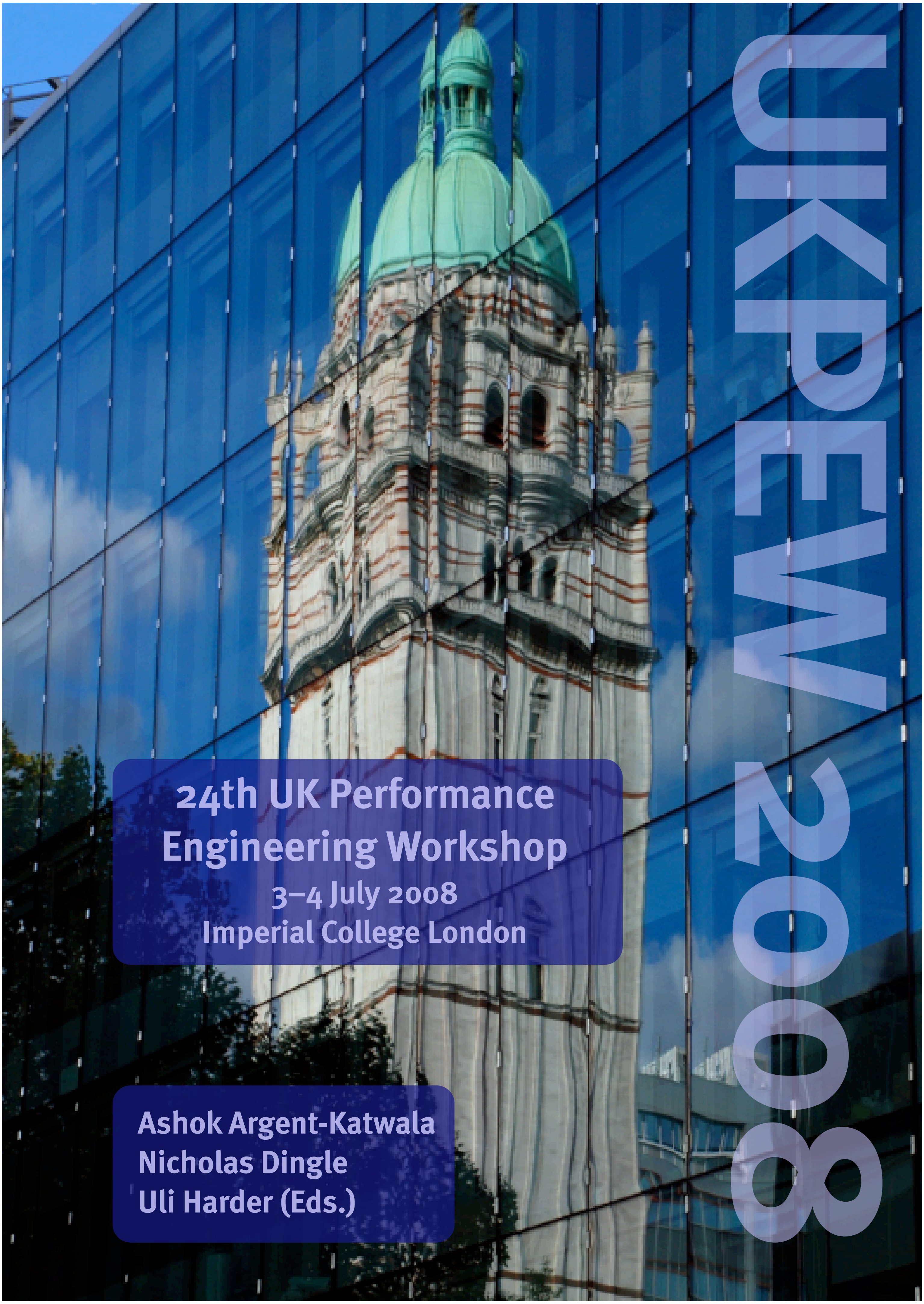


UKPEW 2008

24th UK Performance Engineering Workshop

3–4 July 2008

Imperial College London

Ashok Argent-Katwala
Nicholas Dingle
Uli Harder (Eds.)

24th UK Performance Engineering Workshop
3–4 July 2008

Department of Computing
Imperial College London

Edited by
Ashok Argent-Katwala, Nicholas J. Dingle and Uli Harder

Ashok Argent-Katwala, Nicholas J. Dingle and Uli Harder (Eds.)
Department of Computing
Imperial College London
Huxley Building
180 Queen's Gate
London
SW7 2RH

Departmental Technical Reports (Print): ISSN 1469-4166
Departmental Technical Reports (Online): ISSN 1469-4174
ISBN 978-0-9559703-0-6

Published as part of the Technical Reports Series (DTR08-9) of the Department of Computing, Imperial College London, Huxley Building, 180 Queen's Gate, London SW7 2RH.

Preface

Welcome to UKPEW 2008 at Imperial College London. This is the second time the event has been hosted by Imperial; the last time this happened was 16 years ago in 1992. Other previous locations of UKPEW were:

2007 Edge Hill University
2006 Poole
2005 Newcastle
2004 Bradford
2003 Warwick
2002 Glasgow
2001 Leeds
2000 Durham
1999 Bristol
1998 Edinburgh
1997 Ilkley (Bradford University)
1996 Edinburgh
1995 Liverpool John Moores
1994 Edinburgh
1993 Loughborough
1992 Imperial College, London
1991 Edinburgh
1990 Bradford
1989 Edinburgh
1988 Edinburgh
1987 Edinburgh (Heriot-Watt)
1986 Edinburgh
1985 1st UKPEW, Edinburgh

This year UKPEW features two keynote speeches: one by Professor Henri Bal from Vrije Universiteit Amsterdam and the other by Adam Grummitt from Metron Technology Ltd. Professor Bal's specialisation is parallel and distributed computing; he has, for example, published work (along with John Romein) on the use of a 144-processor parallel computer to solve the game of Awari, which required the exploration of 889,063,398,406 board positions. Metron Technology Ltd. produces performance management software packages which provide measurement, analysis, planning and reporting capabilities on a wide range of operating systems. Adam Grummitt is currently chair of the UK Computer Measurement Group and is also very active in the IT Infrastructure Library arena.

In total the proceedings include 29 papers from various UK institutions, including Bradford, Edinburgh, Glasgow, Heriot-Watt, Newcastle, Surrey, UCL, Warwick and of course Imperial. In addition to these places there are contributions from Algeria, France, Germany, Holland, Hungary, Iran, Oman, Pakistan and Ukraine. The topics include Grid Computing, Web and E-commerce, Performance Modelling Techniques, Power Management and Wireless Networks.

The social events this year will be a workshop dinner at *Med Kitchen* and before that a trip up the Queen's Tower which will provide some welcome exercise and a fantastic view of the London skyline.

Special thanks go to the "volunteer" referees who very kindly agreed to look through all the original submissions: Soraya Zertal, Felipe Franciosi, Richard Hayden and Fernando Martínez Ortuño. Also, we would like to thank Barbara Claxton, Ann Halford and Teresa Ng who helped with the local organisation.

The programme committee consisting of Ashok Argent-Katwala, Nicholas J. Dingle and Uli Harder doubled up as the local organisers, with Ashok sorting out

the most important items: accommodation and the workshop dinner. They were supported by the conference chairs Jeremy Bradley and William Knottenbelt.

And of course we need to thank the steering committee of UKPEW who gave us the opportunity to hold the event at Imperial College London this year:

Irfan Awan (Bradford)	Stephen Jarvis (Warwick)
Jeremy Bradley (Imperial)	Rob Pooley (Heriot-Watt)
Stephen Gilmore (Edinburgh)	Nigel Thomas (Newcastle)

We also need to thank the EPSRC who gave money to this event through a locally held grant (EP/D061717/1). The Department of Computing at Imperial also supported the event by making the room hire very affordable.

We hope you will all enjoy this year's UKPEW and support the event next year at its 25th anniversary.

London, July 2008
The Workshop Co-chairs & Programme Commitee

Contents

Hypercube Communication Structures Analysis via Parametric Petri Nets*

Zaitsev D.A.[1] Shmeleva T.R.[2]

Abstract

A model of a hypercube communication structure of an arbitrary size with an arbitrary number of dimensions in the form of parametric Petri net was constructed. A technique of the linear invariants calculation for parametric Petri nets was developed which allowed the analysis of transmissions involving an arbitrary number of communicating devices (routers) in hypercube. The compulsory buffering of the packets inevitably leads to possible blockings of communicating devices. The structure of complex deadlocks involving an arbitrary number of communicating devices caused by both the chain (cycle) of blockings and the isolation was studied. In real-life networks, described deadlocks lead to considerable decrease of performance.

1 Introduction

As the number of communicating devices and the structure of a network are varying considerably for real-life networks, a technique is required that could manage an arbitrary number of devices constituting an arbitrary structure.

Ajmone Marsan [1] started studying Petri nets composed as a repetition of a basic component for the performance evaluation of CSMA/CD bus LAN. The linear structure was built but the technique of its analysis for an arbitrary number of components was not presented. In [2] the parametric composition of functional Petri nets [3] was applied for the analysis of linear structures of communicating devices. A simpler direct approach [4] was applied to tree-like structures for the verification of switched Ethernet protocols.

The goal of the present paper is the generalization of the approach described in [4] on hypercube communication structure of an arbitrary size with an arbitrary number of dimensions. In [4] the type Petri net model of Ethernet switch with the compulsory buffering of frames was built and used for the composition of tree-like communication structure. As far as routing and switching tables are not represented in the model [4], the same model can describe both switches and routers. We use its modification as the model of a generalized communication device for the composition of hypercube communication structures. The only difference is the number of ports and the disposition of ports on the facets of a hypercube for the further composition of hypercube communication structures (HCS). The direction for future work is the generalization of the obtained results for an arbitrary structure and the modeling of cut-through devices as well.

For the analysis of parametric Petri nets properties linear invariants [5] are used. According to [6,7], an ideal model of a communication system should be bounded, safe and live Petri net. P-invariants allow the proof of boundedness and safeness. But t-invariants do not allow the proof of liveness. That is why special graphs are introduced to prove that the model is not live via deadlocks observation.

* Work supported by NATO grant ICS CLG 982698

1 Odessa National Academy of Telecommunications, Ukraine, zsoftua@yahoo.com

2 Odessa National Academy of Telecommunication, Ukraine, tishtri@rambler.ru

A. Argent-Katwala, N.J. Dingle and U. Harder (Eds.): UKPEW 2008, Imperial College London, DTR08-9 pp. 358–371, http://ukpew.org/

2 Model of Communication Device

Communication devices of the packet-switching networks, such as switches and routers, consist of a finite number of ports and implement the forwarding of the arrived (from a port) packet to the destination port. Packet header information and switch/router address table are used for the calculation of the destination port number. The device implements either the compulsory buffering of packets into its internal buffer or cut-through possibilities. Even cut-through devices employ the buffering when the destination port is busy. Ports work in full-duplex mode providing two channels: for the receiving and the sending of packets; moreover, ports have their own buffers for each channel with the capacity available for the storing of one packet usually.

In the present work we abstract from packet headers and address tables and consider devices with the compulsory buffering only. All the capacities are measured in number of packets.

Let us consider a d dimension space, where $d=1,2,...$. Each communication device is represented by the hypercube of the size 1 in d dimension space. The communication structure composed by connected communication devices constitutes the hypercube of the size k, where $k=1,2,...$ So the total number of devices is $N_{dev}=k^d$. Each device $R^{i_1,i_2,...,i_d}$ has its index $(i_1,i_2,...,i_d)$, where $i_u=\overline{1,k},\ u=\overline{1,d}$. The model of the hypercube communication structure is denoted as $H_{d,k}$. Further we describe the Petri net model of a device and then the composition of a communication structure model via connections of a device with its neighbors.

The model of a hypercube device is denoted as $H_{d,1}$ (number of dimensions equals to d, size of the structure equals to 1). On each facet of a hypercube device $R^{i_1,i_2,...,i_d}$ in d-dimension space a port is situated. So each device has $N_{port}=2\cdot d$ ports; two ports for each dimension are situated at the opposite facets of the hypercube. To denote opposite facets for a dimension j ($j=\overline{1,d}$) the number of the direction is used. The direction is denoted by the variable n; the value $n=1$ is used for the direction to zero in the corresponding dimension and the value $n=2$ is used for the opposite direction to infinity. So the ports may be denoted with the following indices $port_{j,n}^{i_1,i_2,...,i_d}$, where $i_u=\overline{1,k},\ u=\overline{1,d}$, $j=\overline{1,d}$, $n=1,2$. Each port is represented by the two channels (input, output), each channel is represented by a pair of places: one place for the packets buffer, the other – for the buffer capacity. So each port of the device $R^{i_1,i_2,...,i_d}$ is represented by the four following contact places:

$pi_{j,n}^{i_1,i_2,...i_d}$ - input buffer of packets;

$pil_{j,n}^{i_1,i_2,...i_d}$ - capacity of input buffer (equals to 1);

$po_{j,n}^{i_1,i_2,...i_d}$ - output buffer of packets;

$pol_{j,n}^{i_1,i_2,...i_d}$ - capacity of output buffer (equals to 1).

The inside of the device contains $N_{port}+1$ following places. The packets redirected to the port $port_{j',n'}^{i_1,i_2,...,i_d}$ are stored in the corresponding place $pb_{j',n'}^{i_1,i_2,...i_d}$

and one place $pbl^{i_1,i_2,...i_d}$ contains the capacity of the internal buffer, where $j' = \overline{1,d}$, $n' = 1,2$. Notice that the internal buffer is represented by the set of places $pb_{j',n'}^{i_1,i_2,...i_d}$ (one place for each port) to distinguish the number of the destination port given by indices j',n'.

The transitions of the device $R^{i_1,i_2,...,i_d}$ provide the redirection of the input packets from an input port buffer place $pi_{j,n}^{i_1,i_2,...i_d}$ into one of the internal buffer places $pb_{j',n'}^{i_1,i_2,...i_d}, j' \neq j, n' \neq n$ and then the transmission of the packets from the internal buffer place $pb_{j',n'}^{i_1,i_2,...i_d}$ to the output buffer of the target port $po_{j',n'}^{i_1,i_2,...i_d}$. Moreover, the limitations of buffers capacities should be taken into consideration: check and decrease the buffer size at putting the packet into the buffer; increase the buffer size at getting the packet from the buffer. So each port $port_{j,n}^{i_1,i_2,...,i_d}$ of the device $R^{i_1,i_2,...,i_d}$ is supplied by $N_{port} = (N_{port} - 1) + 1$ following transitions:

- one transition for the output channel $to_{j,n}^{i_1,i_2,...i_d}$ with the input arcs from places $pb_{j,n}^{i_1,i_2,...i_d}$, $pol_{j,n}^{i_1,i_2,...i_d}$ and the output arcs to places $po_{j,n}^{i_1,i_2,...i_d}$, $pbl^{i_1,i_2,...i_d}$;

- $N_{port} - 1$ transitions $ti_{j,n,j',n'}^{i_1,i_2,...i_d}$, $j' = \overline{1,d}$, $n' = 1,2$, $j' \neq j$, $n' \neq n$ for the input channel with the input arcs from places $pi_{j,n}^{i_1,i_2,...i_d}$, $pbl^{i_1,i_2,...i_d}$ and the output arcs to places $pb_{j',n'}^{i_1,i_2,...i_d}$, $pil_{j,n}^{i_1,i_2,...i_d}$.

The formal parametric description of the net $H_{d,1}$ is the following:

$$\left(\left(\begin{matrix}\left(ti_{j,n,j',n'} : pi_{j,n}, pbl \rightarrow pb_{j',n'}, pil_{j,n}\right), \\ j' = \overline{1,d},\ n' = 1,2,\ j' \neq j,\ n' \neq n; \\ \left(to_{j,n} : pb_{j,n}, pol_{j,n} \rightarrow po_{j,n}, pbl\right)\end{matrix}\right), j = \overline{1,d}, n = \overline{1,2}\right) \quad (1)$$

If the net $H_{d,1}$ is considered as a model of the device $R^{i_1,i_2,...,i_d}$ in the hypercube structure, the upper indices of its hypercube cell should be added:

$$\left(\left(\begin{matrix}\left(ti_{j,n,j',n'}^{i_1,i_2,...i_d} : pi_{j,n}^{i_1,i_2,...i_d}, pbl^{i_1,i_2,...i_d} \rightarrow pb_{j',n'}^{i_1,i_2,...i_d}, pil_{j,n}^{i_1,i_2,...i_d}\right), \\ j' = \overline{1,d},\ n' = 1,2,\ j' \neq j,\ n' \neq n; \\ \left(to_{j,n}^{i_1,i_2,...i_d} : pb_{j,n}^{i_1,i_2,...i_d}, pol_{j,n}^{i_1,i_2,...i_d} \rightarrow po_{j,n}^{i_1,i_2,...i_d}, pbl^{i_1,i_2,...i_d}\right)\end{matrix}\right), j = \overline{1,d}, n = \overline{1,2}\right).$$

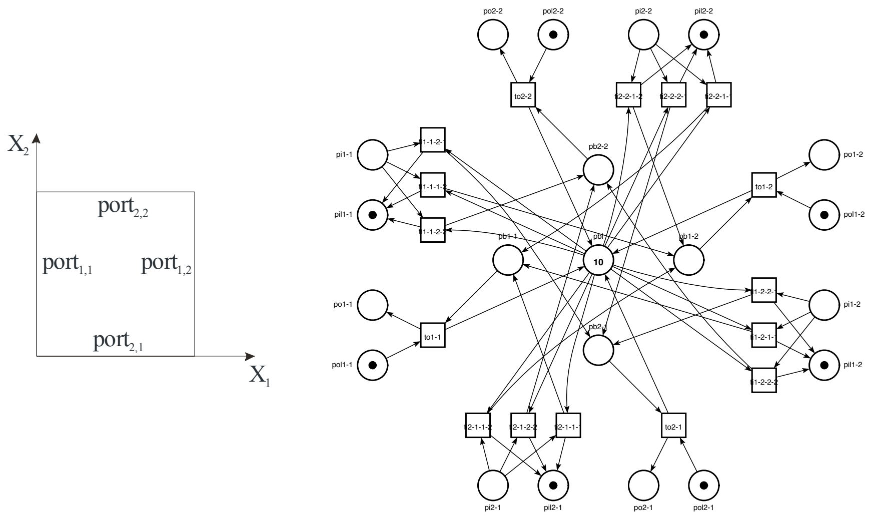
The notation of the transition connections in the form

$$t : px_1,..., px_u \rightarrow py_1,..., py_v$$

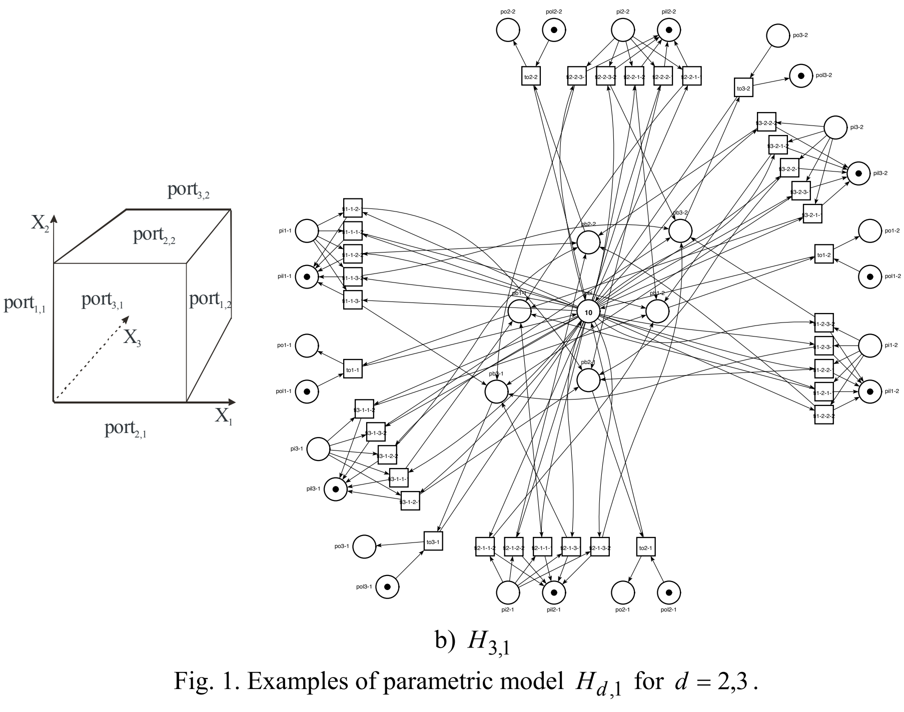
is widely used for Petri nets, for instance, in Tina[8] software. It means that transition t has input arcs from places $px_1,..., px_u$ and output arcs to places $py_1,..., py_v$, where u,v are the numbers of input and output places correspondingly. If necessary, the multiplicities of arcs are added using the notation $p_i * l$, where l is the multiplicity of the corresponding arc connecting p_i and t.

The net represented by (1) is called parametric Petri net because its description has the parameter d for the calculation of the elements indices. The size of the net $H_{d,1}$ is unlimited and represented by the parameter d.

The parametric model $H_{d,1}$ is illustrated by the examples for the following concrete number of dimensions $d = 2,3$ shown in Fig. 1. It is rather difficult to visualize models for larger numbers of dimensions.



a) $H_{2,1}$



b) $H_{3,1}$

Fig. 1. Examples of parametric model $H_{d,1}$ for $d = 2,3$.

3 P-invariants of Communication Device Model

Using the parametric description (1) of the communication device model $H_{d,1}$ given in the previous section the following system was constructed for the calculation of p-invariants:

$$\begin{cases} to_{j,n}: \ \ xpb_{j,n} + xpol_{j,n} = xpo_{j,n} + xpbl, \\ ti_{j,n,j',n'}: \ \ xpi_{j,n} + xpbl = xpb_{j',n'} + xpil_{j,n}, \\ j = \overline{1,d}, \ \ n = 1,2, \ \ j' = \overline{1,d}, \ \ n' = 1,2, \ \ j' \neq j, \ \ n' \neq n. \end{cases} \quad (2)$$

Notice that the system (2) has parametric form; its parameter is the number of dimensions d. System was constructed directly on the description (1) using the usual rule [5] that each equation corresponds to transition and contains sums for its input and output arcs, which are equal. Sums should be calculated using the multiplicities of arcs but all the arcs of (1) have the multiplicity equaling to unit.

The total number of system (2) equations is $N^{t}_{d,1} = 4 \cdot d^2$.

The total number of system (2) variables is $N^{p}_{d,1} = 10 \cdot d + 1$.

To study p-invariants of the model for any number of dimensions the system (2) should be solved in the parametric form. The obtained parametric solution of the system (2) has the following form:

$$\begin{pmatrix} (pi_{j,n}, pil_{j,n}), \ \ j = \overline{1,d}, \ \ n = 1,2; \\ (po_{j,n}, pol_{j,n}), \ \ j = \overline{1,d}, \ \ n = 1,2; \\ (pbl, (pb_{j,n}, \ \ j = \overline{1,d}, \ \ n = 1,2)) \\ ((pb_{j,n}, pi_{j,n}, po_{j,n}), \ \ j = \overline{1,d}, \ \ n = 1,2) \\ (pbl, ((pil_{j,n}, pol_{j,n}), \ \ j = \overline{1,d}, \ \ n = 1,2)) \end{pmatrix} \quad (3)$$

The way of the solutions description is common enough for sparse vectors and especially for the Petri net theory. Only nonzero components are mentioned by the names of the corresponding places. The nonzero multiplier 1 is omitted; in case it is not the unit, the notation $p * x$ is used where x is the value of the invariant for place p. Such notation is adopted in the Tina software [8] which was used for obtaining the Petri net figures in this paper. A line of the matrix (3) gives us a set of lines according to the used indices i, j, n except the last two lines which contain variable number of components given by indices.

A heuristic algorithm was employed for the construction of the matrix (3) but further the proof is presented that (3) is a solution of (1). The fact that (3) is the basis solution is not required for the conclusion about p-invariance of $H_{d,1}$.

The total number of solutions in the matrix (3) is $N^{pinv}_{d,1} = 4d + 3$.

Lemma 1. Each line of the matrix (3) is a solution of the system (2).

Proof. Let us substitute each parametric line of (3) into each parametric equation of the system (2). It gives us the correct statement. At the substitution, the different names of indices are chosen. For instance, let us substitute the fourth line of (3)

$((pb_{l,m}, pi_{l,m}, po_{l,m}), \ \ l = \overline{1,d}, \ \ m = 1,2)$

into the second equation of (2)

$xpi_{j,n} + xpbl = xpb_{j',n'} + xpil_{j,n}, j = \overline{1,d}, \ \ n = 1,2, \ \ j' = \overline{1,d}, \ \ n' = 1,2, \ \ j' \neq j, \ \ n' \neq n$.

For each concrete equation given by valid tupil (j,n,j',n') the solution contains $pi_{j,n}$ at $l=j,m=n$ and $pb_{j',n'}$ at $l=j',m=n'$, moreover the other variables of the equation $xpbl, xpil_{j',n'}$ are not mentioned in the solution. So we obtain:

1+0=1+0 and further 1=1

for each equation.

The first two solutions of (3) make slight difference: they represent series of lines given by their indices. Let us substitute the first parametric line of (3)

$(pi_{l,m}, pil_{l,m}),\ \ l=\overline{1,d},\ \ m=1,2;$

into the second parametric equation of (2)

$xpi_{j,n}+xpbl=xpb_{j',n'}+xpil_{j,n}, j=\overline{1,d},\ \ n=1,2,\ \ j'=\overline{1,d},\ \ n'=1,2,\ \ j'\neq j,\ \ n'\neq n$.

We obtain:

- when $l\neq j$ or $m\neq n$: 0+0=0+0 and further 0=0;
- when $l=j$ and $m=n$: 1+0=0+1 and further 1=1.

In the same way all the 5x2 combinations are checked. ■

Theorem 1. The net $H_{d,1}$ is a p-invariant Petri net for an arbitrary natural number d .

Proof. Let us consider the sum of the fourth and the fifth lines of the matrix (3) which represent the solutions of the system (2) according to Lemma 1:

$((pb_{j,n}, pi_{j,n}, po_{j,n}),\ \ j=\overline{1,d},\ \ n=1,2)$

plus

$(pbl,((pil_{j,n}, pol_{j,n}),\ \ j=\overline{1,d},\ \ n=1,2))$

equals to

$$(pbl,((pil_{j,n}, pol_{j,n}, pb_{j,n}, pi_{j,n}, po_{j,n}),\ \ j=\overline{1,d},\ \ n=1,2)) \qquad (4)$$

As all the $N^{p}_{d,1}=10\cdot d+1$ places are mentioned in this invariant, the net $H_{d,1}$ is a p-invariant Petri net for an arbitrary natural number d . Moreover, as each component of (4) equals to the unit, the net $H_{d,1}$ is a safe and bounded Petri net for an arbitrary natural number d . ■

4 Composition of HCS

The connections of communication devices in the hypercube are provided by the fusion (union) of the corresponding contact places of neighbor devices.

Let us consider an internal communication device $R^{i_1,\dots i_j,\dots,i_d}$, $i_u=\overline{2,k-1}$, $u=\overline{1,d}$, $j=\overline{1,d}$:

– places of $port^{i_1,\dots,i_j,\dots,i_d}_{j,1}$ are fused with the corresponding places of $port^{i_1,\dots,i_j-1,\dots,i_d}_{j,2}$, device $R^{i_1,\dots i_j-1,\dots,i_d}$ in such a way that place $po^{i_1,\dots,i_j,\dots i_d}_{j,1}$ is fused with $pi^{i_1,\dots,i_j-1,\dots i_d}_{j,2}$, place $pol^{i_1,\dots,i_j,\dots i_d}_{j,1}$ – with $pil^{i_1,\dots,i_j-1,\dots i_d}_{j,2}$, place $pi^{i_1,\dots,i_j,\dots i_d}_{j,1}$ – with $po^{i_1,\dots,i_j-1,\dots i_d}_{j,2}$, place $pil^{i_1,\dots,i_j,\dots i_d}_{j,1}$ – with $pol^{i_1,\dots,i_j-1,\dots i_d}_{j,2}$;

– places of $port_{j,2}^{i_1,...,i_j,...,i_d}$ are fused with the corresponding places of $port_{j,1}^{i_1,...,i_j+1,...,i_d}$, device $R^{i_1,...i_j+1,...,i_d}$ in such a way that place $po_{j,2}^{i_1,...,i_j,...i_d}$ is fused with $pi_{j,1}^{i_1,...,i_j+1,...i_d}$, place $pol_{j,2}^{i_1,...,i_j,...i_d}$ – with $pil_{j,1}^{i_1,...,i_j+1,...i_d}$, place $pi_{j,2}^{i_1,...,i_j,...i_d}$ – with $po_{j,1}^{i_1,...,i_j+1,...i_d}$, place $pil_{j,2}^{i_1,...,i_j,...i_d}$ – with $pol_{j,1}^{i_1,...,i_j+1,...i_d}$.

To avoid duplicity, the names of the places for the zero direction ports $n=1$ will be considered with respect to the current device, for the infinity direction ports $n=2$ – with respect to the neighbor devices and their zero direction ports $n=1$. So the names of the fusion places have only the indices of the zero direction ports $n=1$. Moreover, to simplify further notations, the places with the indices of the infinity direction ports $n=2$ on the facets (borders) of the communication hypercube are named with respect to non existing devices with the indices equaling to $k+1$. So the names of ports with the indices of the infinity direction $n=2$ do not appear in the hypercube. The communication hypercube structure described above is denoted as $H_{d,k}$. An example of $H_{d,k}$ for $d=3, k=4$ is represented in Fig. 2.

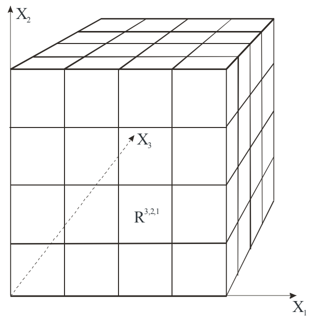


Fig. 2. Scheme of the communication structure $H_{3,4}$

The formal description of $H_{d,k}$ composition is given with the following:

$$\left(\left(\begin{array}{l} pi_{j,1}^{i_1,...,i_j+1,...i_d} := po_{j,2}^{i_1,...,i_j,...i_d} \cup pi_{j,1}^{i_1,...,i_j+1,...i_d} \\ pil_{j,1}^{i_1,...,i_j+1,...i_d} := pol_{j,2}^{i_1,...,i_j,...i_d} \cup pil_{j,1}^{i_1,...,i_j+1,...i_d} \\ po_{j,1}^{i_1,...,i_j+1,...i_d} := pi_{j,2}^{i_1,...,i_j,...i_d} \cup po_{j,1}^{i_1,...,i_j+1,...i_d} \\ pol_{j,1}^{i_1,...,i_j+1,...i_d} := pil_{j,2}^{i_1,...,i_j,...i_d} \cup pol_{j,1}^{i_1,...,i_j+1,...i_d} \end{array}\right), i_u=\overline{1,k-1}, u=\overline{1,d}, j=\overline{1,d}\right),$$

$$\left(\left(\begin{array}{l} pi_{j,1}^{i_1,...,i_j+1,...i_d} := po_{j,2}^{i_1,...,i_j,...i_d} \\ pil_{j,1}^{i_1,...,i_j+1,...i_d} := pol_{j,2}^{i_1,...,i_j,...i_d} \\ po_{j,1}^{i_1,...,i_j+1,...i_d} := pi_{j,2}^{i_1,...,i_j,...i_d} \\ pol_{j,1}^{i_1,...,i_j+1,...i_d} := pil_{j,2}^{i_1,...,i_j,...i_d} \end{array}\right), i_u=\overline{1,k-1}, u=\overline{1,d}, j=\overline{1,d}, u \neq j, i_j=k\right).$$

The union sign $\cup$ denotes the fusion of places; the left column gives new names of places.

5 P-invariants of HCS

Using the abstract description of the communication hypercube model $H_{d,k}$ given in the previous section, the following system is constructed for the calculation of p-invariants:

$$\begin{cases} to_{j,1}^{i_1,\ldots,i_d}: \; xpb_{j,1}^{i_1,\ldots,i_d} + xpol_{j,1}^{i_1,\ldots,i_d} = xpo_{j,1}^{i_1,\ldots,i_d} + xpbl^{i_1,\ldots,i_d}, \\ ti_{j,1,j',n'}^{i_1,\ldots,i_d}: \; xpi_{j,1}^{i_1,\ldots,i_d} + xpbl^{i_1,\ldots,i_d} = xpb_{j',n'}^{i_1,\ldots,i_d} + xpil_{j,1}^{i_1,\ldots,i_d}, \\ to_{j,2}^{i_1,\ldots,i_j,\ldots,i_d}: \; xpb_{j,2}^{i_1,\ldots,i_j,\ldots,i_d} + xpil_{j,1}^{i_1,\ldots,i_j+1,\ldots,i_d} = xpi_{j,1}^{i_1,\ldots,i_j+1,\ldots,i_d} + xpbl^{i_1,\ldots,i_j,\ldots,i_d}, \\ ti_{j,2,j',n'}^{i_1,\ldots,i_j,\ldots,i_d}: \; xpo_{j,1}^{i_1,\ldots,i_j+1,\ldots,i_d} + xpbl^{i_1,\ldots,i_j,\ldots,i_d} = xpb_{j',n'}^{i_1,\ldots,i_j,\ldots,i_d} + xpol_{j,1}^{i_1,\ldots,i_j+1,\ldots,i_d}, \\ \qquad j=\overline{1,d},\; j'=\overline{1,d},\; n'=1,2,\; j'\neq j,\; n'\neq n,\; i_u=\overline{1,k},\; u=\overline{1,d}. \end{cases} \tag{5}$$

The total number of system (5) equations is $N_{d,k}^{t} = 4 \cdot d^2 \cdot k^d$.

The total number of system (5) variables is $N_{d,k}^{p} = (6 \cdot d + 1) \cdot k^d + 4 \cdot d \cdot k^{d-1}$.

The obtained parametric solution has the following form:

$$\begin{pmatrix} (pi_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}, pil_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}),\; j=\overline{1,d},\; (i_u=\overline{1,k},\; u=\overline{1,d},\; u\neq j),\; i_j=\overline{1,k+1}; \\ (po_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}, pol_{j,1}^{i_1,\ldots i_j,\ldots,i_d}),\; j=\overline{1,d},\; (i_u=\overline{1,k},\; u=\overline{1,d},\; u\neq j),\; i_j=\overline{1,k+1}; \\ (pbl^{i_1,\ldots,i_d}, (pb_{j,n}^{i_1,\ldots,i_d}, j=\overline{1,d},\; n=1,2)),\; i_u=\overline{1,k},\; u=\overline{1,d}; \\ ((pb_{j,n}^{i_1,\ldots,i_d}, n=1,2, j=\overline{1,d}, i_u=\overline{1,k}, u=\overline{1,d}),\; ((pi_{j,1}^{i_1,\ldots,i_d}, po_{j,1}^{i_1,\ldots,i_d}), j=\overline{1,d}, i_u=\overline{1,k}, u=\overline{1,d}), \\ \qquad ((pi_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}, po_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}), j=\overline{1,d},\; i_u=\overline{1,k},\; u=\overline{1,d},\; u\neq j,\; i_j=k+1)) \\ ((pbl^{i_1,\ldots,i_d}, ((pil_{j,1}^{i_1,\ldots,i_d}, pol_{j,1}^{i_1,\ldots,i_d}), j=\overline{1,d}),\; i_u=\overline{1,k},\; u=\overline{1,d}), \\ \qquad ((pil_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}, pol_{j,1}^{i_1,\ldots,i_j,\ldots,i_d}), j=\overline{1,d},\; i_u=\overline{1,k},\; u=\overline{1,d},\; u\neq j,\; i_j=k+1)) \end{pmatrix} \tag{6}$$

The total number of solutions is $N_{d,k} = (1 + 2 \cdot d) \cdot k^d + 2 \cdot d \cdot k^{d-1} + 2$.

Lemma 2. Each line of the matrix (6) is a solution of the system (5).

Theorem 3. The net $H_{d,k}$ is a p-invariant Petri net for arbitrary natural numbers d, k.

The proofs of Lemma 2 and Theorem 3 were done in the same way as for the net $H_{d,1}$ (Section 3).

6 Adding Models of Terminal Devices

The communication devices are attached to each other constituting a communication structure but they are created only for the packets transmission among the terminal devices: workstations and servers. In the present work, the client-server technique of interaction is not studied, so the types of terminal devices are not distinguished. An abstract terminal device provides at least two basic functions: send packet and receive packet. These basic functions are implemented in the model represented in Fig. 3.

The model contains an internal buffer of the packets qb; transition si models the receiving of the packets, while transition so models the sending. The model keeps the balance of input and output packets; the limitation of buffer qb size is not considered.

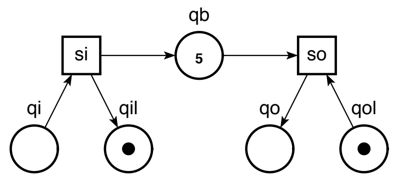


Fig. 3. Petri net model of a terminal device

The terminal devices shown in Fig. 3 are attached to the border ports of the hypercube structure; the corresponding model is denoted as $HT_{d,k}$. The terminal device in hypercube structure is denoted as $A^{i_1,...i_j,...,i_d}$, where $i_u = \overline{1,k}, u = \overline{1,d}, j = \overline{1,d}, u \neq j, i_j \in \{1,k\}$ and attached to the communication device $R^{i_1,...i_j,...,i_d}$. The formal description of $HT_{d,k}$ composition using $HT_{d,k}$, $A^{i_1,...i_j,...,i_d}$ is given with the following:

$$\left(\begin{pmatrix} pi_{j,1}^{i_1,...,i_j,...i_d} := qo^{i_1,...,i_j,...i_d} \cup pi_{j,1}^{i_1,...,i_j,...i_d} \\ pil_{j,1}^{i_1,...,i_j,...i_d} := qol^{i_1,...,i_j,...i_d} \cup pil_{j,1}^{i_1,...,i_j,...i_d} \\ po_{j,1}^{i_1,...,i_j,...i_d} := qi^{i_1,...,i_j,...i_d} \cup po_{j,1}^{i_1,...,i_j,...i_d} \\ pol_{j,1}^{i_1,...,i_j,...i_d} := qil^{i_1,...,i_j,...i_d} \cup pol_{j,1}^{i_1,...,i_j,...i_d} \end{pmatrix}, i_u = \overline{1,k}, u = \overline{1,d}, j = \overline{1,d}, u \neq j, i_j = 1\right),$$

$$\left(\begin{pmatrix} pi_{j,1}^{i_1,...,i_j+1,...i_d} := qi^{i_1,...,i_j,...i_d} \cup pi_{j,1}^{i_1,...,i_j+1,...i_d} \\ pil_{j,1}^{i_1,...,i_j+1,...i_d} := qil^{i_1,...,i_j,...i_d} \cup pil_{j,1}^{i_1,...,i_j+1,...i_d} \\ po_{j,1}^{i_1,...,i_j+1,...i_d} := qo^{i_1,...,i_j,...i_d} \cup po_{j,1}^{i_1,...,i_j+1,...i_d} \\ pol_{j,1}^{i_1,...,i_j+1,...i_d} := qol^{i_1,...,i_j,...i_d} \cup pol_{j,1}^{i_1,...,i_j+1,...i_d} \end{pmatrix}, i_u = \overline{1,k}, u = \overline{1,d}, j = \overline{1,d}, u \neq j, i_j = k\right).$$

In the same way as in Section 5 it was proven that $HT_{d,k}$ is a p-invariant Petri net for any given natural numbers d and k.

7 T-invariants and Deadlocks of HCS

For the calculation of t-invariants the same approach is applied. The only difference is that for t-invariants each equation corresponds to place and variables correspond to transitions. It gains us that the Petri net $H_{d,k}$ is not t-invariant, but it is quite trivial because the modeled system is open as the terminal devices are not attached. It was proven that the model of closed system with attached terminal devices $HT_{d,k}$ is a t-invariant Petri net for arbitrary natural numbers d, k. But the consistency of the model does not imply its liveness.

Each pair of neighbor communication devices can fall into a local deadlock, for instance, when the device $R^{i_1,...,i_j,...,i_d}$ got l packets directed to the device $R^{i_1,...,i_j+1,...,i_d}$ and the device $R^{i_1,...,i_j+1,...,i_d}$ got l packets directed to the device $R^{i_1,...,i_j,...,i_d}$ and, moreover, the input and output buffers of their common port are occupied with the packets, where l is the limitation of the internal buffer size (marking of places $pbl^{i_1,\dots,i_j,\dots,i_d}$, $pbl^{i_1,\dots,i_j+1,\dots,i_d}$). Such a situation constitutes the t-dead marking for the transitions of both devices while other transitions of the net $HT_{d,k}$ are potentially live.

But the structure of all the possible deadlocks is more sophisticated. We show that deadlocks occur either in cycles (chains) of blockings involving a few communicating devices (where the pair is a particular case) or because of isolation with surrounding deadlocks.

For the description of complex deadlocks of the net $HT_{d,k}$, the graph $GH_{d,k}$ of connections is constructed. In the graph $GH_{d,k}$ each node corresponds to communication device and has arcs directed to its neighbors. An example of internal node connections for $GH_{3,k}$ is shown in Fig. 4. An arc with two arrows denotes two arcs of opposite directions.

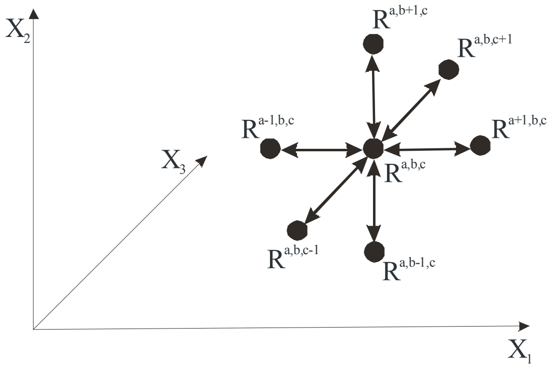


Fig. 4. Node connections of the graph $GH_{3,k}$

A directed simple cycle in the graph $GH_{d,k}$ represents a deadlock of the communication hypercube $HT_{d,k}$. In a deadlock cycle, each arc connecting a pair of neighbor devices $R^{i_1,\dots,i_u,\dots,i_d}$, $R^{i_1,\dots,i'_u,\dots,i_d}$, $|i_u - i'_u| = 1$ means that $R^{i_1,\dots,i_u,\dots,i_d}$ blocks itself iff it got l packets directed to $R^{i_1,\dots,i'_u,\dots,i_d}$, its output buffer of the port connecting $R^{i_1,\dots,i_u,\dots,i_d}$ with $R^{i_1,\dots,i'_u,\dots,i_d}$ contains a packet and the device $R^{i_1,\dots,i'_u,\dots,i_d}$ is blocked also. When the cycle ends, the last device of it blocks itself because the first device is blocked and can not receive packets.

Let us prove that all the transitions of a blocked device $R^{i_1,\dots,i_u,\dots,i_d}$ are dead. For distinctness we denote

$$r = \begin{cases} 1, i_u - i'_u = -1, \\ 2, i_u - i'_u = 1. \end{cases}$$

All the transitions $ti^{i_1,\dots,i_d}_{j,n,j',n'}$ are dead because marking of their input place $pbl^{i_1,\dots,i_d}$ equals to zero, so the device cannot receive packets. All the transitions $to^{i_1,\dots,i_d}_{j,n}$, $n \neq r$ are dead because each of their input places $pb^{i_1,\dots,i_d}_{j,n}$ has zero marking. The transition $to^{i_1,\dots,i_d}_{u,r}$ is dead because marking of its input place $pol^{i_1,\dots,i_d}_{u,r}$

is zero. So the device cannot send packets. Notice that marking of $pol_{u,r}^{i_1,\dots,i_d}$ cannot be changed because $R^{i_1,\dots,i'_u,\dots,i_d}$ is blocked.

Non-simple cycles of $GH_{d,k}$ represent deadlocks also. For instance, if device $R^{i_1,\dots,\dots,i_u,\dots,i_v,\dots,i_d}$ belongs to two simple cycles and it got two output arcs directed to $R^{i_1,\dots,\dots,i'_u,\dots,i_v,\dots,i_d}$ and $R^{i_1,\dots,\dots,i_u,\dots,i'_v,\dots,i_d}$ it means that $R^{i_1,\dots,\dots,i_u,\dots,i_v,\dots,i_d}$ blocks itself and $R^{i_1,\dots,\dots,i'_u,\dots,i_v,\dots,i_d}$, $R^{i_1,\dots,\dots,i_u,\dots,i'_v,\dots,i_d}$ are blocked also. In this case $R^{i_1,\dots,\dots,i_u,\dots,i_v,\dots,i_d}$ blocks itself having a_u packets directed to $R^{i_1,\dots,\dots,i'_u,\dots,i_v,\dots,i_d}$ and a_v packets directed to $R^{i_1,\dots,\dots,i_u,\dots,i'_v,\dots,i_d}$, where $a_u + a_v = l$, and moreover each corresponding output port buffer of $R^{i_1,\dots,\dots,i_u,\dots,i_v,\dots,i_d}$ contains a packet when $a_u > 0$ ($a_v > 0$). Inductive reasoning gives the proof for d simple cycles.

The other kind of deadlocks is induced by the isolation of a device by deadlocks containing all its neighbor devices. It can be done with one simple cycle as well. For instance, in $GH_{3,k}$, the following cycle R^{i_1-1,i_2,i_3}, R^{i_1-1,i_2+1,i_3}, R^{i_1,i_2+1,i_3}, R^{i_1,i_2+1,i_3+1}, R^{i_1,i_2,i_3+1}, R^{i_1+1,i_2,i_3+1}, R^{i_1+1,i_2,i_3}, R^{i_1+1,i_2-1,i_3}, R^{i_1,i_2-1,i_3}, R^{i_1,i_2-1,i_3-1}, R^{i_1,i_2,i_3-1}, R^{i_1-1,i_2,i_3-1}, R^{i_1-1,i_2,i_3} contains all the neighbors of R^{i_1,i_2,i_3} (R^{i_1-1,i_2,i_3}, R^{i_1,i_2+1,i_3}, R^{i_1,i_2,i_3+1}, R^{i_1+1,i_2,i_3}, R^{i_1,i_2-1,i_3}, R^{i_1,i_2,i_3-1}), so the device R^{i_1,i_2,i_3} is blocked because of isolation. The isolation of a node can be generalized on the blocking of a simple chain by the isolation of its last node.

So a deadlock is a chain of blockings where the last node is blocked because:

1) it coincides with the first node;
2) it belongs to other deadlock;
3) it is isolated by other deadlocks.

It is very significant that occurred deadlocks create more possibilities for new deadlocks occurring. So the process has avalanche-like character. A full deadlock involving all the devices (and all the transitions) occurs when cycles (chains) contain all the devices in the hypercube. It requires at least $(l+1)\cdot k^d$ packets provided by the terminal devices. But if isolations of devices occur a little number of packets is required.

In spite of the fact that rather sophisticated hypercube communication structures were studied, the described deadlocks in the chains (cycles) of blockings and isolations are hard-nosed for real-life communication graphs, where devices with the compulsory buffering are used. We believe that these deadlocks may be purposely inflicted by the specially situated generators of the peculiar traffic. In real-life networks, the blocking of the devices is overcome with the time-out mechanisms causing the cleaning of the buffers but it leads to the considerable fall of the network performance as soon as the situation is repeated by the special generators of perilous traffic.

8 Conclusions

Thus, in the present paper, the technique of the linear invariants calculation for parametric Petri nets with the regular structure was presented. The technique was studied on the example of a communication hypercube of an arbitrary size with an arbitrary number of dimensions.

The application of the technique allowed the analysis of transmissions, involving an arbitrary number of communicating devices. The modeled telecommunication device constitutes a generalized router/switch with the compulsory buffering of the packets. Such positive properties of the communication structure as safeness and consistency were obtained using the linear invariants of infinite Petri nets.

It was grounded that the compulsory buffering of the packets inevitably leads to the possible blockings of communicating devices. The structure of the complex deadlocks involving an arbitrary number of communicating devices caused by both the chain (cycle) of blockings and the isolation was studied.

Though, in real-life networks, the deadlocks are overcome by the cleaning of the buffers via the time-out mechanism, it leads to the considerable decrease of the network performance and moreover might be inflicted by the ill-intentioned traffic.

References

[1] Marsan A.M., Chiola G., Fumagalli A. An Accurate Performance Model of CSMA/CD Bus LAN // Advances in Petri Nets, LNCS, 1987, Vol. 266, P. 146-161.

[2] Zaitsev D.A., Zaitsev I.D. Verification of Ethernet protocols via parametric composition of Petri net // INCOM'2006: 12th IFAC/IFIP/IFORS/IEEE/IMS Symposium Information Control Problems in Manufacturing, May 17-19 2006, Saint-Etienne, France, p. 261-267.

[3] Zaitsev D.A. Functional Petri Nets. Universite Paris-Dauphine, Cahier du Lamsade 224, Avril 2005, 62p (www.lamsade.dauphine.fr/cahiers.html).

[4] Shmeleva T.R. Verification of switched Ethernet protocols using infinite Petri nets // COMINFO'2007: Third International conference on Modern information-communication technologies, September 24-28, 2007, Levadia, Ukraine, p. 161-163.

[5] Murata T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE, April 1989, Vol. 77, p. 541-580.

[6] Berthelot G., Terrat R. Petri Nets Theory for the Correctness of Protocols // IEEE Trans. on Communications, no. 12, 1982, vol. 30, p. 2497-2505.

[7] Diaz M. Modelling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Model // Computer Networks, no 6, 1982, p. 419-441.

[8] Berthomieu B., Ribet O.-P., Vernadat F. The tool TINA - construction of abstract state space for Petri nets and Time Petri nets // International Journal of Production Research, Vol. 42, no. 4, 2004 (http://www.laas.fr/tina).