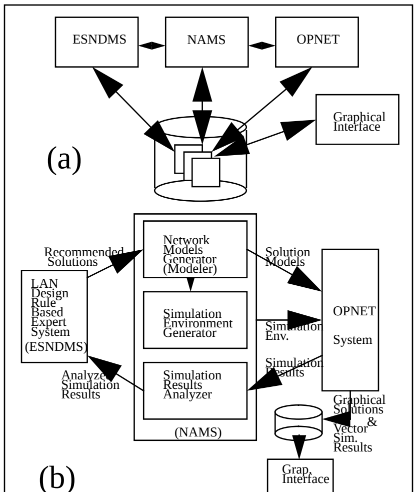


Figure 1 END architecture

2. END System Architecture and Functionality

END is a complex system that relies on extensive cooperation between expert systems, network simulation tools, and programing interfaces. The four main components of END, as shown in Figures 1a and 1b, are:

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

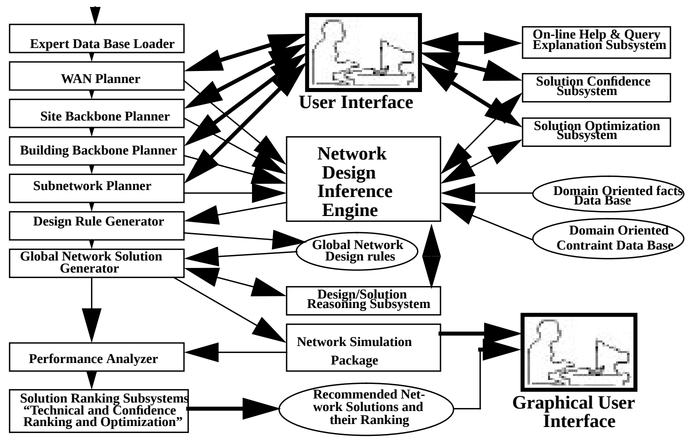


Figure 2 The ESNDMS rule-based expert system

- **A rule-based expert system, ESNDMS,** Fahmy&Douligeris[2], that recommends network feasible solutions according to the user requirements, and evaluates the performance of such solutions after being simulated by a network simulation package;

- **A network simulation package, OPNET**[4], that models and simulates the recommended network solutions;

- **An interface to the network simulation package**, **NAMS**, Fahmy&Douligeris[5], that allows the expert system to communicate with the network simulation package; and

- **A graphical user interface,** that displays a graphical representation of the network feasible solutions recommended by the expert system and the simulation results.

The ESNDMS rule-based expert system, as shown in Figure 2, has a number of planning subsystems (WAN, Site backbone, Building backbone, and Subnetwork) which are responsible for adaptively planning the user queries, and the network design parameters. In the **User Interface** subsystem ESNDMS interacts with the user to obtain a general description of the networking project. The description is obtained through a number of hierarchical questions starting from the highest possible network level, which is the number of network sites, and the WAN interconnectivity between the different sites, passing by the number of buildings in each site, the number of floors in each building. etc., and ending with the number of workstations and servers in the departmental LANs.

The **Expert Planning** subsystems then translate the user project general description into a tree of design guidelines.

A **Network Design Inference Engine** uses this tree of design guidelines to find all the possible solutions for the user project, utilizing a database of domain oriented facts and constraints.

An optional **Solution Optimization** subsystem interacts with the user with a new set of questions, depending on the different solutions obtained in the above section of the system. The answers to these questions are used to filter the solutions to the most suitable solutions for the user environment.

The **Global Network Solution Generator** uses the design rules/facts created by the **Design Rule Generator** for each site, building, and subnetwork to obtain the global network feasible solutions. These solutions are passed to the simulation package interface to be modeled, simulated and evaluated.

A Performance Analyzer receives the simulation results from the simulation package interface (NAMS), in conjunction with the global network solutions from the Global Network Solution Generator, to start classifying the different solutions with respect to their significance in each measured performance parameter.

3. Learning System Architecture

Machine learning is required by END to perform the same design tasks more effectively when encountering the same or similar design cases, Segre[6]. This can be done by keeping the results of previous network simulations or performance measurements as training cases. Such training cases are used for either direct referencing when encountering the exact design cases again, or for predicting the performance measurements of other similar designs.

Learning is also required to assist END to be updated with the recent networking technologies, design techniques, and the changes in the specifications of the system pre-existing technologies. Knowledge acquisition was found to be the only way to train END to learn about the above issues. This is mainly due to the nature of this knowledge being new knowledge that can not be acquired by experience. This type of learning can be done by employing sophisticated user interfaces, natural language fronts, and expert rule builders, to acquire:

- Specifications for the new local area networking technologies (topologies and cabling systems),
- Specifications for the new wide area networking technologies,
- New design techniques for the above new technologies, and
- Updated specifications of the existing technologies.

Two coupled and mutually supportive machine-learning mechanisms characterize our learning model: neural networks and knowledge acquisition.

3.1. Learning Subsystem Architecture

The learning subsystem is divided into six main learning modules as shown in Figure 3:

- ■ Simulation Results Learner/Predictor
- ■ Wide Area Network Learner
- ■ Cabling System Learner
- ■ Topology Learner
- ■ Design Techniques Learner
- ■ Technology Evolution Learner

The Simulation Results Learner/Predictor module has two main learning submodules:

- **Interactive Simulation Results Learning submodule**: This submodule is triggered before each simulation to check the possibility of predicting the simulation

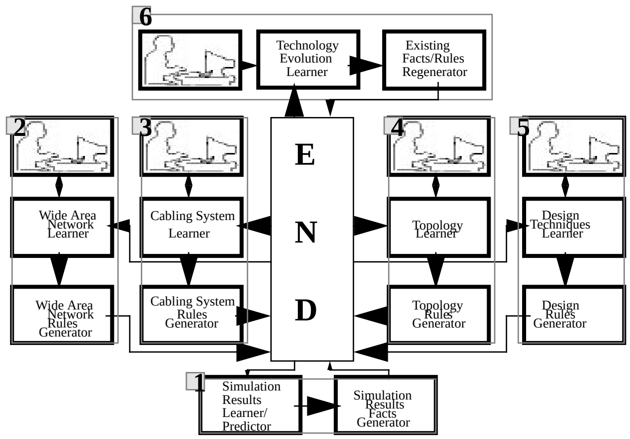


Figure 3 Learning subsystem

results without actually running the simulation. If the prediction can be done in this region of simulation space with less than 10% prediction error, END uses the predicted values. Otherwise, END runs the simulation, after which it acquires the simulation results for this design case as a training case. This is done individually for each subnetwork of the global solution.

One of the advantages of the interactive learning is that the operator can choose to run the system in a purely interactive mode, i.e. no simulations, by simply transforming the learned simulation results of nearest network configuration to the current case. The accuracy of this transformation operation should be reported to the design expert before using it, to decide if the transformation is acceptable or a new simulation should be run for this typical subnetwork.

- **Noninteractive Simulation Results Learning submodule**: Network simulations normally take long time for an interactive system. The noninteractive simulation results learning submodule is designed to be triggered by the END operator during the off-peak hours of the system to generate a large number of subnetwork cases, run their simulations, and learn their performance measurements results for future references.

The simulation results learner/predictor module also has a rule generation submodule which generates the simulation results facts clustered according to the subnetwork properties.

The next five learning modules have an associated user interface submodule to be able to interact with the human expert (knowledge engineer) and obtain the required knowledge. The first four of these five learners are triggered in two cases: 1) by the system operator to instruct the system for a new technology or design technique, and 2) when there is no solution for the a specific design case, the system will prompt the operator if he/she wants to enter a new networking technology or design technique which can give solution for the given design case. The last learner is triggered periodically to update the system knowledge about the pre-existing technologies. Finally, each of these learning modules has a rule generation submodule which will generate/update the expert system facts/rules corresponding to the acquired knowledge.

4. Learning Examples

The following four subsections will describe the main learning tasks carried out by the learning subsystem.

4.1. Learning Simulation Results

Consider END proposing two optional Ethernet and Token Ring solutions for a 6 workstations, 1 server, and imaging database subnetwork. Also consider that the system initially has no trained cases i.e. it can not predict the simulation results for the above two options. The ESNDMS's **Global Network Solution Generator** will choose to run the simulation for the two optional solutions. After the simulations have finished, the **Simulation Results Learner** will generate the following two simulation results:

simulation_result('e id 6 1',37875)
simulation_result('t id 6 1',38990)

These results indicate that for Ethernet (e) and Token Ring (t) subnetworks with 6 workstations, 1 server, and imaging database (id) load characteristics, the throughput is 37875 and 38990 Kbytes respectively in a 20 Sec simulation time.

In any future END runs, if the system proposes a subnetwork with any of the above configurations, the ESNDMS's **Global Network Solution Generator** recognizes that the system has been trained with the results of the subnetwork simulations and it passes the subnetwork as a zero node subnetwork to the simulation package (Normally such subnetworks are parts of the global network solution, therefore such subnetworks are passed to the simulation package containing no nodes in-order not to consume any simulation time and to allow the rest of the network solution to be simulated).

Consider that after using the system for solving a sufficient number of design problems, END proposes a Token Ring solution for a specific subnetwork. In this case, the learning subsystem uses a 2-4-1 Feed-forward Back-propagation neural network to predict the subnetwork throughput from the results of previous simulation runs of similar design cases and passes the subnetwork configuration to the simulation package as a zero node subnetwork.

In-order to achieve a prediction error less than 10%, it has been observed was that the system should be trained with at least 45 training cases for each combination of network protocol, network topology, and network load pattern. If the prediction error is higher than 10% or the system is not sufficiently trained with cases in this region of the simulation space, an actual simulation will be run for the current design case.

After the simulation, the ESNDMS's **Performance Analyzer** obtains the corresponding predicted performance results of the zero nodes subnetworks from the learning subsystem's neural network.

A neural network approach is favored over algebraic approaches for three reasons: 1) the inaccuracy of the algebraic approaches when the subnetworks are op-

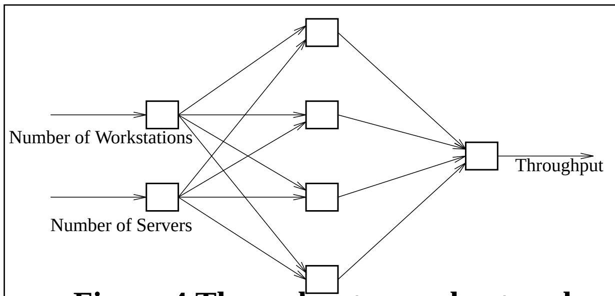


Figure 4 Throughput neural network

erating near their bandwidth saturation, 2) the nature of the subnetwork properties space of being a Multi-dimensional space, and 3) the capability of neural networks to accommodate new learned results and utilizing them in increasing the efficiency of the next prediction processes.

Figure 4 shows the neural network used to compute the throughput versus specifications prediction for the learning subsystem. The number of servers and the number of workstations are the two inputs to the network, while the anticipated throughput, expressed in Kbytes, is the output. The network weights are randomly and uniformly initialized in the -1 to 1 range. Momentum is used to accelerate training. The flat spot elimination technique is implemented to escape local minima and a 1% tolerance between training and supervised output is enforced to prevent over-training. Learning rate and momentum values around 0.5 were found to yield good results. Both inputs and the output are normalized to 1.

In a testing environment, the network was trained with patterns containing about 45 input-output sets. Training proceeds for 20,000 epochs, at which time another 45 testing pattern are presented. The maximum error was found to be less than 10%, and the average error was found to be less than 3%.

At the moment a separate neural network is used for each combination of network protocol, network topology, and load pattern. A future research interest is to use a single neural network for the overall prediction process.

4.2. Learning Networking Technologies

In this subsection, we examine a practical example involving the use of the learning subsystem to learn a new networking technology and its design techniques. The 100VG-AnyLAN is one of the most recent high speed Local Area Network technologies released in the market about a year ago. The 100VG is a star shaped networking topology with 100Mbps data transfer rate, and the node-to-node cabling distance is 100 meters using type 3,4, or 5 unshielded twisted pair cables and 150 meters using shielded twisted pair cables. A shared media access protocol called the demand-priority is used.

The learning subsystem starts by invoking the Topology Learner which issues a number of questions acquiring the above facts from the human expert or the knowledge engineer regarding the 100VG technology. The Topology Learner issues these questions through its user interface module. The answers to these questions are passed to the Topology Rule Generator to generate the domain-oriented facts for this networking topology. The learning subsystem then invokes the Cabling system Learner which investigates other groups of facts from the human expert through its user interface and it passes the answers to its rule generator which generates the domain oriented constraints. Figure 5 shows a listing of this learning session.

```
☞Enter the Name of the New Topology: 100VG
☞What is the topology access method
 1. Ethernet CSMA/CD
2. IBM Token Passing
3. FDDI Token Passing
4. Ethernet CSMA/CD Frame Switching
5. ATM Cell switching
6. Other
Enter your choice: 6
☞Enter the Name of the New Access Method: demand-priority
☞What is the type of topologies media bandwidth [Shared(s) or Dedicated(d)]: s
☞Enter the Max. number of workstations per physical network segment: 100
☞What is the physical layout of the topology [Star(s) , Star-Ring(r) or bus(b)]: s
☞Does the Topology support redundant links [Yes(y) or No(n)]: y
☞The Topology is considered to be preferred in an IBM mainframe environment [Yes(y) or No(n)]: n
☞Enter the Topology Data Transfer Rate in (Mbps): 100
1. Thin coaxial cable (RG-58)
2. Thin coaxial cable (RG-62)
3. Thin coaxial cable (RG-59)
4. Thick coaxial cable (Yellow cable)
5. Unshielded twisted pair cables (not-rated/type 3/type 4)
6. Unshielded twisted pair cables (type 5)
7. shielded twisted pair cables (IBM type 1,2,6)
8. Multi mode fiber optics cables (50/62.5 or 62.5/125 micro meter)
9. Single mode fiber optics cables (category 1)
☞Enter the number corresponding to the supported cables: 5
☞Enter the distance supported by the 100VG topology using unshielded_twisted_pair_cable cabling system: 100
☞Enter the budget for networking a node in 100VG topology using unshielded_twisted_pair_cable cabling system: 225
☞Other Cabling Systems [Yes(y) or No(n)]: y
1. Thin coaxial cable (RG-58)
2. Thin coaxial cable (RG-62)
3. Thin coaxial cable (RG-59)
4. Thick coaxial cable (Yellow cable)
5. Unshielded twisted pair cables (not-rated/type 3/type 4)
6. Unshielded twisted pair cables (type 5)
7. shielded twisted pair cables (IBM type 1,2,6)
8. Multi mode fiber optics cables (50/62.5 or 62.5/125 micro meter)
9. Single mode fiber optics cables (category 1)
☞Enter the number corresponding to the supported cables: 6
☞Enter the distance supported by the 100VG topology using super_unshielded_twisted_pair_cable cabling system: 100
☞Enter the budget for networking a node in 100VG topology using super_unshielded_twisted_pair_cable cabling system: 275
☞Other Cabling Systems [Yes(y) or No(n)]: y
1. Thin coaxial cable (RG-58)
2. Thin coaxial cable (RG-62)
3. Thin coaxial cable (RG-59)
4. Thick coaxial cable (Yellow cable)
5. Unshielded twisted pair cables (not-rated/type 3/type 4)
6. Unshielded twisted pair cables (type 5)
7. shielded twisted pair cables (IBM type 1,2,6)
8. Multi mode fiber optics cables (50/62.5 or 62.5/125 micro meter)
9. Single mode fiber optics cables (category 1)
☞Enter the number corresponding to the supported cables: 7
☞Enter the distance supported by the 100VG topology using shielded_twisted_pair_cablecabling system: 150
☞Enter the budget for networking a node in 100VG topology using shielded_twisted_pair_cablecabling system: 300
☞Other Cabling Systems [Yes(y) or No(n)]: n
```

Figure 5 Learning session of the 100VG technology

4.3. Learning Networking Design Techniques

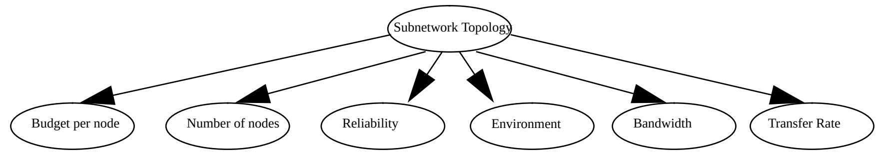
On learning new networking technologies we should be sure that these technologies have the same series of constraints which used to apply to the network components and functional units of the technologies already in the system. We should also be sure that these technologies have the same conceptual problems, choice criteria, and design goals as those of the old technologies.

Learning any new networking topology like the 100VG should be accompanied with verifying the topology choice criteria in an interaction session with the design expert. We also have to check if any new criteria is required to be added to the choice process. The topology choice decision tree was supposed to look like figure 6a, if the same choice criteria of the old topologies are used. Figure 6b describes the verification interaction session with the design expert which leads to a new topology choice decision tree as shown in Figure 6c.

In the interaction session the design expert specifies that the 100VG topology will be chosen according to all the general topology choice criteria except the computing resources vendor criterion “ibm_environment”. The expert also specified an extra choice criterion named “new_design_criteria” for the 100VG topology. The learner then asks the design expert about the type and the value of parameters of this criterion to generate the corresponding built-in domain oriented fact. Finally, the design learner asks how to treat the value entered by the system operator for this criteria in reference to the built-in value for this topology. The expert answer was that for this topology to be chosen the value entered for this criterion in the general design specifications should be smaller than the built-in value.

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

(a) Existing technologies topology design tree

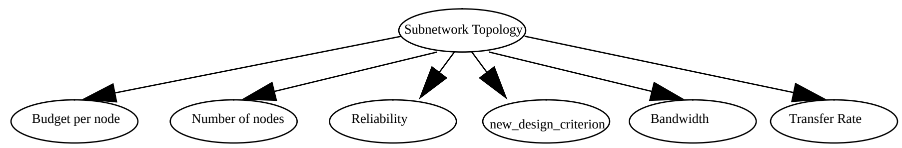


(b) Design Techniques Learning Interaction session

☞The 100VG topology choice depends on the budget for networking a single node [Yes(y) or No(n)]? y
☞The 100VG Topology choice depends on the Max. number of workstation [Yes(y) or No(n)]? y
☞The 100VG Topology choice depends on the cabling system reliability [Yes(y) or No(n)]? y
☞The 100VG Topology choice depends on the computing resources vendor [Yes(y) or No(n)]? n
☞The 100VG Topology choice depends on the type of network bandwidth [Yes(y) or No(n)]? y
☞The 100VG Topology choice depends on the network data transfer rate [Yes(y) or No(n)]? y
☞The 100VG Topology choice depends on any other factors [Yes(y) or No(n)]? y
☞Enter the name of the new 100VG topology choice criterion? new_design_criterion
☞Enter the type of parameters this criterion has [Numeric(n), Yes/No(y), or Fussy Labels (f)]? n
☞Enter the value of new_design_criterion for the 100VG networking topology? 300
☞The user entry for this new design criterion should be greater(g) or smaller(s) than the build in value? s
☞The Topology choice depends on any other factors [Yes(y) or No(n)]? n

Figure 6 Learning session of the 100VG topology design techniques

(c) 100VG topology design tree



4.4. Learning Technology Evolution

☞The ethernet_star topology supports the unshielded_twisted_pair_cable cabling system with a price 250$
Do you want to update the Price? n

☞The ethernet_star topology supports the super_unshielded_twisted_pair_cable cabling system with a price 350$
Do you want to update the Price? y
Enter the new price [350]: 300

☞The ethernet_star topology supports the shielded_twisted_pair_cable cabling system with a price 400$
Do you want to update the Price? n

☞The ethernet_star topology supports the thin_coaxial_cable_RG58 cabling system with a price 400$
Do you want to update the Price? n

☞The ethernet_star topology supports the thin_coaxial_cable_RG59 cabling system with a price 350$
Do you want to update the Price? n

☞The ethernet_star topology supports the thin_coaxial_cable_RG62 cabling system with a price 350$
Do you want to update the Price? n

☞The ethernet_star topology supports the aui_cable cabling system with a price 450$
Do you want to update the Price? n

☞The ethernet_star topology supports the multi_mode_fiber_optics_cable cabling system with a price 600$
Do you want to update the Price? n

☞The ethernet_star topology supports the single_mode_fiber_optics_cable cabling system with a price 650$
Do you want to update the Price? n

☞The ethernet_star topology supports any new cabling system? y
Enter the name of the new cabling system: new_cable
Enter the distance supported by the ethernet_star topology using the new_cable cabling system: 150
Enter the budget for networking a node in ethernet_star topology using the new_cable cabling system: 300

☞The ethernet_star topology supports any new cabling system? n

Figure 7 Technology evolution interaction session

Once a networking technology becomes commercially available, most of if its specifications remain the same except for the cabling system support and pricing. Pricing was the principle motivation for implementing the Technology Evolution Learning module, since it is an important factor in the network design process carried by END. The module is designed to be triggered every period of time to update the system knowledge about the pricing and the extensions in the cabling systems support of the pre-existing technologies. An interaction session for updating the pricing and cabling system support knowledge of the ethernet_star networking topology is shown in Figure 7. In this session, the operator updates the price of the ethernet_star topology support for the super_unshielded_twisted pair cabling system from $350 to $300. Also the operator adds the topology support for a new cabling system named new_cable with 150 meters node distance and $300 of budget to network a single node. After the interaction session the learning module will regenerate the updated knowledge-base for the ethernet_star networking technology through it facts generator.

5. Conclusions

Neural networks and machine-learning play an important role in communication networks design decision making. The integration of expert systems techniques, neural networks, traditional machine-learning methods, and network evaluation and optimization using simulation packages provides a flexible framework for decision making during the various stages of design. One of the features of the communication networks field is the fact that it is continuously changing and advancing. This paper presented an introduction to the use of neural networks and machine-learning for enhancing the functionality of automated network design systems. We employed neural networks to train the expert system to use its previous experience to improve the design life-cycle, and machine-learning to acquire new networking technologies. This methodology enhances interactive design analysis, and increases the expert system scope, efficiency, applicability and updating ability.

References

1. Hany I. Fahmy and Christos Douligeris, *END: An Expert System Designer*, IEEE Network Magazine, November/December 1995, Vol. 9, No. 6, pp. 18-27.
2. Hany I. Fahmy and Christos Douligeris, *ESNDMS: An Expert System for Network Design Modeling and Simulation*, submitted to IEEE Expert Magazine.
3. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, *Learning Internal Representations, Parallel Distributed Processing: Explorations in the Mircostructure of Cognition*, Vol. 1, D.E. Rumelhart and J.L. McClelland, eds., MIT Press, Cambridge, Mass., 1986,pp. 318-362.
4. *OPNET Network Modeler, OPNET Modeler Manuals*, MIL3, Inc., Washington DC, 1994.
5. *OPNET External Model Access, OPNET External Interfaces Manual*, MIL3, Inc., Washington DC, 1994.
6. Alberto Maria Segre, *Applications of Machine Learning*, IEEE Expert, Vol 6, June 1992, pp. 30-31.