Annales UMCS Informatica AI 5 (2006) 323-332

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Alternative methods of designing multi channel optical communication systems

Miroslaw Hajder, Miroslaw Mazurek*

Department of Distributed Systems, Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland

Abstract

This article attempts to discuss the problems in building new topologies utilizing a few alternative methods of designing systems like embedding, d-optimum topology. We propose to utilize Cartesian Product to describe graphs as a right mathematical solution to adjacency matrix. We propose methodology of building multi channel systems using a few original algorithms.

1. Introduction

In the context of parallel and distributed computation, the problem of embedding one interconnection network into another is the fundamental importance. Currently available computation or communication tasks utilized in parallel and distributed systems show that in the row of an accidental capacity, computing power of currently used computers is not sufficient. Constant growth of computing complexity in the computer systems brings growth of systems capacity. At the beginning this growth was accomplished by enhancement to parameters of base elements utilized to build computing structures. In the process of designing topologies, besides utilizing base topologies, meaning topologies described by particular graphs with specific abilities such as: small degree, small diameter, symmetrical or a large number of nodes, should be considered embedding as a right tool to build new topologies. To achieve this goal we could use logical operations utilized on the matrix of corresponding graphs in particular the Cartesian Product. This new developed original topologies are characterized with predictable parameters described in base topologies. As topologies there could be used those with opposite parameters (for example bus and hypercube).

Currently conducted research showed that in building new multi channel communication systems a special usage has regular and symmetric topologies

*Corresponding authors: *e-mail addresses*: {mhajder; mirekmaz}@prz.rzeszów.pl

[1]. They secure minimal computation complexity of the designing process and liveness and also large scalability assuring building networks with an unlimited number of nodes.

2. The definition of the designing issue

Assuming that, there is N number of points described by the Cartesian coordinates (x_i,y_i), where $i=1,2,...N$. These points should be related among themselves. Those connections are defined with the loading matrix $\|\lambda_{ij}\|$, where $i,j=1,2,...,N$, $\lambda_{ii}=0$.The defined points should be a basis for building a two-level network, using minimum cost. This means that a certain L number of points should be chosen out of N and they should be assigned a function of communication nodes. Next, each of the remaining N-L nodes should be assigned a complete network built among the commutation nodes. The commutation node together with the assigned subscribers will be further called "a cluster", and the commutation node itself will be sometimes called "a cluster's centre". The subscriber that occurs in the place where the commutation node is located is assigned exclusively to that node. The cost of this assigning equals zero.

The cost of building a W network is defined by the following expression:

$$W = S_1 + S_2 + S_3, \tag{1}$$

where S_1 – the cost of subscriber's comunication channels, S_2 – the cost of commutation nodes, S_3 – the cost of lines connecting commutation nodes.

In segments, the constant function of commutation nodes cost depending on a Λ capacity has the following form:

$$C_{WK}(\Lambda) = \sum_{i=1}^{m} \left(U(\Lambda - \Lambda_i) - U(\Lambda - \Lambda_i) \right) d_i, \tag{2}$$

where m is the number of constancy segments, Λ_i – the segments limits of the function constancy for $i=1,2...,m+1$,

$$U(x) = \begin{cases} 0, & x \le 0 \\ 1, & x > 0 \end{cases},$$

d_i – the value of function cost in the constancy interval $(\Lambda_i, \Lambda_{i+1})$.

The function of the connection line depending on a stream λ transmitted along that line and its lenghts r, can be described in following way:

$$C_{KL} = \begin{cases} \left\lceil \frac{\lambda}{\lambda_0} \right\rceil \left(a_1 e^{-c_1 r} + b_1 \right) & r < r_0, \\ \left\lceil \frac{\lambda}{\lambda_0} \right\rceil \left(a_2 r + b_2 \right) & r \ge r_0. \end{cases} \tag{3}$$

The multiplier $\left\lceil \frac{\lambda}{\lambda_0} \right\rceil$ in the function of the communication line value has got the following sense: λ_0 – the load value to which one communication channel is sufficient for operating. If the load $\lambda > \lambda_0$, then the expression $\left\lceil \frac{\lambda}{\lambda_0} \right\rceil$ is assessment from the top of the number of channels, necessary to send some information with a specified quality.

We will further assume that if a capacity of the commutation node and a capacity of the used communication lines exceed the value of the operated load, then the load operation is executed with a required quality. In other words, the probabilistic and time requirements are satisfied.

Assuming that the flow generated externally between points *i* and *j* creates a stationary Poisson's stream with an average value λ_{ij}, $i, j=1,2,...N$, $\lambda_{ii} = 0$, in the commutation node, there is a memory of an unlimited size, announcements are operated in the order of appearing and the movement is directed through the shortest path, designated on the basis of branches creating a route. While the operating, the priorities are not used. We will analyze a network with packet commutation. The interval movement of the network will not be taken into consideration. We also assume that the network will work without fault.

The design issue defined above is non-linear, discreet and combinatoric. There is a considerable number of algorithms intended to solve. Precise methods are based on branches and limit methods. Unfortunately, their application is limited because of a very high computation complexity. In practice, those methods are applied while designing network in a size of up to tens nodes. Nowadays, heuristic methods are the most commonly applied. A new approach to optimizing networks was created on their basis. Most often, heuristic methods use an iteration algorithm of branches exchanging.

A basic approach to a local optimum solution to a given assignment is based on a division into sub-assignments (defining the number of communication nodes, their distribution; defining the subscribents assigned to the nodes; defining the internode relations) and solving them sequentially.

The *L* number of commutation nodes is mostly assigned and chosen out of computed section (in the case of some assumptions about subscribents distribution and area as well as loading characteristics) on the basis of analytical assessment or that obtained from mathematical patterns [2].

2.1. Definitions and designations

For the *S* network structure, with an assumption that $N>2$, the set of communication nodes will be designated as $D_S = \{i:\ \deg(i) > 1,\ i = 1,2,...,N\}$, $|D_S| = L$, the set of subscribers (excluding subscribers located directly in the

communication node) $A_S = \{j:\ \deg(j) > 1,\ \ j = 1,2,...,N\}$, $A_S = N - L$, where *deg(j)* is the degree of the *j* node in a graph that demonstrates the *S* network structure, Z_i is a set of subscribers joined to the communication node $i \in D_S$.

The P_{ij} potentials for the pairs of subscribers *i* and *j*, will be designated in the following way: $P_{ij} = \lambda_{ij} / r_{ij}^2$, $i,j=1,2,...,N,\ i \neq j$. The load generated by the subscriber *j* equals $\Lambda_j^+ = \sum_{i=1}^{N} \lambda_{ij}$. The load assigned for the subscriber *j* is designated as $\Lambda_j^- = \sum_{i=1}^{N} \lambda_{ij}$. The cluster's potential v_l is designated by the following formula:

$$v_l = \sum_{j \in Z_i} \frac{\Lambda_j^+ + \Lambda_j^-}{r_{jl}^2}, \qquad l \in D_s . \tag{4}$$

In the class of searched network structures we will specify the following surrounding systems. Let a certain *S* network structure be assigned.

A surrounding in relation to a connection line. We will use the definition: "a surrounding in relation to a connection line", to call all the network structures, which differ from *S* in joining only one element.

A surrounding in relation to a cluster's centre. The term "surrounding of the *S* structure in relation to a cluster's centre" will be used to call all the network structure which have the same framework of clusters, but differ from *S* in one of cluster centres.

A surrounding in relation to cluster's formation. The term "surrounding in relation to cluster's formation of *S* structure", will be used to call all the S_1 structures, for which the following dependences are satisfied: $\forall i \in D_{S_1}, Z_i^S \subseteq Z_i^{S_1}$. The surrounding of *S* structure of a given type is created by all the S_1 structures which have got one cluster fewer than *S* structure. The centres of the clusters in S_1 are the same points as in *S*. If any point is in the centre of both structures, then all the subscribers joined to this point in *S* will be also joined in S_1.

A surrounding in relation to a centre transmission. The term "surrounding of the *S* network structure in relation to a centre transmission" will be used to all the structures which are received from *S* in the way, that a subscriber from one of the clusters becomes a centre of some other cluster.

3. Algorithms specification

3.1. Algorithm 1

In the first step of algorithm, we build a two-level centralized network with minimum cost in a given structure class if the total load of a communication node is bigger than the Λ_m capacity of the most effective mechanism. We assign

a certain sufficiently big value to cost of a given commutation node (for example 10^{10} - i, where I is a number of algorithm iteration) and network cost is described exclusively as a sum of the commutation nodes cost.

A structure obtained in this way is then modified in the following way: one of the subscribers become a centre of a new cluster, and some part of subscribers is transmitted to this cluster from the other clusters.

If introducing an additional commutation node causes improvement of network costs, then that structure is chosen as the next one which brought the best results. In relation to a given structure, the procedure of introducing an additional centre is applied cyclically. So far, adding a new cluster's centre has not brought effects in a from of reducing costs. As an algorithm of subscribers assigned with an introduction of an additional commutation node, the following algorithm is used. While analyzing the next subscriber $j \in Z_i$ for all the commutation nodes $l \in D$, we compute the values of cluster's potentials for the commutation nodes with the help of the formula (4), with an assumption that the subscriber j belongs to cluster Z_l.Then we designate l^* for which $v_{l^*} = \max_{l \in D_S} v_l$. If $l^* \neq i$ then the subscriber j is assigned to a l^*node.

The above procedure is iterationally repeated until full browsing of a subscribers' list, neither of them (nor their group) is transferred among clusters. The computational complexity of the cluster equals O(N^4).

3.2. Algorithm 2

This algorithm is different from the first one in its manner of the subscribers assignment to the nodes realization. In a given case, the procedure is processed as local search algorithm in surrounding of a given network structure relatively to the assignment change. Local assignment is realized with the most decrease method where the search algorithm in the whole surrounding is proceeded and a network structure with the least cost is chosen. The source network is changed to the returned one, and local searching is iterations carried on until the structures with smaller cost are not present in surrounding of a given network structure. Computational complexity of the algorithm is O(N^5).

If for all $j = 1,2,...N$, $\Lambda_j^+ + \Lambda_j^- < \Lambda_m$, then algorithms 1 and 2 present acceptable assignments. Actually, if in some start iterations unacceptable network variants are obtained, ultimately the network becomes unacceptable (in a border case the complete network with grade N arises).

3.3. Algorithm 3

In order to specify a localization of commutation nodes, the full search procedure of all possible combinations of N subscribers for L centers. The preliminary assignment is achieved with the maximal potential method which is

described as follows: the subscriber $j \in A_S$ is assigned to i communication node for which $P_{ij} = \max_{i \in D_s} P_{ij}$, i.e. to center relatively to which it has the smallest potential. As an assignment algorithm the procedure from the algorithm 1 is used.

3.4. Algorithm 4

In order to specify a localization of communication nodes, the full search procedure of all possible combinations of N subscribers as L centers is used. In order to specify a preliminary assignment the following algorithm is used: the subscriber $j \in A_S$ is assigned to commutation nodes i^* for which:

$$C_{LK}\left(\Lambda_j^+ + \Lambda_j^-, r_{ij}\right) = \min_{i \in D_s} C_{LK}\left(\Lambda_j^+ + \Lambda_j^-, r_{ij}\right).$$

As an assignment algorithm the procedure as in the algorithm 2 is used. Algorithms 3 and 4 have expotential complexity which is a result of implemented procedure of center choice.

3.5. Algorithm 5

In order to specify localization points of commutation nodes, the values of $P_i = \sum_{j=1 j \neq i} P_{ij}$, for $i = 1,2,...,N$ are computed. Next, the set $\{P_i \mid i = 1,2,...N\}$ is sorted the increasing order and from it the first L modules (specific number) are chosen, referring to subscribers who are candidates for cluster's centers.

The general algorithm scheme of assignment is as follows. The potential matrix P is a set and a set D is that of localization points commutation nodes. For each $i \in D$, the list $Z(i)$ of subscribers incoming to the cluster from a center in point i is created. Moreover, the list $\overline{I}$ of subscriber numbers, not incoming to any cluster is created. At the beginning $\overline{I} = A$ and $Z(i) = i$.

Next, from the list $\overline{I}$ the first element j_1, i.e. a certain subscriber not incoming to any of clusters and for all cluster centers $i \in D$ the summarized potential of node j_1 with the cluster subscribers whose the center is i, $p_{j_1}(i) = \sum_{I \in Z(i)} P_{i_i l}$. This value characterizes the degree of j_1 subscriber attraction to cluster from the center i.

Next, the values $p_{j_1}(i)$ are compared for all $i \in D$. If there exists such a i^* that $p_{j_1}(i^*) \geq \beta p_{j_1}(i)$ for $\forall i \in D,\ i \neq i^*$ then the node referring to j_1 is assigned to the center i^* then deleted from the list $\overline{I}$ and added to the list $Z(i^*)$. The value β defines some threshold attractions (in this case $\beta = 20$). If such a i^* does not

exists then j_1 returns to list $\overline{I}$ and next from the list the other number j_2 is chosen.

The described iteration is carried out for all $j \in \overline{I}$. If during the next full passage of the list $\overline{I}$ no node has been assigned, then the new value β is changed into a value of 0.9β and the whole process is repeated from the beginning. The algorithm's computing complexity is $O(N(N+\beta L))$.

4. Alternative methods of Designing New Topology

Nowadays most of the algorithms for designing logical topology are based on solving MILP problems. The methods are characterized by a huge computational complexity and relatively low effectiveness of designing. For this reason we have to apply other methods which are based on the use of logical topology. These methods are prepared in many different ways and then projected on physical topology. We also propose methodology of building logical topologies based on utilization of regular and symmetric graphs [3].

4.1. Product Network

Embedding of one topology within another gives new capacity to build new communication network and also allows to formulate problem finding effective representation of data structures or to allocate circuits in the distributor of VLSI. In order to characterize operation of embedding, particular representation of topology must be adopted. Let the interconnection network be modeled by an undirected graph $G(V_G,E_G)$, where the set of nodes V_G represents the computers and the set of all edges E_G represents network connections.

Given the undirected graphs $G(V_G,E_G)$ and $H(V_H,E_H)$, the direct product graph $G \times H$ has a node set $V_G \times V_H$. Let x and y be nodes of graph G, and let u and v be nodes of graph H. Then $((x,u), (y,v))$ is an edge of graph $G \times H$, if either (x,y) is an edge of graph G and $u=v$ or (u,v) is an edge of graph H and $x=y$. Then the product graph can be modeled as graph G with a node consisting of the graph H, or as graph H with the node consisting of the graph G.

The hypercube is a good general-purpose interconnection network. It allows for some good interconnection characteristics. An n-node hypercube can embed many other n-node networks like complete binary trees, meshes, mesh of trees, etc. One of the hypercube's advantages, scalability is also liability. When the node degree is a slow-growing function of the number of nodes, we get the same communication interconnection properties for different orders of a graph, and thus scalability [4].

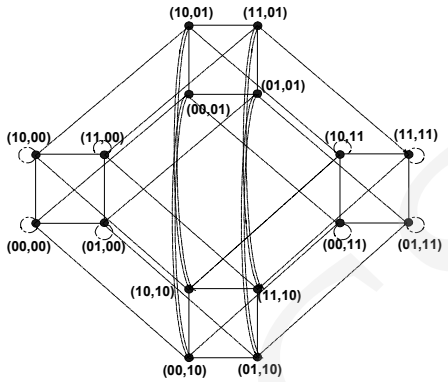

Fig. 1. Example of the product network of deBruijn(2) and hypercube (2)

Figure 1 shows the representation of $HD(2,2)$. Given a node (u,v) will be called u the hypercube-part-label and v the deBruijn-part-label. It is important that the nodes with the same deBruijn-part-label form a hypercube of m-order, while the nodes with the same hypercube-part-label form a deBruijn graph of n order. It follows that there are 2^n hypercube subgraphs $H(m)$ in $HD(m, n)$ where the nodes in each $H(m)$ have the same deBruijn-part-label. Identification of these subgraphs will be by their corresponding deBruijn-part-labels. Alternatively, $HD(m, n)$ can be thought of as having 2^m deBruijn subgraphs $D(n)$ where the nodes in each $D(n)$ have the same hypercube-part-label. We will identify these subgraphs by their corresponding hypercube-part-labels.

Table 1 shows, that hypercube, deBruijn and hyper-deBruijn, all of the same size, posses the same diameter m+n with a different degree. The deBruijn graphs have the least degree 4, while the hypercube has the maximum degree $m+n$, and the hyper-deBruijn graph can have any desired degree of $HD(m, n)$ is $m+4$, and is independent of n. Therefore from the point of view of node degree $HD(m, n)$ is of interest when $n \geq 4$ [5].

Table 1. Summary of parameters Product Network

Parameter	Hypercube	deBruijn	Hyper-deBruijn
Nodes	$N = 2^{m-n}$	$N = 2^{m+n}$	$N = 2^{m+n}$
Degree	m+n	4	m+4
Diameter	m+n	m+n	m+n
Fault-tolerance	m+n-1	1	m+2

4.2. Designing Logical Topology with the use of d-optimum graphs

One of the most important tasks of logical topology is to keep the high level of liveness of the system. It can be achieved by applying redundancy of logical communication channels. The original design algorithm of the fault tolerant

topology is presented below. In most cases designing such a mesh is based on an edge connectivity limitation system of ρ_k graph, which represents communication architecture. It can be presented as:

$$\rho_k = \rho(G, P_k) \geq t_k, \quad k = \overline{1, L}. \tag{5}$$

The *G*-*S* graph ($S \subseteq V_G$, where V_G – the set of nodes of the *G* graph) is incoherent and each of its components contains fewer than *k* nodes in relation to a P_k condition set.

Supposing $D(G)$ was a capacity coefficient of *G* graph the ρ_k value in the homogeneous graphs of δ degree (where $1 \leq k \leq D(G)$) would accomplish the maximum of significance.

Due to this fact, we should maximize the *D(G)* value in the process of designing communication mesh topology. The above statement can be applied to heterogeneous graphs which fulfil the condition of *V*-homogeneity ($\forall X \in V_G, \delta \leq \deg(X) \leq \delta + 1$, where $\delta = \lceil 2M/N \rceil$, *M* – a number of the *G* graph edges, *N* – a number of the *G* graph nodes, *deg*(*X*) – a degree of the *x* node in the *G* graph).

It can be proved that the homogeneous graphs with *N* nodes, δ degree and *D* capacity coefficient are applied to the following formula:

$$1 + \delta \sum_{k=0}^{D-1/2} (\delta - 1)^k \leq N, \; D \text{ being odd}, \tag{6}$$

$$2\delta \sum_{k=0}^{D-2/2} (\delta - 1)^k \leq N, \; D \text{ being even}. \tag{7}$$

The above formula can be used to determine one of the parameters δ, *N, D* with determining the two others at the same time (particularly the capacity coefficient of the homogeneous graph with *N* nodes and δ degree. The notion of *D*-optimum graphs need to be introduced. *D*-optimum graph is a graph where

$$\forall e = (X, Y) \in V, \; d_{G-e}(X, Y) = d(G - e), \tag{8}$$

$d_{G\text{-}e}(X,Y)$ – the distance between *X* and *Y* nodes in *G*-*e* graph, $d(G–e)$ – the diameter of *G*–*e* graph. It can be proved that:

$$D(G) \geq d(G) + 1. \tag{9}$$

In *D*-optimum graphs the capacity coefficient has been limited from the bottom through the diameter of the graph. It can be used in the synthesis of fault tolerant systems [6]. In order to determine *D*-optimum graph, there are used e.g. genetic algorithms, heuristic algorithms.

Conclusions

By utilizing methods of embedding, Latin square or d-optimum graph in design of new topologies we allow to accomplish new original topologies. These new structures are characterized by predictable parameters described over by

attributes of the base topologies. Operation of the Cartesian Product by one means guarantee of a small degree of computation nodes by the other means it is possible to achieve a small diameter. Embedding allows connection of two popular topologies with fixed attributes and attaining new topology, which inherits all the above abilities of topologies. Additional attribute of this operation is the fact that the number of nodes is growing by means of multiplicity and the diameter is growing by addict ability. Increasing structure is also characterized by a simpler structure of implementation of the network algorithms based on fundamental and commonly known in base topologies algorithms. All these attributes are particularly useful in the system scalability and allow to increase the network size, retaining a stabilized number of nodes.

References

[1] Hajder M., Dymora P., Mazurek M., *Improvement methods of multichanel bus topologies parameters*, Sixth International Conference Soft Computing and Distributed Processing – WSIiZ Rzeszów, (2001).

[2] Hajder M., Mazurek M., Dymora P., *Designing of multichannel optical communication system topologies criteria optimization*, Conference: "Informatyka – badania i zastosowania", Annales UMCS Informatica, Kazimierz Dolny, (2003).

[3] Hajder M., Mazurek M., *Grafy regularne w projektowaniu wielokanałowych optycznych systemów komunikacji*, Zeszyty Naukowe Politechniki Rzeszowskiej, Informatyka, Rzeszów, 1 (2003), in Polish.

[4] Mazurek M., *Hyper-deBruijn topology as a result of the Network Product*, International Conference of Science and Technology Scalnet'04, Scalable Systems and Computer Networks Design and Aplications, Ministry of Education and Science of Ukraine Kremenchuk State Polytechnical University, Kremenchuk, (2004).

[5] Ganesan E., Pradhan D.K., *The Hyper-deBruijn Networks: Scalable Versatile Architecture*, IEEE Transactions on parallel and distributed systems, 4(9) 1993.

[6] Hajder M., Mazurek M., *Wybrane aspekty syntezy i analizy systemów komunikacji optycznej*, Współczesne problemy sieci komputerowych – nowe technologie, WNT, (2004), in Polish.