

# The method for optimal server placement in the hypercube networks

Jan Chudzikiewicz

Military University of Technology,  
ul. S. Kaliskiego 2, 00-908  
Warszawa, Poland  
Email: jchudzikiewicz@wat.edu.pl

Tomasz Malinowski

Military University of Technology,  
ul. S. Kaliskiego 2, 00-908  
Warszawa, Poland  
Email: tmalin@ita.wat.edu.pl

Zbigniew Zieliński

Military University of Technology,  
ul. S. Kaliskiego 2, 00-908  
Warszawa, Poland  
Email: zzielinski@wat.edu.pl

**Abstract**— In this paper, the problem of determining the most effective server placement in the hypercube network structure was considered. The algorithm consisting of two stages: first stage for the server placement and the second for generating the appropriate communication structure was described. The correctness of the algorithm has been verified through simulation tests, prepared and implemented in Riverbed Modeler environment. The results of these tests for exemplary structures were presented. Some properties of the server placement in the 4-dimensional hypercube network with soft degradation were investigated.

## I. INTRODUCTION

The computer networks with a regular structure as torus or hypercube ([1]–[4]) could be used in many kinds of specialized critical application (for instance military, aerospace or medical systems). An interconnection network with the hypercube logical structure is a well-known interconnection model for multiprocessor systems ([5]–[6]) and still hypercube networks are the field of interest of many theoretical studies concerning (among others) resource placement problem, which has been intensively studied in [7]–[12].

Specialized systems with critical application are usually used in real-time mode and required both very high reliability and high efficiency of data processing throughout all the network life cycle. In order to achieve high reliability of the system the network could be considered as soft degradable computer network [12]–[13]. In this kind of networks a processor identified as faulty is not repaired (or replaced) but access to it is blocked. New (degraded) network continues work after resources reassigning and under the condition that it meets special requirements. In turn, the efficiency of the system will depend heavily on the availability of resources (data bases, files or web data), which is determined by their placement in the network. So, for this kind of networks there is necessity for applying effective methods of resources placement. In the work [12] an analysis of the different schemas of resources placement

in the 4-dimensional hypercube network with soft degradation was conducted.

Designing and exploitation of special networks in critical application is a comprehensive task that requires addressing a number of theoretical and practical problems. One of the problems is a skillful resources deployment in the network and modification of resources deployment after each phase of the network degradation. One of considered in the literature the resource placement problem is a combination the distance- $d$  and the  $m$  adjacency problems, where a non-resource node must be a distance of at most  $d$  from  $m$  processors nodes [7]–[10], [12]. In [10] a perfect deployment has been introduced and analyzed which is defined as the minimum number of resources processors which are accessible by working processors. The definition - perfect deployment is a characteristic of the value of the generalized cost of information traffic in the network at a given load of tasks. In [12] the notion of  $(m,d)$ -perfect resources placement in the hypercube type structure  $G$  has been extended to the such allocation of  $k$  resources which minimizes the average distances between the working processors and resource processors in the structure  $G$ .

We investigate the case when a specialized computer system is based on the 4-dimensional hypercube skeleton network with communication nodes which could communicate between themselves via cable connections. The main task of the hypercube network is to provide efficient access to resources managed by the server (or cluster of servers) connected directly to one of the network nodes and semi-stationary clients communicating with the assigned network nodes via wireless links. The execution of applications by a client processor requires an access to server services and resources, also some results returned by the server must be submitted to other clients. We assume that all clients are responsible for performing the same or very similar tasks. Thus all clients will generate similar workload of the network. The problem which arises for the given network structure is to determine the most effective server placement in the network structure.

The main goal of this paper is to give an effective method of solving the server placement problem for the hypercube network along with its soft degradation process.

A generalized cost of a network traffic with a specified resources deployment and workload of a network is usually tested through experimental measurements or examined with the use of simulation methods. In the paper we apply a two phased approach. In the first stage we solve the problem of a server placement in the given network structure on the base of analytically determined attainability measure, which was proposed in [12]. It should be noticed that real cost of information traffic in a network for a given deployment of the server with resources depends on the nature of the tasks performed by clients in the network. In the second stage we have examined this problem with the use of simulation methods for the specified server deployment determined by the simple analytical method and given type of task load of the network.

We see our contributions as follows. Firstly, we have extended the approach proposed in [12] to the determining server placement in the hypercube network with soft degradation on the base of nodes attainability calculation. Secondly, we propose the algorithm of the communication structure assignation with the use of dendrite calculation. Next, we show the feasibility of this approach by applying obtained results to some possible structures of degraded 4-dimensional hypercube network and verifying effectiveness of server placement by simulation experiments.

The rest of the paper is organized as follows. In Section II, a basic definitions and properties were introduced. The calculation of radius and attainability for exemplary structures were presented. In Section III, the proposal of the algorithm determining server placement was presented. An illustration of the main algorithm steps for the exemplary structure was given. In Section IV, the results of simulation tests for verification the algorithm (implemented in Riverbed Modeler environment) were described. In Section V, some concluding remarks were presented.

## II. BASIC DEFINITIONS AND ASSUMPTIONS

**Definition 1.** The logical structure of processors network we call the structure of  $n$ -dimensional cube if is described by coherent ordinary graph  $G = \langle E, U \rangle$  ( $E$  – set of computer,  $U$  – set of bidirectional data transmission links), which nodes can be described (without repetitions) by  $n$ -dimensional binary vectors (labels) in such a way that

$$[\delta(\varepsilon(e'), \varepsilon(e'')) = 1] \Leftrightarrow [(e', e'') \in U] \quad (1)$$

where  $\delta(\varepsilon(e'), \varepsilon(e''))$  is Hamming distance between the labels of nodes  $e'$  and  $e''$ .

The Hamming distance between two binary vectors  $\varepsilon(e')$  and  $\varepsilon(e'')$  complies with the dependency:

$$\delta(\varepsilon(e'), \varepsilon(e'')) = \sum_{k \in \{1, \dots, n\}} (\varepsilon(e')_k \oplus \varepsilon(e'')_k)$$

where:

- $\varepsilon(e')_k$  – the  $k$ -th element of the binary vector  $\varepsilon(e')$ ,

- $\oplus$  – modulo 2 sum.

We investigate the case when skeleton of the network has the logical structure of 4-dimensional hypercube (Fig. 1). A topology of the hypercube may be represented by an ordinary consistent graph whose nodes are described by 4-dimensional binary vectors such that the Hamming distance between vectors (labels) of the adjacent nodes equals one. If  $|E| = 2^4$  and  $|U| = 2|E|$ , then such graph we called (non labeled) 4-dimensional cube and will be denote by  $H^4$ . Thus  $H^4$  is a regular graph of degree of 4 i.e. such that the degree of a node  $e \in E$  we determine as  $\mu(e) = |E(e)|$ , where  $E(e)$  is a set of nodes adjacent to the node  $e \in E$  and  $\mu(e) = 4$  for each node  $e$  of the graph  $H^4$ .

Let  $d(e, e'|G)$  be the distance between nodes  $e$  and  $e'$  in a coherent graph  $G$ , that is the length of the shortest chain (in the graph  $G$ ) connecting node  $e$  with the node  $e'$ .

Let  $r(e|G) = \max_{e' \in E(G)} d((e, e')|G)$  be the greatest distance from the node  $e \in E(G)$  to another node of the set  $E(G)$ , and  $r(G)$ , and  $D(G)$  (respectively) denote the radius and the diameter of a graph  $G$  i.e.  $r(G) = \min\{r(e|G): e \in E(G)\}$  and  $D(G) = \max\{d(e', e''|G): \{e', e''\} \subset E(G)\}$ .

**Property 1.** For the 4-dimensional cube  $H^4$  the equation is complied

$$D(H^4) = r(H^4) = 4.$$

It is known that  $D(G) \leq 2r(G)$ .

If  $r(e|G) = r(G)$  then the node  $e$  is called the central node of the network  $G$ .

Denote by  $E^{(d)}(e|G) = \{e' \in E(G): d(e, e'|G) = d\}$  for  $d \in \{1, \dots, D(G)\}$ , and by

$$\begin{aligned} \varsigma(e|G) &= (\varsigma_1(e|G), \dots, \varsigma_{r(e|G)}(e|G)) \text{ for} \\ \varsigma_d(e|G) &= |E^{(d)}(e|G)| \end{aligned} \quad (2)$$

**Definition 2.** Let  $\varphi(e|G) = \sum_{e' \in E(G)} d(e, e'|G)$  ( $e \in E(G)$ ) be attainability of the computer  $e$  in the network  $G$  and by  $\Phi(G) = \sum_{e \in E(G)} \varphi(e|G)$  attainability of the network  $G$ .

Using (2) we have

$$\varphi(e|G) = \sum_{d=1}^{r(e|G)} d \varsigma_d(e|G) \quad (3)$$

**Property 2.**  $\Phi(H^4) = 512$  because  $\forall_{e \in E(H^4)}: (r(e|H^4) = 4 \wedge \varsigma_d(e|H^4) = \binom{4}{d})$ . Using (3) we have  $\forall_{e \in E(H^4)}: \varphi(e|H^4) = 32$  and  $|E(H^4)| = 2^4$ , then  $\Phi(H^4) = |E(H^4)| \varphi(e|H^4)$  [12].

**Example 1.** Figure 1 presents all the seven possible cyclic structures upon the occurrence of  $k = 7$  consecutive failures of processors of the network  $H^4$  which are the subgraphs of  $H^4$  [13].

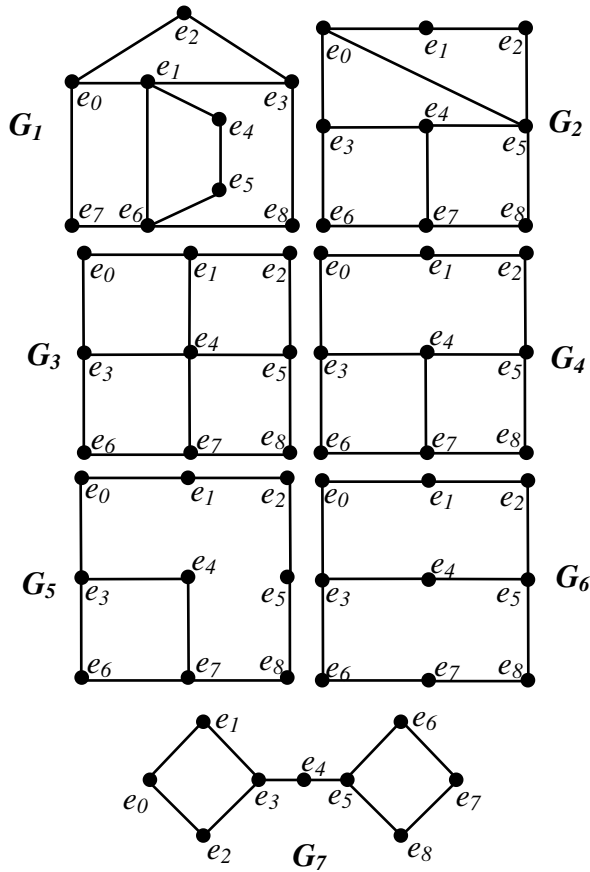

 Fig. 1 Example of cyclic subgraphs of  $H^4$  order 9 [13]

TABLE I.

 THE  $r(e|G)$ ,  $r(G)$  AND  $D(G)$  FOR THE STRUCTURES PRESENTED ON THE FIGURE 1

$r(e G_i)$	$r(e G_1)$	$r(e G_2)$	$r(e G_3)$	$r(e G_4)$	$r(e G_5)$	$r(e G_6)$	$r(e G_7)$
$e \in E(G)$							
$e_0$	3	3	4	4	4	4	6
$e_1$	2	4	3	4	4	4	5
$e_2$	4	4	4	4	4	4	5
$e_3$	3	3	3	3	4	3	4
$e_4$	3	3	2	3	4	3	3
$e_5$	4	3	3	3	4	3	4
$e_6$	3	4	4	4	4	4	5
$e_7$	3	4	3	4	4	4	6
$e_8$	3	3	4	4	4	4	5
$r(G)$	2	3	2	3	4	3	3
$D(G)$	4	4	4	4	4	4	6

It should be noticed, that for the given network structure  $G$  on the base of the obtained measures  $r(e|G)$  it would be rational to choose the server placement at the central node of the network or in the node with the minimum value  $r(e|G)$ . In some cases (let's consider the structures  $G_2, G_4, G_5, G_6$ ) we are not able to choose the best server placement. Then we can have determined  $\zeta(e|G)$  using (2) and  $\varphi(e|G)$  using (3) for these structures. The determined values of  $\zeta(e|G)$ ,  $\varphi(e|G)$  and  $\Phi(G)$  are presented in table II.

**Definition 3.** Let  $T = \langle E, U^* \rangle$  be the dendrite i.e. such coherent acyclic partial graph of  $G$  that:

$$\exists \langle e', e'' \rangle \in U \Rightarrow \langle e', e'' \rangle \in U^* \Leftrightarrow [(d(e_i, e') \neq d(e_i, e'')) \wedge d(e', e'') = 1] \text{ for } r(e_i) = \min_{e \in E(G)} r(e).$$

The dendrite  $T$  is a communication structure of  $G$ . The method for determined the dendrite  $T$  is presented in section III.

TABLE II.

 THE  $\zeta(e|G)$ ,  $\varphi(e|G)$  AND  $\Phi(G)$  FOR THE STRUCTURES  $G_2, G_4, G_5, G_6$  PRESENTED ON THE FIGURE 1

$G_2$						$G_4$					
$d(e, e' G)$	1	2	3	4	$\varphi(e G_2)$	1	2	3	4	$\varphi(e G_4)$	
$e \in E(G)$											
$e_0$	3	4	1	0	14	2	3	2	1	18	
$e_1$	2	2	3	1	19	2	2	3	1	19	
$e_2$	2	3	2	1	18	2	3	2	1	18	
$e_3$	3	3	2	0	15	3	3	2	0	15	
$e_4$	3	4	1	0	14	3	4	1	0	14	
$e_5$	4	3	1	0	13	3	3	2	0	15	
$e_6$	2	3	1	1	18	2	3	2	1	18	
$e_7$	3	2	2	1	17	3	2	2	1	17	
$e_8$	2	4	2	0	16	2	3	2	1	18	
$\Phi(G_2)$					144	$\Phi(G_4)$					152

$G_5$						$G_6$					
$d(e, e' G)$	1	2	3	4	$\varphi(e G_5)$	1	2	3	4	$\varphi(e G_6)$	
$e \in E(G)$											
$e_0$	2	3	2	1	18	2	3	2	1	18	
$e_1$	2	2	3	1	19	2	2	3	1	19	
$e_2$	2	2	2	2	20	2	3	2	1	18	
$e_3$	3	2	2	1	17	3	2	3	0	15	
$e_4$	2	3	2	1	18	2	4	2	0	16	
$e_5$	2	2	3	1	19	3	3	2	0	15	
$e_6$	2	3	2	1	18	2	3	2	1	18	
$e_7$	3	2	2	1	17	2	2	3	1	19	
$e_8$	2	3	2	1	18	2	3	2	1	18	
$\Phi(G_5)$					164	$\Phi(G_6)$					156

### III. THE METHOD AND THE ALGORITHM FOR DETERMINING A SERVER PLACEMENT AND COMMUNICATION STRUCTURE

The method consists two stages. In the first stage for  $G$ , as the first node we choose a node that  $r(e_i) = \min_{e \in E(G)} r(e)$  or  $\varphi(e_i|G) = \min_{e \in E(G)} \varphi(e|G)$  – *server placement*. In the second stage for the chosen node we determine the dendrite  $T$ , which is a communication structure satisfying the condition  $d_{\max}(e_i|T) = r(e_i)$ . Based on the presented method the algorithm for determining the *server placement* and the communication structure was developed.

*The algorithm for determining the server placement and communication structure.*

#### Step 1.

Determine  $r(e|G)$  for  $e \in E(G)$ .

#### Step 2.

Choose a node  $e_i \in E(G)$  such that  $r(e_i) = \min_{e \in E(G)} r(e)$ .

If  $|\{e_i\}| > 1$  go to step 3 else go to step 5.

**Step 3.**

Determine  $\varphi(e|G)$  for  $e \in E(G)$ .

**Step 4.**

Choose a node  $e_i \in E(G)$  such that  $\left(\varphi(e_i|G) = \min_{e \in E(G)} \varphi(e|G)\right) \wedge \left(\mu(e_i) = \max_{e \in E(G)} \mu(e)\right)$ .

Selected node  $e_i$  will be a central node of dendrite.

**Step 5. (second stage)**

Set  $k = 1$  and  $(E(T) = E^0(e_i|G) = \{e_i\})$ .

**Step 6.**

$E(T) = E(T) + E^k(e_i|G)$ .

$U^* = U^* \cup_{(e' \in E^{k-1}(e_i|G)) \wedge (e'' \in E^k(e_i|G))} (e', e'')$ .

**Step 7.**

$k = k + 1$ .

Check if  $k > r(e_i)$ .

YES

Go to step 8.

NO

Return to step 6.

**Step 8.**

The end of the algorithm.

Figure 2 is an illustration of the algorithm. In the first stage of the algorithm (steps 1-4) the central node  $e_1$  for the structure  $G_1$  (figure 1) was appointed. In the table III the results of determining radius (A) and attainability (B) of  $G_1$  nodes are presented. Determination of the attainability of  $G_1$  nodes is unnecessary, and should be used only if the steps 1-2 of the algorithm will not allow unambiguous to determine the central node for given structure  $G$ .

Dendrite  $T$  determined in the second stage of the algorithm for the central node  $e_1$ , is one of the possible (but optimal) communication structure.

TABLE III  
THE RESULTS OF DETERMINING THE RADIUS (A), AND THE  
ATTAINABILITY (B) OF THE  $G_1$  NODES

A		B					
$e \in E(G_1)$	$r(e G_1)$	$d(e, e' G_1)$	1	2	3	4	$\varphi(e, G_1)$
$e_0$	3	$e_0$	3	3	2	0	15
$e_1$	2	$e_1$	4	4	0	0	12
$e_2$	4	$e_2$	2	3	2	1	18
$e_3$	3	$e_3$	3	3	2	0	15
$e_4$	3	$e_4$	2	3	3	0	17
$e_5$	4	$e_5$	2	3	2	1	18
$e_6$	3	$e_6$	4	3	1	0	13
$e_7$	3	$e_7$	2	4	2	0	16
$e_8$	3	$e_8$	2	4	2	0	16
$\Phi(G_1)$							140

**IV. THE RESULTS OF SIMULATION STUDIES**

Procedure for determining the best location of resources in the hypercube network specified in the section III has been verified through simulation tests. The aim of the test was to confirm the correctness of the theoretical considerations and arguments.

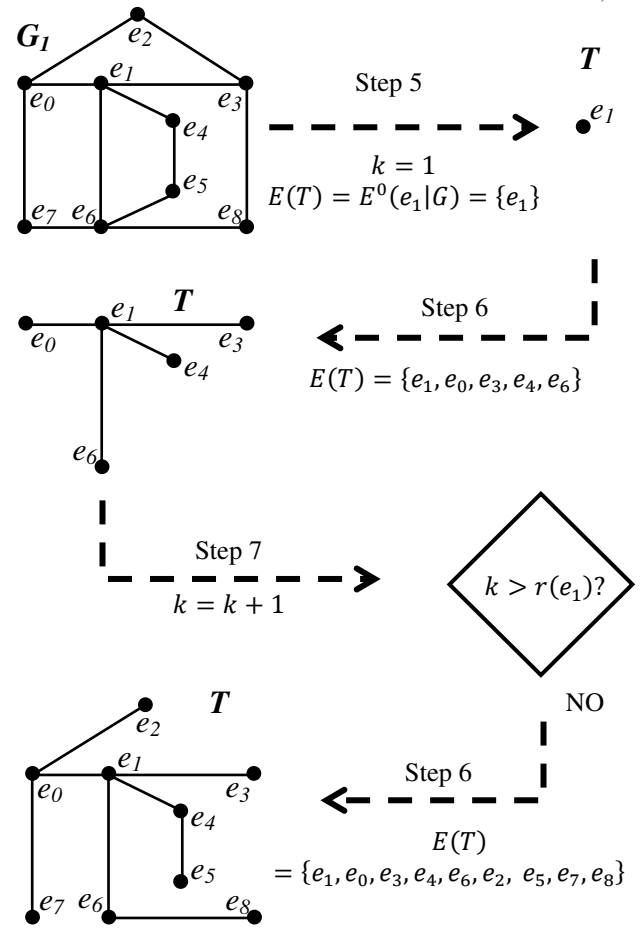


Fig. 2 An illustration of the algorithm steps

Simulation studies have been prepared and implemented in Riverbed Modeler environment. Subgraph  $G_1$  (figure 1) has been the subject of research.

Nodes have been modeled as routers and LAN segments attached to them. The cases when the server (with different typical and popular network services, chosen arbitrarily by authors) is connected to the selected node within  $T$  set of  $G_1$  structure were examined. In the figure 3 different  $T$  set of substructures (communication structures) correspond to different simulation scenarios is shown. The circle indicates node which the server is connected to.

For example, the network topology of the single scenario with  $T$ 's dendrite 1 was shown in the figure 4.

The name of the simulation scenario (*Dend\_1* in figure 4) was associated with the node's number, which the server was connected to (number of the dendrite's central node).

The server was acting as a database server, ftp server, web server, and the node with which it was possible to communicate through VoIP (*Voice over IP*). Workstations within LAN segments (ten workstations in each segment) were functioning as the server's clients. All network services have used the standard application models, available at Riverbed Modeler ("High Load" ftp and database models, "Heavy Browsing" http model and "PCM Quality Speech" voice model) [15].

The communication structure (skeleton of the network) was modeled as a set of routers connected via 1,5 Mb/s links.

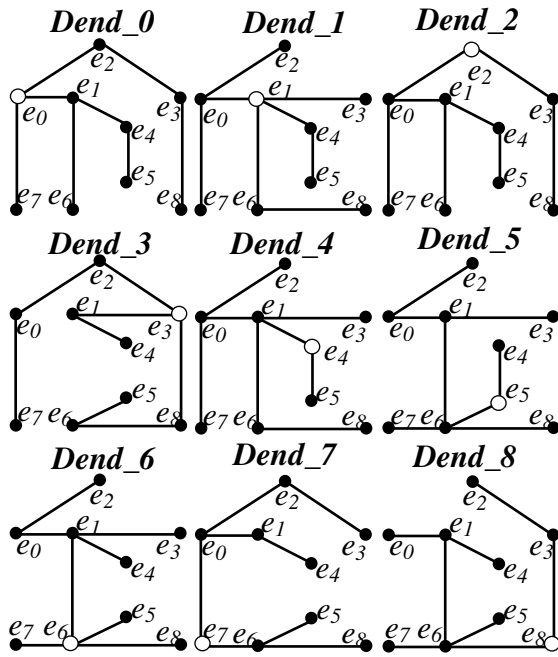


Fig. 3 The  $T$  of dendrites of  $G_1$

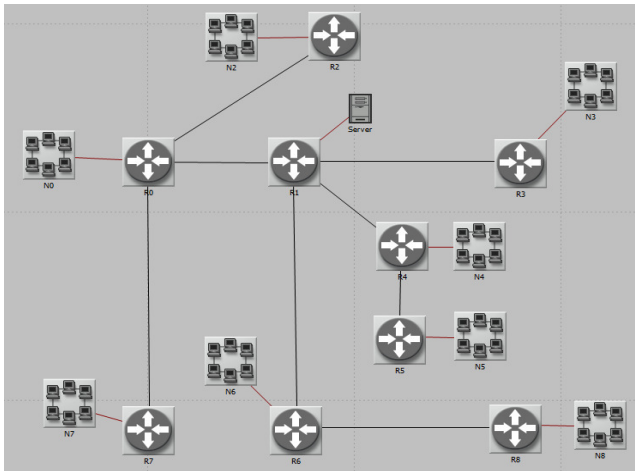


Fig. 4 Network topology of  $Dend_1$  scenario

Some interesting results, confirming the correctness of the procedure for determining the server placement and communication structure, are shown in the figures below. The dotted lines (e.g. Fig. 5) corresponds to the results obtained for the structure  $Dend_1$ , which is, according to the procedure, the most effective (the best) communication structure for subgraph  $G_1$ .

For each simulation scenarios ( $Dend_0$  to  $Dend_8$ ) five characteristics were determined.

#### A. End-to-end delay ( $EE\_Del$ )

End-to-end delay is average delay in seconds for all LAN segments nodes communicating with the server through

VoIP. The lowest value of  $EE\_Del$  is desired (it is the best score).

Results obtained during the simulation are presented in the figure 5. The figure 6 shows the average values of  $EE\_Del$ .

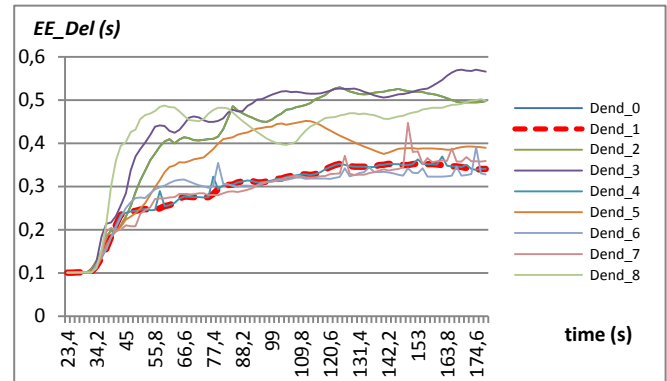


Fig. 5 End-to-end delay for VoIP transmission

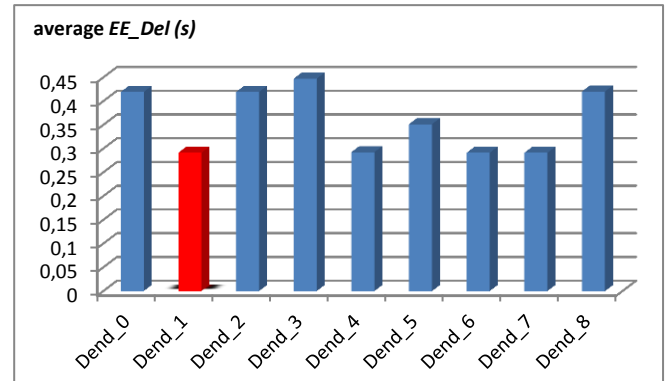


Fig. 6 Average End-to-end delay for  $Dend_0$  to  $Dend_8$  structures

#### B. TCP Delay ( $TCP\_Del$ )

$TCP\_del$  represents delay of TCP packets in seconds. This value is measured from the time an application data packet is sent from the source TCP layer to the time it is completely received by the TCP layer in the destination node. It is average delay in the complete network, for all connections. The lowest values are the best.

The results are presented in figures 7 and 8.

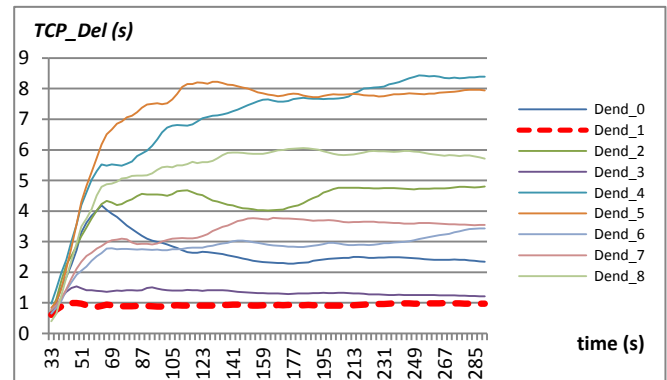


Fig. 7 TCP delay for TCP-based services

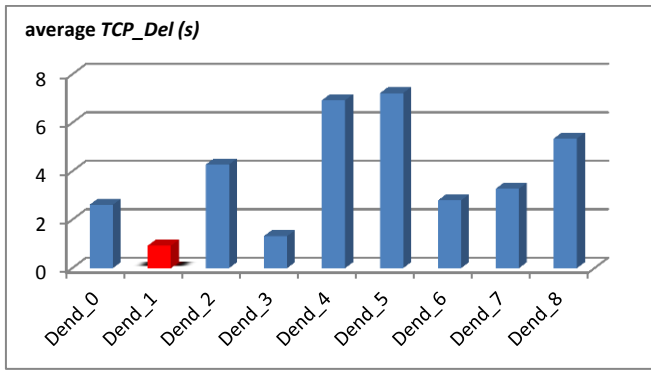
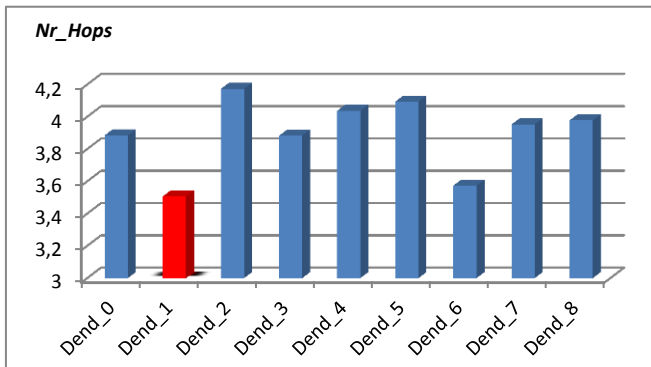


Fig. 8 Average TCP delay for TCP-based services

### C. Number of Hops (Nr\_Hops)

*Nr\_Hops* represents an average number of IP hops taken by data packets reaching at a destination node.

We expected the lowest value for *Dend\_1* structure and results are presented in the figure 9.

Fig. 9 Average Number of Hops for *Dend\_0* to *Dend\_8* structures

### D. Response Time (Res\_Time)

*Res\_Time* is average time elapsed between sending a request and receiving the response packet in seconds. It was measured for all the server's services (database, WWW and FTP).

The selected graph, *Response Time* for the database service was presented in the figure 10.

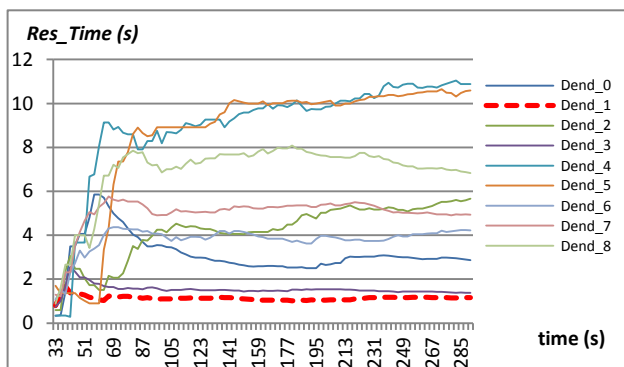
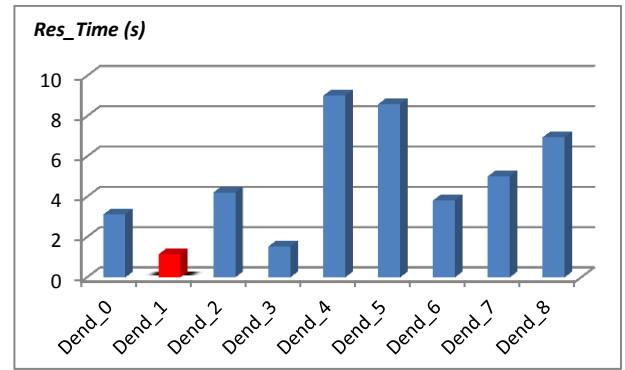


Fig. 10 Response Time for database service

Average values of the database server's response time for each simulation scenario are shown in the figure 11.

Fig. 11 Average Response Time for database service for all  $G_1$ 's dendrites

### E. Traffic Received (Traf\_Rec)

*Traf\_Rec* is an average number of bytes per second forwarded to server's application by the transport layer in the complete network. It was measured for all the server's service and treated as a transmission speed indicator, so we expected highest values for the best communication structure (*Dend\_0* in the drawings of the selected service – Fig.12 and Fig.13).

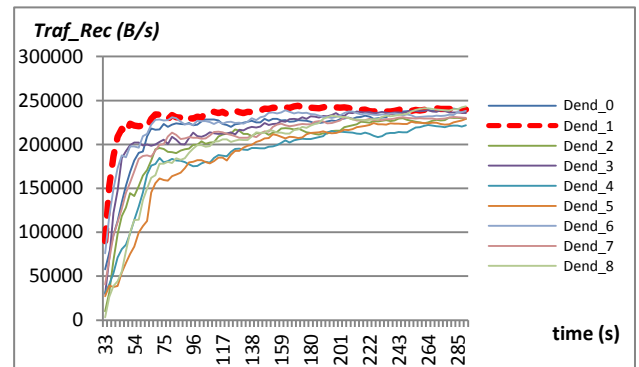
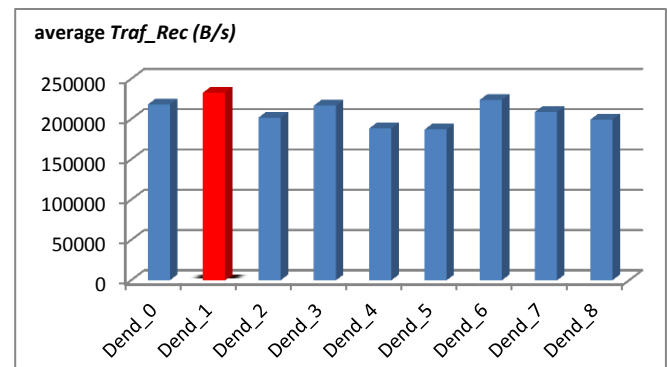


Fig. 12 Traffic Received for the FTP server

Fig. 13 Average FTP Traffic Received for all  $G_1$ 's dendrites

All the results are presented in Table III. It should be noted that the best result was reported in most measurements for *Dend\_1* structure (winning factor - 78%).



TABLE IV  
THE SIMULATION RESULTS FOR DENDRITES OF G 1 STRUCTURE

	Nr_Hops	TCP_Del (s)	EE_Del (s)	HTTP		DB		FTP		Winning factor (%)
				Res_Time (s)	Traf_Rec (B/s)	Res_Time (s)	Traf_Rec (B/s)	Res_Time (s)	Traf_Rec (B/s)	
Dend_0	3.74	2.59245	0.419	1.113	70013	3.129	15163	8.22	170518	
Dend_1	3.43	0.87554	0.291	0.641	136128	1.137	18214	5.37	188778	78
Dend_2	4.10	4.043734	0.419	1.364	34099	4.210	14121	14.25	153116	
Dend_3	3.75	1.307666	0.447	0.936	119991	1.532	18350	6.86	168044	11
Dend_4	3.87	6.015415	0.292	1.959	25875	9.018	12745	19.35	140109	
Dend_5	3.88	6.610183	0.351	1.968	28586	8.599	11137	19.67	134400	
Dend_6	3.46	2.499316	0.291	0.873	66342	3.818	15677	10.45	179894	
Dend_7	3.76	2.953067	0.291	1.397	43371	5.018	16540	11.25	161662	
Dend_8	3.88	5.09135	0.420	1.417	36789	6.954	14086	16.71	143929	

## V. CONCLUSION

Correctness of developed algorithm and its usefulness for determining server placement and the optimal communication structure in the hypercube network with soft degradation on the base of nodes attainability calculation was confirmed by simulation tests. Although the simulation tests conducted mainly related to typical network services that results could be transferred to the network-critical applications. One of tested services which might be regarded as the critical application was capability of communicating through VoIP (Voice over IP). Obviously different system parameters

aren't equally important and their importance depends on the application type. In the paper the influence of the network communication structure for obtained values of exemplary parameters was investigated.

Further work will address adaptation of the algorithm for various types of network structures. Work will focus on networks with dynamic structure reconfiguration.

## REFERENCES

- [1] H. Hongwei, S. Wei, X. Youzhi, Z. Hongke, "A Virtual Hypercube Routing Algorithm for Wireless Healthcare Networks", Chinese Journal of Electronics, Vol.19, No.1, Jan. 2010, pp. 138-144.
- [2] Po-Jen Chuang, Bo-Yi Li, Tun-Hao Chao, "Hypercube-based Data Gathering in Wireless Sensor Networks", Journal Of Information Science And Engineering 23, 2007, pp. 1155-1170.
- [3] A. Z. Zieliński, J. Chudzikiewicz, Arciuch, R. Kulesza, "Sieć procesorów o łagodnej degradacji i strukturze logicznej typu sześciianu 4-wymiarowego", in Metody wytwarzania i zastosowania systemów czasu rzeczywistego, L. Trybus, Ed., Warszawa: Wydawnictwo komunikacji i Łączności, 2011, pp. 219-232. (in Polish).
- [4] A. Arciuch, "Reliability state of connections in a microprocessor network with binary hypercube structure", Electrical Review, R.86, No. 9/2010, pp. 154-156.
- [5] T. Ishikawa, "Hypercube multiprocessors with bus connections for improving communication performance", IEEE Trans. Computers, Vol. 44, No. 11, 1995, pp. 1338-1344.
- [6] A. B. Izadi, F. Özunger, "A Real-Time Fault Tolerant Hypercube Multiprocessor", IEE Proceedings - Computers and Digital Techniques, Vol. 149, No. 5, 2002, pp. 197-202.
- [7] B. F. AlBdaiwia, B. Bose, "On resource placements in 3D tori", Journal of Parallel Distributed Computer vol. 63, 2003, pp. 838-845.
- [8] B. F. AlBdaiwia, B. Bose, "Quasi-perfect resource placements for two-dimensional toroidal networks", Journal of Parallel Distributed Computer vol. 65, 2005, pp. 815-831.
- [9] M. M. Bae, B. Bose, "Resource Placement in Torus-Based Networks", IEEE Transactions on Computers, vol. 46, no. 10, October 1997, pp. 1083-1092.
- [10] N. Imani, H. Sarbazi-Azad, A.Y. Zomaya, "Resource placement in Cartesian product of networks", Journal of Parallel Distributed Computer 70, 2010, pp. 481-495.
- [11] P. Moizadeh, H. Sarbazi-Azad, N. Yazdani, Resource Placement in Cube-Connected Cycles, The International Symposium on Parallel Architectures, Algorithms, and Networks, IEEE Computer Society, 2008, pp. 83-89.
- [12] J. Chudzikiewicz, Z. Zieliński, "On some resources placement schemes in the 4-dimensional soft degradable hypercube processors network", Advances in Intelligent and Soft Computing. Proc. of the Ninth Int. Conf. on Dependability and Complex Systems DepCoS-RELCOMEX (W. Zamojski at al. Eds., Series Ed.: Kacprzyk Janusz), Springer 2014, pp. 133-143.
- [13] Z. Zieliński, „Podstawy diagnostyki systemowej sieci procesorów o łagodnej degradacji i strukturze hipersześcianu”, Wojskowa Akademia Techniczna, Warszawa, 2012, 182p. (in Polish).
- [14] T. Malinowski, A. Arciuch, "The procedure for monitoring and maintaining a network of distributed resources", Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 947-954, ACSIS, Vol. 2, DOI: 10.15439/2014F159.
- [15] A. S. Sethi, V.Y. Hnatyshin, "The Practical OPNET User Guide for Computer Network Simulation", Chapman and Hall/CRC, 2012.