Performance Modelling of a Computer Integrated Manufacturing and Management System

Jozef B. Lewoc[1*], Antoni Izworski[2], Slawomir Skowronski[2]

[1]Design, Research and Translation Agency Leader (Leading Designer)
 53-138 Wroclaw, ul. Powst. Sl. 193/28, Poland; *leader@provider.pl
[2]Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeze Wyspianskiego 27, Poland

Abstract. This paper presents a predicted target architecture of an integrated manufacturing and management system, based on a metropolitan-type infranet and an industrial process control and monitoring network. Such systems are severely required by prospective users, especially large-scale manufacturing enterprises. Nevertheless, neither big manufacturers of computer integrated manufacturing systems nor big manufacturers of computer integrated systems are apt to develop combined computer integrated manufacturing and management systems. Considering that, a team of volunteer design and research workers initiated some work oriented toward facilitating future development of such combined systems. Since the combined systems will be novel ones, prior results of research work conducted for the manufacturing systems or the management systems separately will not be valid, since the actual requirements will have to cover the needs of both manufacturing and managerial applications. This refers to, among other things, to performance evaluation. To evaluate performance, a method developed for packet switching networks with end-to-end acknowledgement was applied. The network is modelled as a set of closed routes consisting of a user and a series of service stations (communication links, switches, host processes). The paper describes the investigations carried out for the case study. Some consideration is given to the performance evaluation accuracy, basing on the validation work results obtained from analytical work, simulation and measurements on the Polish pilot wide area network.

Introduction

The Computer-Integrated Manufacturing and Management (CIMM) systems are severely needed by prospective users, especially big and medium manufacturing enterprises. In spite of that, big manufacturers and vendors of Integrated Management (IM) systems, having at their disposal financial resources sufficient to develop and implement at least pilot CIMM systems, are no apt to enter the manufacturing domain [1,2]. The basic reason for that seems to be the fact that the big IM system manufacturers and vendors do not possess at their disposal the designers and implementers educated and experienced enough to cope with both manufacturing and management issues. Such persons are available, paradoxically many of them are available in poorly developed countries where the designers must have possess a wide scope of experience in order that they are successful [3], but they are dispersed in various industrial and/or research organisations and they do not have at their disposal the financial resources needed to develop novel IT systems of the CIMM type.

In such circumstances, a reasonable duty of research and development workers seems to be to carry out the initial work on the prospective project of CIMM systems, even if no financing is available for such work. This work may be considered as a volunteer work or a work for the society that have financed their earlier research and development work.

A team of IT experts and process technologists was established and worked out an approach to development of CIMM systems that seems to be feasible, rather fast and economic [4,5,6,7]. The next step that can and should be made by the research and development workers even before the CIMM project is established is that of performance evaluation of probable hardware struc-

tures.

The task of performance evaluation is important in the case of CIMM since the project concerns a novel IT system for which nobody possesses the sense of performance and no rules of thumb concerning performance could have been developed earlier. The performance evaluation issues connected with development of CIMM systems is the subject matter of the present paper.

The authors believe that a specific in-depth example is more educational than a general study, in particular in the context described above, and, therefore, the form of a case study has been adopted for the present paper.

The following section presents the architecture of the CIMM system under investigation.

1 Exemplary CIMM Architecture

The pilot Polish CIMM system has been designed several years ago for the biggest then Polish manufacturer of household appliances, Polar, Wroclaw (the Enterprise), employing several thousand people. The CIMM issues were analysed for the Enterprise already in the early nineties, but the project was abandoned because of the severe down economy period in Poland in the nineties. In conformity with the early design work, it was assumed that the target organisational structure of the Enterprise would be that depicted in Figure 1.
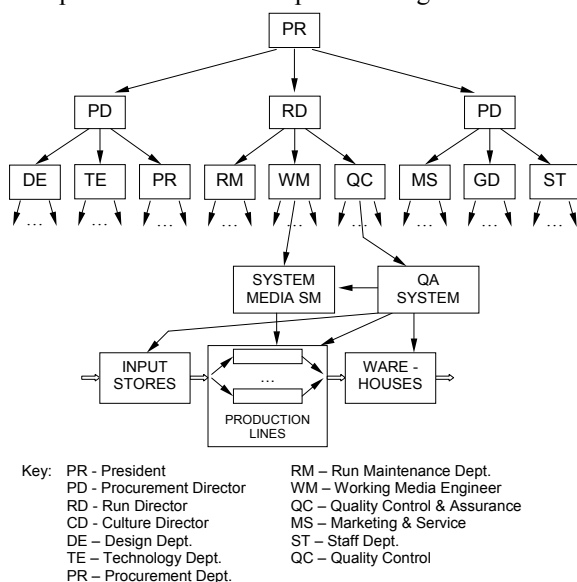

Key: PR - President   RM – Run Maintenance Dept.
      PD - Procurement Director   WM – Working Media Engineer
      RD - Run Director   QC – Quality Control & Assurance
      CD - Culture Director   MS – Marketing & Service
      DE – Design Dept.   ST – Staff Dept.
      TE – Technology Dept.   QC – Quality Control
      PR – Procurement Dept.

Figure 1. General organisational diagram of the enterprise.

For the general Enterprise organisation presented in Figure 1, the hardware architecture of the CIMM sys-

tem, depicted in Figure 2, has been proposed.

The architecture is based on the infranet metropolitan network operating under the TCP/IP protocol suite [8] and on the process control and monitoring network of the LonWorks type [9].
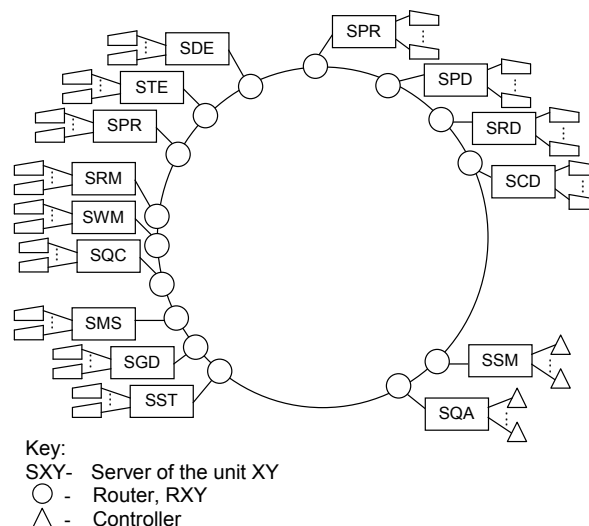

Key:
SXY-   Server of the unit XY
○ -   Router, RXY
△ -   Controller

Figure 2. Hardware architecture of the CIMM system

2 The Network under Investigation

It is assumed that the time needed for transfer of data via any switch (server, gateway, node or router) is insignificant in comparison with the time needed to transfer the data via a co-operating data link. This is in conformity with the specifications of actual hardware proposed for the CIMM system since data is transferred via network switches as 8-bit byte blocks transferred via fast direct memory access (DMA) channels and internal processing of the data is usually connected only with the message headers.

However, the method proposed enables to consider the delays in network switches though the calculations may be somewhat more complicated.

With the assumption that the delays in transmission links are insignificant, the hardware structure of the CIMM system may be presented as in Figure 2. The network depicted in Figure 2 is composed of 30 switches (15 routers and 15 servers). Let the individual switches and data links be numbered as shown in Figure 3 (server Nos: 1- 15, router Nos: 16-30, link pairs (for both transmission directions): (1,2) – (85,86).
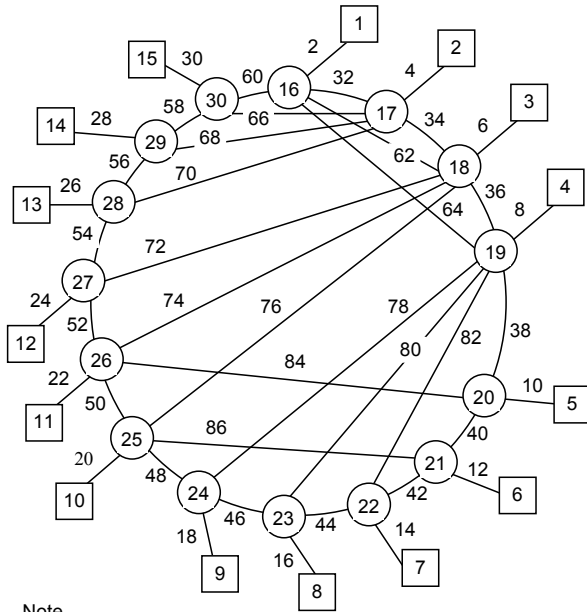
Note
Link number, i =2k, k=1,2,..,43 is, in fact, a pair of numbers, 2k and 2k-1; the first is the number of the link directed from the node of the lower number to that of the higher one and the other in the opposite direction.

Figure 3. Numbering of links and nodes.

Upon the network of Figure 3, there is stretched a set of closed routes. Exemplary closed routes in the CIMM network are the connections between a Marketing & Service Department worker and the general enterprise database (the President's database) (closed route No. s = 166; ref. Figure 4a) and between a technologist and the system Media database (closed route No. $s = 279$; ref. Figure 4b).
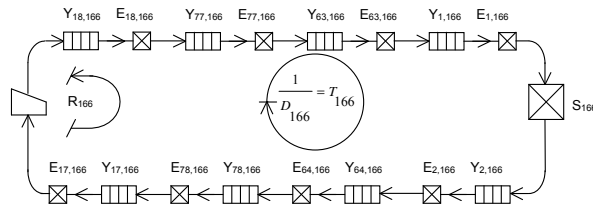
a) Exemplary closed route of a Marketing & Service employee using general enterprise database
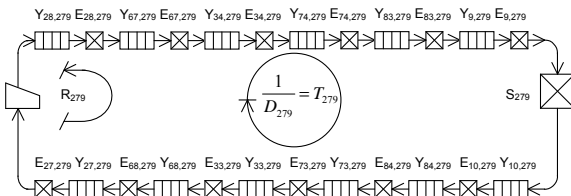
b) Exemplary closed route of a technologist using the System Media

Key:

$$s \in S = 1, \dots, 299; \; i \in I = 1, \dots, 86$$

Rs = round route (trip) delay for s-th closed route
T_s = s-th closed route cycle time
D_s = s-th closed route throughput
$Y_{i,s}$ = waiting time of s-th closed route to i-th link
$E_{i,s}$ = service time of s-th closed route to i-th link
M_s = s-th closed route thinking time
S_s = s-th closed loop ultimate service time

Note: Upper case letters denote random variables while lower case letters – relevant mean values

Figure 4. Exemplary closed routes.

Note: it is assumed that any *p*-th server, $p \in 1, \dots, 15$ (ref. Figure 3) is able to support 20 closed routes of numbers $s = (p-1)\,20, \dots, (p-1)\,20 + 19$.

Note also that the network described here as a set of closed routes may be considered as a case of the Kelly networks [5] where the entity (packet) sojourn time in any switch is a function of the number of packets in all routes in the network. However, the classic queuing theory does not provide solutions for the Kelly networks [10].

In theory, the problem may be solved by the stochastic queuing system methods. Unfortunately, the stochastic queuing system methods are not suitable for actual network designers and, even more, closed networks always produce severe problems [11].

Another class of performance evaluation methods are Petri nets, either deterministic or stochastic [12]. However, Petri nets are only formalisms and do not provide anything of merits, not covered by the basic classic or stochastic performance evaluation methods.

Therefore, it was decided to adapt the approximate performance evaluation method [13], developed and validated earlier for computer and communication networks with end-to-end acknowledgement, to the cases of the CIMM system and to apply the method for performance evaluation for the case study under discussion.

3 The Approximations

The reference [13] is hardly available now. Therefore, some basic assumptions and results will be repeated in the present paper.

In addition to the denotations defined in Figure 3, the following will be used in the present paper:

A_i is the set of closed routes beginning at the i-th link, such that the $s(i)$-th user is connected directly to the i-th link:

$$A_i = \{s_{i,1}, s_{i,2}, ..., s_{i,a_i}\} \tag{1}$$

B_i is the set of closed routes passing via the i-th link but not beginning at that link:

$$B_i = \{z_{i,1}, z_{i,2}, ..., z_{i,b_i}\} \tag{2}$$

The link preceding the i-th link in the s-th closed route is denoted by $h = hs(i)$, provided that $s \in B_i$.

The set of closed routes parallel to the s-th closed route in the i-th link is defined by (3).

$$C_i = B_i \cap (A_h \cup B_h) \tag{3}$$

The power of C_i is given by (4).

$$\overline{C_i} = c_i \tag{4}$$

The mean cycle time of closed routes encountered by the s-th route at the i-th link is approximated by (5).

$$\bar{t}_{i,s} = \frac{a_i + b_i - 1}{\displaystyle\sum_{l \in \{A_i \cup B_i\}/\{s\}} \frac{1}{t_i}}; \frac{0}{0} = 0; \tag{5}$$

The mean waiting time of the closed routes encountered by the s-th route at the i-th link is given by (6).

$$\bar{y}_{i,s} = \frac{\bar{t}_{i,s}}{(a_i + b_i - 1)} \left(\sum_{l \in \{A_i \cup B_i\}\backslash\{s\}} \frac{y_{i,l}}{t_i} \right); \tag{6}$$

The mean service time for the closed routes encountered by the s-th route at the i-th link is given by (7).

$$\bar{e}_{i,s} = \frac{\bar{t}_{i,s}}{(a_i + b_i - 1)} \left(\sum_{l \in \{A_i \cup B_i\}\backslash\{s\}} \frac{e_{i,l}}{t_l} \right); \tag{7}$$

The mean thinking time for the closed routes encountered by the s-th route at the i-th link is given by (8).

$$\bar{m}_{i,s} = \frac{\bar{t}_{i,s}}{(a_i + b_i - 1)} \left(\sum_{l \in \{A_i \cup B_i\}\backslash\{s\}} \frac{e_{i,l}}{t_l} \right); \tag{8}$$

The definitions of the mean values defined above for the set of closed routes that are not parallel to the s-th route are as follows:

$$\bar{t}'_{i,s} = \frac{a_i + b_i - c_i}{\displaystyle\sum_{l \in \{A_i \cup B_i\}\backslash C_i} \frac{1}{t_i}}; \tag{9}$$

$$\bar{y}'_{i,s} = \frac{\bar{t}'_{i,s}}{a_i + b_i - c_i} \sum_{l \in \{A_i \cup B_i\}\backslash C_i} \frac{y_{i,l}}{t_i}; \tag{10}$$

$$\bar{e}'_{i,s} = \frac{\bar{t}'_{i,s}}{a_i + b_i - c_i} \sum_{l \in \{A_i \cup B_i\}\backslash C_i} \frac{e_{i,l}}{t_l}; \tag{11}$$

$$\bar{m}'_{i,s} = \frac{\bar{t}'_{i,s}}{a_i + b_i - c_i} \sum_{l \in \{A_i \cup B_i\}\backslash C_i} \frac{m_{i,l}}{t_l}; \tag{12}$$

For all closed routes, the balance equations (13) have been defined. The basic reasons for the balance equations are as follows.

The first set of equations in (13) is obvious: the cycle time, t_s, for any closed loop is a sum of the thinking time, m_s, all waiting times, $y_{i,s}$, and all transmission times, $e_{i,s}$, for the closed loop involved (ref. the examples in Figure 4).

For the two other equation sets in (13), the unknown mean values are approximated by the mean values of uniformly distributed variables and the probabilities that any entity (packet) is in any specific state (thinking, waiting for transmission, transmission) is approximated by the mean duration for that state divided by the mean cycle time for the variable under consideration. Another assumption is that there may exist one and only one entity (packet) in any closed loop of the network.

The second equation set in (13) refers to the case of an entity (packet) of the s-th closed route beginning at the i-th link. This entity may find there an entity of any closed route passing via the i-link with the probability

$(a_i+b_i-1) \dfrac{\bar{e}_{i,s}}{\bar{t}_{i,s}\left(1 - \dfrac{e_{i,s}}{t_s}\right)}$ (the latter dividend describes the condition that the entity (packet) under consideration is not in the state of transmission).

The expression in the parenthesis in the second equation set of (13) is a sum of three terms, denoted here by $e_{i,s}(A + B + C)$. After multiplying, the first term is the mean value of the time from the instant that the entity of the s-th closed loop finds the entity in transfer till the instant that the latter transmission is completed, i.e. $\dfrac{\bar{e}_{i,s}}{2}$ in accordance with the assumptions accepted.

The second term approximates the queue that may have been gathered during the time interval that the entity (packet) found in transfer via the *i*-th link (there are $(a_i + b_i - 2)$ eligible candidates), provided that the possible candidate is not in the thinking state (the quotient $(1 - \dfrac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}})$).

The third term approximates the queue that may have gathered during the time interval from the instant that the entity encountered in the *i*-th link has been in transfer till the instant that the entity of the *s*-th closed loop appears at the *i*-th link (there are $((a_i + b_i) - 1)$ eligible candidates to be multiplied by the mean probability the candidate is in the waiting state $(\dfrac{\overline{y}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}})$ provided that it is not in the state of thinking $(1 - \dfrac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}})$.

The third equation set in (13) describes the routes passing via the *i*-th link but not beginning there. It is approximated that any loop parallel to the *s*-th one is of no impact on the waiting time spent by any entity of the *s*-th closed loop at the *i*-th link (the complete share of the parallel loops in the waiting time of the loop under consideration is included at the beginning link). Therefore, the averaging in this equation set is done only for the closed loops that are non-parallel to the one under consideration (formulae (9)-(12)). Except of that, this set of equations is analogous to the second one (note that in this case, the minimum value of c_i equals to 1 (formulae (3), (4)).

The set of equations (5) – (13) enables to compute iteratively the basic unknown mean values of the network performance, i.e. the closed route cycle time, t_s, the mean throughput in the *s*-th closed route, $d_s = \dfrac{1}{t_s}$, and the round-trip delay, $r_s = t_s - m_s$.

$$\mathbf{I}_{s \in S}\ (t_s = \sum_{i \in I_s}(y_{i,s} + e_{i,s}));$$

(13)

where *S* is the set of all closed loops, $\overline{\overline{S}} = v$.

$$\mathbf{I}_{s \in A_i}\ (y_{i,s} = \frac{(a_i + b_i - 1)\overline{e}_{i,s}^2}{\overline{t}_{i,s}(1 - \dfrac{e_{i,s}}{t_s})}(\frac{1}{2} +$$

$$+ (a_i + b_i - 2)(1 - \frac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}})\frac{\overline{y}_{i,s}}{\overline{t}_s - \overline{e}_{i,s}} +$$

$$(a_i + b_i - 1)(1 - \frac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}})\frac{\dfrac{\overline{e}_{i,s}}{2}}{\overline{t}_{i,s} - \overline{e}_{i,s}});$$

$$\mathbf{I}_{s \in B_i}\ (y_{i,s} = \frac{(a_i + b_i - c_i)\overline{e}_{i,s}'^2}{\overline{t}_{i,s}'(1 - \dfrac{e_{i,s}}{t_s})}(\frac{1}{2} +$$

$$(a_i + b_i - c_i - 1)(1 - \frac{\overline{m}_{i,s}'}{\overline{t}_{i,s}' - \overline{e}_{i,s}'})\frac{\overline{y}_{i,s}'}{\overline{t}_{i,s}' - \overline{e}_{i,s}'} +$$

$$+ (a_i + b_i - c_i)(1 - \frac{\overline{m}_{i,s}'}{\overline{t}_{i,s}' - \overline{e}_{i,s}'})\frac{\dfrac{\overline{e}_{i,s}'}{2}}{\overline{t}_{i,s}' - \overline{e}_{i,s}'});$$

4 Validation of Approximations

Validation with accurate results for cyclic queuing systems

The problem of performance evaluation of homogeneous star-topology networks is the same as that investigated for cyclic queuing networks. The accurate solutions in the form of limit probabilities of the entity (packet, request) numbers have been known since early seventies. These results, for exponential thinking time, were employed to validate the approximations presented above.

Several hundred comparisons were done for the number of the customers (closed routes), v, changing between 2 and several dozen. The relative error of the mean cycle time, t_s, calculated for the set of comparison reached its maximum less than 0.03 at $v \approx 14$ and decreased for v tending to zero or to infinity.

Validation with simulation

In order that the approximations may be validated, a fast WAN simulator was developed on the basis of earlier simulation studies for real-time computer control systems. This simulator was used, primarily, for the Polish pilot wide area network MSK. However, both actual and planned network configurations (including those to operate at much higher transmission rates) were investigated. In addition, available foreign simulation results for local and wide area networks were also used for validation.

The number of validation experiments was higher that 500, with the number of closed loops and links equal up to 500 and 100, respectively. The maximum relative error found was lower than 0.08. However, in the case of results obtained for MSK, where the fair conditions of comparison could be ensured (this, obviously, was not true for foreign simulation investigation), the maximum relative error of the cycle time (and throughput) was lower than 3.5.

Validation with measurements

To investigate MSK and the approximations presented above, an internal communication network measuring tool Sitwa was developed and implemented.

It was used, primarily, to validate the simulation results on the existing possible configurations of MSK. The investigations showed that the simplifying assumptions in simulation (e.g. omission of the flow control packets and/or frames) did not result in significant simulation errors. The maximum relative error of the approximations under discussion did not exceed 0.05.

5 Exemplary Results

The set of equations (5) ÷ (13) was solved iteratively. The basic user characteristics, r_s and d_s, for the exemplary closed routes, $s = 166$ and $s = 279$, are presented in Figure 5 ÷ 8.

The closed route, $s = 166$, passes via the most severely loaded links, $i = 1$ and $i = 2$. This results in that r_{166} reaches more than 4 s at m_s values close to 0 (ref. Figure 5). If it is assumed that an annoying value (i.e. the value that can not be accepted by the Enterprise employees) of rs is that higher than 2 s, then the m_s values below some 0.5 s should be avoided.

| s	I_s	s	I_s
1÷18	{Ø}	160	16,79,35,76,83,11.12,86,75,36,80,15
19	2,62,74,83, 9,10,84,73,61,1	161÷178	18,77,63,1,2,64,78,17
20	2,62,76,85,11,12,86,75,61,1	179	18,77,35,74,83,9,10,84,73,36,78,17
21÷38	4,31,1,2,32,3	180	18,77,35,76,85,11,12,86,75,36,78,17
39	4,34,74,83,9,10,84,73,33,3	181÷198	20,75,71,2,1,62,76,19
40	4,34,76,85,11,12,86,75,33,3	199	20,75,74,83,9,10,84,73,76,19
41-58	6,61,1,2,62,5	200	20,85,11,12,86,19
59	6,74,83,9,10,84,73,5	201-218	22,73,61,2,1,62,72,21
60	6,76,85,11,12,86,75,5	219	22,83,9,10,84,21
61÷78	8,63,1,2,64,7	220	22,73,75,85,11,12,86,75,72,21
79	8,35,73,83,9,10,84,74,36,7	221-238	24,71,61,2,1,62,72,23
80	{Ø}	239	24,71,74,83,9,10,84,73,72,23
81÷98	{Ø}	240	24,71,76,85,11,12,86,75,72,23
100	{Ø}	241÷258	28,69,31,1,2,32,70,25
101÷118	{Ø}	259	26,69,34,74,83,9,10,84,73,33,70,25
119	{Ø}	260	26,69,34,76,85,11,12,86,75,33,70,25
120	{Ø}	261÷278	28,67,31,1,2,32,68,27
121÷138	14,81,63,1,2,64,82,13	279	28,67,34,74,83,9,10,84,73,33,68,27
139	14,81,35,74,83,9,10,84,73,36,82,13	280	28,67,34,76,85,11,12,86,75,33,68,27
140	14,81,35,76,85,11,12,86,75,36,82,13	281÷298	30,65,31,1,2,32,66,29
141÷158	16,79,63,1,2,64,80,15	299	30,65,34,74,83,9,10,84,73,33,66,29
159	16,79,35,74,83,9,10,84,73,36,80,15	300	30,65,34,76,85,11,12,86,75,33,66,89

Table 1. I_s versus s
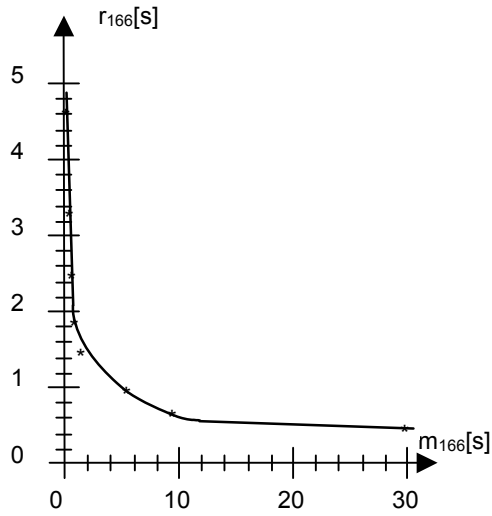
Figure 5. r_{166} versus m_{166}.

The d_{166} values (the mean throughput values depicted in Figure 6) show a definite saturation (congestion) at m_s values below ca. 0.5 s. If the mean offered load for the 166-th closed route is defined by (14).
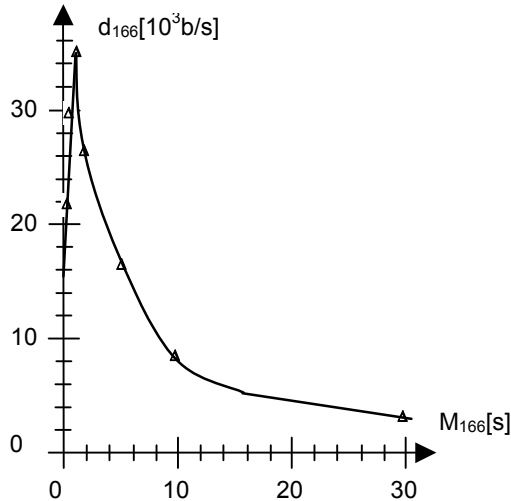

Figure 6. d_{166} versus m_{166}.

$$t_{166}^{off} = \sum_{i \in 166} (e_{i,s} + m_s) \tag{14}$$

Then the increased offered load results in the decreased mean throughput, d_{166}, for the given scenario and for $m_s < 0.5\,s, s = 1, \dots, 300$.

The closed route, $s = 279$ (Figure 7), shows only the acceptable r_s values (below 1 s while the values not greater than 2 s are acceptable as not annoying ones) and the ds value is decreasing monotonically with the offered load increasing ($m_s \rightarrow 0, s = 1, \dots, 300$) (Figure 8). The reason is that the 279-th closed route does not pass via any link under heavy traffic. Therefore, even when the mean thinking time is decreased towards zero, no saturation (congestion) occurs in the links passed by the 279-th closed route.
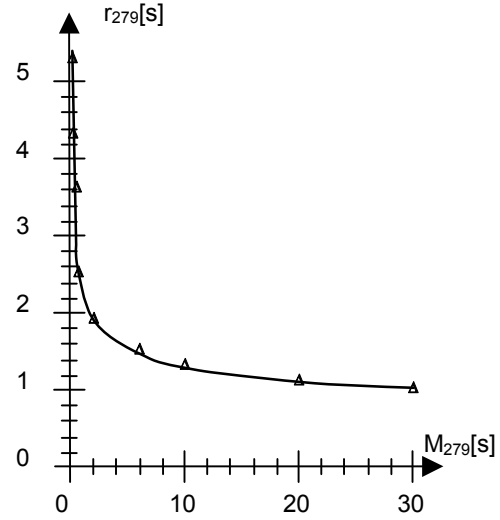

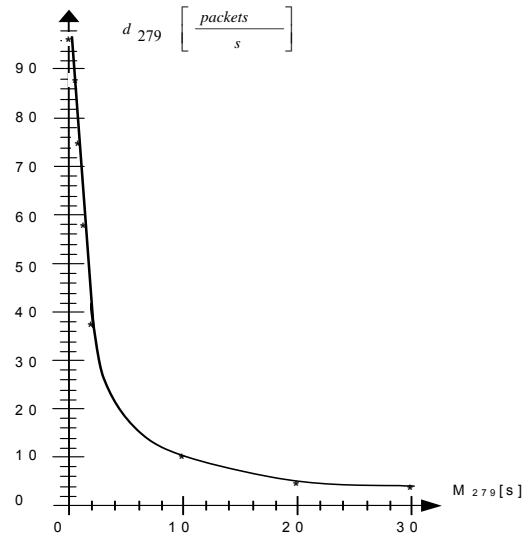**Figure 7.** r_{279} versus m_{279}

Figure 8. d_{279} versus m_{279}

6 Final Remarks

The authors do not claim that the method devised and verified some dozen years ago needs no further work. On the contrary, they do realise that the method validated primarily on the pilot Polish Interuniversity Computer Network MSK, operating at rather low transmission rates and in a limited configuration, needs further validation and possible tuning. In addition, the method may and should be upgraded: thanks to the rapid growth of the computing power that has happened during the last decade, more accurate approximations may be done for the balance equations (e.g., the simplifying assumption that an entity (packets) may find only one entity of one closed loop in the queue to some link) may be discarded. Then the balance equations become more complicated but may be still solved by the iterative method. And the networks with sliding windows or credit-based flow control may be modelled directly, instead of modelling several loops for credits bigger than 1, as it has been done earlier.

However, the authors believe that the method may and should be published in its original form now. Any protest against it, raised by the queuing theory experts pointing at some possible formal deficiencies of the method, would sound rather false: in a general case, the classic queuing theory methods have failed and have not produced any useful tools for performance evaluation of actual computer and communication networks, severely needed by network designers, developers and operators. The authors are designers and/or consultants for actual computer systems and networks. To do their jobs in a proper way, they had to get involved in some research work in the domain of performance evaluation. Any further work is, however, within the duty range of the queuing theory people.

References

[1] Leader, Technical correspondence, Wroclaw, 2006.

[2] Ridler R. Proposal for CIM, Technical correspondence of Leader, Wroclaw, 2000.

[3] Blach LK, et al. In West-East approaches of automation: A case study, Cultural Aspects of Automation '91 (IFAC), Krems 1991.

[4] Franasik L, et al. System Media – a Step toward Computer Integrated Manufacturing and Management systems, *Modelling, Measurement and Control (AMSE Press)*, Vol. 22, No. 3: 37-46, 2001.

[5] Izworski A, Lewoc JB, Skowronski S. Development of Computer Integrated Manufacturing and Management Systems, In: *Information Control Problems in Manufacturing*, Saint Etienne, 2006.

[6] J.B. Lewoc, A. Izworski and S. Skowronski, ALSIS case study: computer integrated manufacturing and management systems, In *Applications of Large Scale Industrial Systems*, (IFAC), Helsinki-Stockholm, 2006.

[7] J.B. Lewoc, Performance Evaluation Problems for Actual Computer Networks, AMSE Monographs, Tassin, 1990.

[8] http://www.acm.org/crossroads/xrds1-1/tcpjmy. ht ml

[9] http://www.echelon.com

[10] Robertazzi TG. Computer Networks and Systems. Queueing Theory and Performance Evaluation, Springer-Verlag, 2000.

[11] Surfozo R. Introduction to Stochastic Networks, Springer-Verlag, 1999.

[12] German R., Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets, Wiley, 2000.

[13] Lewoc JB. Performance Evaluation Problems for Actual Computer Networks, AMSE Monographs, Tassin, 1990.