Indian Journal of Science and Technology, *Vol 9(46), DOI: 10.17485/ijst/2016/v9i46/107523, December 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Network Synthesis: Using Genetic Algorithms for Network Synthesis

Vladimir Sergeyevich Ziyautdinov, Dmitry Mikhaylovich Skudnev* and Mikhail Yuryevich Smirnov

Lipetsk State Pedagogical P. Semenov-Tyan-Shansky University, Lipetsk, Russia; ur_pror@lspu.lipetsk.ru; scoudnev@lspu.lipetsk.ru; iemitn@lspu.lipetsk.ru

Abstract

Objectives: The article deals with problem of creating the new systems of the Ethernet network optimal structure, considering a possibility to adjust such system in compliance with system administrator specification. **Method:** The main tasks of the research have been solved using as follows: graphs theory, queuing theory; large control systems theory; simulation modeling theory; genetic algorithms theory, structural and object-oriented programming methodology. **Findings:** In the present article an important scientific and technical problem associated with synthesis of the Ethernet network optimal structure is being solved. The following new research findings have been obtained: the modification of a genetic algorithm for topology synthesis of the Ethernet local network has been offered, in which compared to the analogues known, a simulation system is used as a utility function which enables to consider the user traffic effect on the Ethernet network characteristics at the evaluation of this structure variant. The methods of a parametric and structural optimization of the Ethernet local network have been developed. The unique methods of optimization of a structured cabling system of the Ethernet corporate network have been developed. **Improvements:** The results obtained are rather universal which enables to use them in different fields of the Ethernet network application: For quality improvement, reduction of value and time required for preparation of design solutions based on the Ethernet networks; in MetroEthernet; in Industrial Ethernet; in the networks different from the Ethernet standards for example, in virtual private networks or in the networks based on multi-protocol label switching MPLS.

Keywords: Genetic Algorithm, Local Network Synthesis, Models

1. Introduction

For the time being the local computer networks have become widely used in different fields of human activity. The Ethernet network has especially gained a widespread use. The architecture of the Ethernet network is determined by Standard IEEE 802.3 and is based on the shared media network function protocol CSMA/CD. There have been obtained several particular results; however, the task of a structural optimization of the Ethernet local computer network has not been solved yet on the whole. In the process of finding of an optimal structure of a local network, a check of the network quality indices (set-up parameters and characteristics such as response time, protocol peculiarities, packet delays, packet loss percentage, collision effect on the network characteristics, etc.) should be performed constantly. For this reason the problem of creating the new systems of the Ethernet network optimal structure, considering a possibility to adjust such system in compliance with system administrator specification, is quite relevant.

The efficiency of functioning of the biological system elements arouses a keen interest in the use of the principles of a biological evolution for the tasks associated with optimization of the systems being of a practical importance for different fields[1].

Compared to the standard optimization methods, the methods of evolutionary and genetic calculations have the following peculiarities: parallel search, random mutation and recombination of good solutions been already

**Author for correspondence*

found. They work well as a simple heuristic optimization method of multivariate and ill-defined functions.

Of these methods the genetic algorithms have gained a widespread use[1]. Genetic algorithm is a computer model of the evolution of artificial "specimen" population. Each specimen is characterized by its own chromosome S_k, a chromosome is interpreted as specimen "genome" and determines the fitness of a specimen $f(S_k);\ k = 1\dots n$; n – number of specimens in the population. Chromosome is a character string $Sk = (S_{k1}, S_{k2} \dots S_{kN})$, N-string length. The characters are interpreted as a "gene" of a specimen located in S chromosome. The task of the algorithm is to maximize the function (S_k)).

Evolution consists of the sequence of generations. The specimens with the high fitness values are selected for each generation. The chromosomes of the specimens selected are recombined and subjected to mutation. Formally, the scheme of a genetic algorithm can be represented in the following way (population of t-generation is defined as $\{S_k(t)\}$):

A0. To create a random initial population $\{S_k(0)\}$.

A1.To calculate the fitness $f(S_t)$ of each specimen S_k of the population $\{S_k(t)\}$.

A2.To form the population of the next generation$\{S_k(t+1)\}$ selecting the specimens S_k based on their fitness $f(S_k)$ and applying genetic operators to the specimens selected.

A3.To repeat steps Al and A2 for $t = 0; 1; 2\dots$ until the condition of evolution search completion is fulfilled.

Local computer network designers come across an important task associated with synthesis of the network with an optimal structure. In this condition there might be required to optimize the network structure or based on the criterion of maximum throughput or on the criterion of minimum cost. The task is unconventional since there are a huge number of methods of creating the network for one and the same initial data. Normally at designing the network structure, the designers take into consideration their own experience and intuition and this does not always lead to the best results. For this reason, it is relevant to create such systems of automated synthesis of the optimal network structure which will make possible for a designer to just specify the desired parameters and at the output obtain the best structure for data specified.

The term "synthesis" of a technical object in the broad sense of the word is close to the term "designing" with respect to the meaning. The task associated with synthesis of a technical object is to obtain a design solution in the form of the description of designed object based on the specified functional use or based on the principles of its functioning.

At object synthesis it is necessary to evaluate the numerous methods of the problem solution. At designing the Ethernet local computer networks, arises a problem associated with selection of an optimal scheme of the connection of switching nodes with workstations at optimization of a throughput (or at minimization of the costs). For solving the problem of network synthesis it is necessary to predetermine (preset) an initial permissible set of elements. At designing the topology of a local computer network and traffic distribution in this network, such "starting" set is as follows:

Set $W^{(N)}$ network nodes/vertices (blocks), N – number of nodes;

Traffic – is set up in matrix form T where T_{ij} – traffic between nodes i and j;

Distance matrix D – between pair of network nodes where D_{ij} – distance between nodes i and j;

Cost of switching equipment which is determined by the type of the equipment depending on the traffic of a local network.

At designing a local network, it is expedient to select the following criteria as the criteria of optimality:

- Minimization of costs for creating a local computer network;
- Maximization of throughput.

In order to solve a defined problem, it will be required to create a matrix of connections between blocks – $R,\quad W_l^k = R\left(W_j^i\right)$. The total cost of a local computer network consists of two parts, namely the cost of switching equipment and cost of cable installation: $C = C_{equip} + C_{cable}$. The cost of the installation of cables between any two nodes is determined based on connection matrix[1].

The problem can be solved in two stages. The first stage is revealing the connections between network nodes, the second one is shaping the traffic flow based on the connections between network components for the purpose of determining a full load.

At the first stage a graph $G=\{W, R\}$ is studied where R – set of graph connections; W – set of blocks (units) of a local network. The framework is selected in the form of a spanning tree to which the other connections can be added further. Algorithm of shaping the tree is based on a sequential exclusion of vertices from a subset D_2 and their inclusion in a subset D_1. At solving this sub-problem it is necessary to consider that there are two types of equipment in a local network: switching equipment (switching unit and repeaters) and workstations. The relevant connection is made based on the equipment type.

The connections of the first type should not be cyclic connections. As typical structures of block connection we will take the following structures: bus, two-level star, three-level star (Figure1).

<insert figure 1 here>

At implementation of the connections of the second type, it is necessary to take into consideration that firstly, they should not be connected with each other and secondly, the number of the channels connected should always be equal to one.

The workstations are the main source of the traffic in the network transmitted via switching units. The load can be divided into two types: incoming load and switching load. The value of a switching load is equal to the sum of the values of the traffic flow coming to a switching node. All traffic flows going through switching nodes can be divided into three types:

- Traffic of a work group – traffic coming from the workstations connected to a switching unit ($\tau_{ðñ}$);
- Downstream traffic – traffic coming to switching units of a lower level (τ_{sw1});
- Upstream traffic – traffic coming to switching units of a higher level (τ_{sw2}).

$$T_{ðñj} = \tau_{sw} + \tau_{sw1} + \tau \quad {}_{2}$$

For the purpose of optimization of a local network it is necessary to find an optimal distribution of loads on switching units in order to avoid overloading or down time of the equipment.

Thus, it is necessary to reformat a connection matrix (by creating the new connections and deleting the portion of the old ones) so that a new structure of the network will provide the best quality indices of a local network in line with specified objective function.

2. Chromosome Formation

The size of chromosomes depends on the number of points of a target space, i.e. from the number of computers in the network and from the number of switching units used, and generally is represented in the form of Table of element interfacing (Table 1).

Table 1. Example of connection of switching units and computers

	SW1	SW2	…	SWm
PC1	1	0	…	0
PC2	0	0	…	1
…	…	…	…	…
PCn		1	…	
SW1	0	1	…	1
SW2	1	0	…	0
…	…	…	…	…
SWm	1	0	…	0

At solving the task of chromosome formation there was made a conclusion that the most expedient thing was to break it down into two subtasks: 1) connection of the computers to switching units; 2) finding the connections of switching units with each other.

For this purpose, it is necessary to break down all sets of blocks W into two subsets: P– set of blocks-workstations; S – set of blocks – switching units.

$$P \cup S = W,\ P \cap S = 0,$$

$$P_l^{(k)} = R\left(S_j^{(i)}\right) \tag{1}$$

$$S_l^{(k)} = R\left(S_j^{(i)}\right) \tag{2}$$

At formation of a chromosome, a connection matrix is built up. When using the interfacing operator R for connecting the computers with switching units (Formula 1) and switching units between each other (Formula 2) we will obtain a reflection of an expanded network structure[1].

Since the computers have only one port and consequently one contact, then in Formula 1 index l can be substituted with 1. From the other side, it does not matter which ports of switching units the computers will be connected to and then, Formula (1) can be written as follows: $P^{(k)} = R\left(S^{(i)}\right)$. It also does not matter which numbers of the contacts are used for connecting the switching units between each other, then in this case Formula 2 will be written as follows: $S^{(k)} = R\left(S^{(i)}\right)$. However, it should be taken into consideration that the number of ports of switching units is limited by value $K(S^k)$.

When solving the first problem, it was cleared up that the number of chromosomes in a genotype is equal to the number of switching units in the network considered. This results in tedious calculations and complicated formation of the initial population. For this reason, all chromosome chain rolls up into one chromosome using a binary encoding.

$$PC \in \{\, 0, 1, \ldots, L \,\},$$

where $L = \left(\sum_{i=1}^{n} SW\right) - 1$ – switching unit pointer (Figure 2).

<insert figure 2 here>

Now, instead of a binary encoding of a chromosome in expanded form where 1 points to the presence of a connection and 0 points to the absence of a connection, we obtain a new type of a chromosome where a connection between a switching unit and a computer is set up by a decimal number – pointer to the number of a switching unit. The pointer to a switching unit has an initial value 0. Such approach simplifies the formation of a parental generation of chromosomes. When forming the chromosomes, it is necessary to consider the following restrictions:

Computer can be connected to only one switching unit

$$\sum_{j=1}^{m} c_{i,j} = 1,$$

where m – number of switching units;

2) The number of computers connected to a switching unit cannot exceed the number of ports available

$$\sum_{j=1}^{n} c_{i,j} \le K_{port},$$

where n – number of computers[1].

As an example, Table 2 shows the sets of viable chromosomes for a local computer network which comprises 50 workstations ($K_{\delta\tilde{n}} = 50$), 7 switching units $\left(K_{sw} = 7\right)$ with 16 ports ($K_{port} = 16$).

When solving the second problem it is necessary to consider that at connection of the switching units with each other, the cyclic connections should not occur (a simple example is given in Figure 3).

<insert figure 3 here>

For the aforementioned example we will make a table of the connection of switching units with each other (Table 3). In this case the topology of the network is set up by connection matrix of switching units.

The Table obtained is symmetric with respect to the main diagonal, correspondently all the connections located below the main diagonal can be ignored. For the formation of a chromosome we will take matrix elements located above the main diagonal and form a single-dimension array. As a result we will obtain a viable chromosome without cyclic connections (Figure 4). The length of a chromosome is equal to the number of matrix elements above the main diagonal and will be as follows: $l_i = n_{sw}\left(n_{sw} - 1\right)/2$ where n_{sw} – number of switching units in the network.

Different developed genetic algorithms of chromosome mutation and crossing over can be applied to a chromosome obtained[1].

<insert figure 4 here>

3. Crossover Formation

Crossover, which is sometimes called a crossing over, fulfills the function of transferring the gene sites from parents to descendants. Crossing over operations can be as single-point as well as multi-point ones. At simple (sin-

Table 2. Set of 10 chromosomes

No. of chromosome variant	**Structure of viable chromosomes in a decimal form**
1	41241342240230603260101162564405264634340613101023
2	04305031353433311460616653046263665662541114002510
3	24211043424264530564544415260033052510335350560544
4	35304023006145122155131203541504206600305643564442
5	42202015555044433125510221056040033506412032454432
6	12662044446456165252001462252561640101665220535445
7	02401422550303210663456352105456516205026332542100
8	33526512121422334155161123062601220460501102354113
9	10150525433642663443466125343041321105454103644064
10	01463134140012536051265665041666366144246622064134

Table 3. Table of connection of 7 switching units

	SW1	**SW2**	**SW3**	**SW4**	**SW5**	**SW6**	**SW7**
SW1	0	1	0	0	1	1	0
SW2	1	0	1	1	0	0	0
SW3	0	1	0	0	0	0	1
SW4	0	1	0	0	0	0	0
SW5	1	0	0	0	0	0	0
SW6	1	0	0	0	0	0	0
SW7	0	0	1	0	0	0	0

gle-point) crossover the parental chromosomes break in some randomly selected position being however the same for both parents. The parents exchange the relevant segments and obtain two genotypes. In two-point cross over, two break points are selected and the parental chromosomes exchange the segment located between them.

At the first stage we select a pair of chromosomes from the parental population and create a temporary population of chromosomes required for selection. This population is dedicated for using a genetic operator for the purpose of formation of a new population of descendants. The parental chromosomes randomly unite in pairs with probability P_c. Subsequently, a possible break or crossing over point is randomly selected for each pair of chromosomes. If a chromosome of each parent consists of genes L, then a crossing over point l_k is selected from the interval $[1, P\text{-}1]$ (Figure 5).

As a result of crossing over of a pair of the parental chromosomes A and B, a pair of descendants C and D form. The chromosome of a descendant C consists of the alleles from locus 1 to locus l_k including the genes of the parent A as well as of the alleles of loci from l_k+1 to L including the genes of the second parent. The chromosome of a descendant D consists of the alleles from locus 1 to locus l_k including the genes of the parent B as well as of the alleles of loci from l_k+1 to L including the genes of the parent A. The action of a crossover operator is shown in Figure 5. Of two descendants obtained only one is selected with probability of 0.5. Subsequently, all the descendants selected are placed in a general population to which a selection operator is applied to.

4. Fitness Function

At finding an optimal topology of a large-scale and multifunctional Ethernet network, it is challenging to create a fitness function of a genetic algorithm in the form of the mathematical expressions considering a network capacity. Therefore, we will use a program for simulation modeling of the processes going on in the Ethernet network. Thus, the values of fitness function will be found using the results of simulation runs.

The following data is transferred to simulation modeling system:

- Number of computers in the network;
- Number of switching units in the network;
- Information about each unit in the network (for a workstation – size of modulated frames, distribution law, etc., for a switching unit – number of ports, size of port buffer, frame processing delay, etc.);
- Connections between units, etc.

Network units load, queue status, information about request status, etc. are the output data of simulation modeling system.

However, it should be taken into consideration that at using the simulation programs, there will be obtained different values of one and the same output parameters in different program runs. This occurs due to a stochasticity of a modeling process since in each new run the random data generators used in a program are started with new constants. Thus, the more runs are performed, the more accurate are the results since the law of large numbers and central limit theorem conditions are executed properly. There was implemented a high speed (simple) method of a simulation system which enables to perform a big number of runs and obtain the values of fitness-function with a required accuracy for the purpose of fitness-function evaluation. When the first populations are used, a genetic algorithm in a general case is located far from an extreme point. Consequently, some calculation error of fitness-function will not significantly affect the quality of selection of the new chromosomes. At getting closer to an extreme point, the number of simulation runs should be increased. For determining the required number of simulation runs there can be used well-developed methods[2].

When designing a large-scale and multifunctional Ethernet corporate network with a big number of switching units and routers, a minimax criterion could be of help. In these conditions it is necessary to minimize a load λ_{max} of high-usage trunks. Minimization of the value λ_{max} is connected with determining the number of logical tracks for providing the application requests from the traffic flow $w^{(i)} = \langle x^{(i)}, y^{(i)} \rangle$. The objective function is described by the expression $\min\left[\max\left(\sum \lambda_{i,j}\right)\right]$ where $\lambda_{i,j}$ – traffic between nodes i and j.

For network optimization there can be used the second criterion, then one more objective function is used for minimizing the cost of the required technical facilities (i.e. minimizing the cost of network creating[1]).

Simulation is performed based on discrete event-driven approach with implementation of the event calendar. The main elements of a modeling system, namely workstation, communication channel and switching unit communicate with each other transferring each other the packet (frame) sequences. Packet sequence rate is random, with a predetermined distributive law. The record data flows and read data flows are divided. There is one record data flow and one read data flow in each element (block). The record flow sends the frames to the adjacent elements and the read flow performs the function of reading the frames. The aforementioned flows are controlled by a global manager. Management data is placed in a global manager. This data activates the element flows in the required sequence. The above mentioned data determines the time of the event simulation start with regard to the frame processing. The workstation for sending the frame at the time t places a relevant object in a global manager. Global manager sequentially processes the objects in such a way so that at the time t a control will be delegated to the record flow of the workstation. Similarly, to the aforementioned, a total interaction of all elements of a modeling system takes place.

Transfer of the information regarding the necessity of reading data can be made not only via a global manager but also using the element method. Since the elements of the system are connected with each other through the pointers, then one element can initiate reading the sent frame by the other element. But in these conditions a global manager is not used and the event mentioned below is implemented. The frame is sent directly to the record flow of a relevant recording element. Thus, a consequence of the events, which follow in succession and do not require a preliminary distribution in time, is implemented. For example, at recording the frame by the workstation it is not required to place in a global manager the event concerning reading this frame by a communication channel since reading is performed straight after the frame transfer.

We will study the algorithm of transferring the frame by the workstation. The workstation in a general case generates transit packets at the time intervals according to the selected law of random variable distribution. The obtained sampled time value associated with new frame

Table 4. Percentage completion of fit-function with cache

Number of computers	Number of chromosomes	Number of completions of fit- function	Number of fit-function values taken from cache	Total fit-function values	% of fit-function completion
50 PC	100	13,010	61,990	75,000	17.35%
	80	10,585	49,295	59,880	17.68%
	60	5,191	38,069	43,260	12.00%
	40	3,839	25,801	29,640	12.95%
	20	1,840	12,680	14,520	12.67%
100 PC	100	25,125	136,025	161,150	15.59%
	80	18,833	109,858	128,691	14.63%
	60	13,496	82,750	96,246	14.02%
	40	83,38	55,322	63,660	13.10%
	20	3,546	27,736	31,282	11.34%
150 PC	100	34,462	201,542	236,004	14.60%
	80	26,451	162,186	188,637	14.02%
	60	18,329	122,669	140,998	13.00%
	40	11,497	81,863	93,360	12.31%
	20	49,21	40,808	45,729	10.76%
200 PC	100	46,967	276,504	323,471	14.52%
	80	34,146	224,317	258,463	13.21%
	60	24,163	168,879	193,042	12.52%
	40	15,492	112,456	127,948	12.11%
	20	6,675	56,054	62,729	10.64%

generation is placed in a global object manager which initiates transferring this frame. At this predetermined time a global manager will read this frame and delegate control to the record flow of the workstation. The record flow will record the frame on its output and send the information to a communication channel about the initialization of the event regarding reading this frame. The record flow of the workstation is waiting for response from a communication channel.

During program execution several types of the output files are generated: network configuration files, event log files in the network elements and modeling files.

Simulation modeling program with a genetic algorithm operates in automatic background mode and shows the results of finding of only the best solution. At using a genetic algorithm together with a simulation modeling program, it is possible to automate the process of network creation. In a genetic algorithm based on a relevant chromosome is generated the file which is downloaded into a simulation modeling program. The file is a program configuration file. Then the file is loaded into a simulation modeling program. The results obtained in the process of the program simulation modeling are exported to a text file. Further the program, which implements a genetic algorithm, reads data obtained and uses it for calculating the value of fitness-function.

Fitness-function determines the relation between maximum load of the network device and minimum value.

5. Optimization of Genetic Algorithm Speed

In the process of genetic algorithm execution not one but a set of solutions is implemented. Such solutions are predefined based on a probability principle. Consequently, approaching to an extreme point occurs in several ways which leads to the use of one and the same chromosomes. Simulation modeling system, which calculates a solution optimality spending a certain time on this, is used as a fit-function. The larger and more complicated the network is, the longer it will take to evaluate the network capacity. In this connection it has been offered to use a special buffer (cache) for storage of a certain set of chromosomes and values of the objective function. As the calculations have shown, the use of a buffer enables to reduce the process of calculation of the fit-function values by 80…90%. Some results of a comparative analysis of a buffer used are given in Table 4.

Using a genetic algorithm, it is possible to find an optimal network structure in which a load on all switching units will be uniform.

6. Conclusion

The results of the research have laid the foundation for the programs developed using a programming environment Delphi and a programming language C++. Graphic programming interfaces enable to enter the next data: number of computers in the network, number of switching units, the Ethernet local network and genetic algorithm specifications.

Based on a modeling analysis, one can make a conclusion about the efficiency of a practical use of genetic algorithms for the purpose of finding the optimal characteristics of the Ethernet local network. Using this program, a system administrator can balance a load in better way and determine a bandwidth margin of the Ethernet channels for a set of information flows.

The results of the research are as follows:

1. A new modification of a genetic algorithm for synthesis of a large-scale and full-service Ethernet corporate network has been offered, in which, compared to the analogues known, the results of operation of high speed variant of the network simulation system for inertialess operation mode of switching units are used in the utility function with deactivating the function of displaying a graphic information on computer screen. This enables to take into consideration the user traffic effect on the Ethernet characteristics at evaluation of this structure variant.

2. A visual system of a simulation modeling has been created which implements the functions of a parametric and structural optimization of a corporate network created using the fragments of the Ethernet, FastEthernet, GigabitEthernet, 10 GigabitEthernet and MetroEthernet based on switching technology, as compared to the prototypes known.

3. The methods of a parametric and structural optimization of the Ethernet local network and structured cabling system of the Ethernet corporate network have been developed.

5. References

1. Blagodarov AV, Pylkin AN, Skudnev DM, Shibanov AP. Modeling and synthesis of the optimal Ethernet network structure. Moscow: Hot Line Telecom; 2014.
2. Pilinsky M, Rutkovskaya D, Rutkovsky L. Neural networks, genetic algorithms and fuzzy systems: translated from Polish by Rudinsky I.D. Moscow: Hot Line Telecom; 2013.