J. B. Lewoc, Antoni Izworski, Slawomir Skowronski, Peter Kopacek / IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org ISSN : 2250-3021
Vol. 2 Issue 1, Jan.2012, pp.175-179

An Actual Communication Network Performance Evaluation Method

J. B. Lewoc*, Antoni Izworski, Slawomir Skowronski** , Peter Kopacek*****

*Design, Research and Translation Agency Leader (Leading Designer), 53-138 Wroclaw, ul. Powst. Sl. 193/28, Poland
**Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeze Wyspianskiego 27, Poland
***Viena Technical University, A-1040 Vienna, Favoritenstrs. 9-12, Austria

Abstract—The queuing system and network theories may be hardly applied to solve performance evaluation problems for actual large-scale computer systems and networks. This is true especially for high performance communication networks such as wireless, satellite or photonic networks. Firstly, the reason is that the theoretical assumptions can hardly be fulfilled in actual networks. Secondly, the methods are too complex and hardly understandable for actual network designers. Thirdly, the well-known methods do not teach too much possible designers of their solutions, in contrast to e.g. simulation modelling. Fourthly, case studies of performance evaluation solutions for large-scale communication networks can hardly be found in the literature. The paper presents an approximate method developed early in the computer network era by a team of the first Polish network designers and is a proposal for an approach for solving significant actual large-scale network design problems.

I. INTRODUCTION

From the beginning of their activity in design and implementation of computer automation systems and networks [1], the design and implementation team (the Team) in the case study country (Poland) had at their disposal the hardware and software tools obsolete of roughly 5, 10 years in comparison to those available to designers in well developed countries. An implication of that fact was that performance evaluation was a must for the Team.

The queuing system and network theory asked from the very beginning [2]In this situation, the design (and implementation) team of Polish pioneering systems simply had to be involved in performance evaluation. It should be noted here that even if they had access to performance data for similar Western systems, the data would not be useful for the Polish systems due to the obsolete tools being applied in the latter. Therefore, from the very beginning, performance evaluation work was initiated, often against the wishes/orders of the project management staff.

From the very beginning [2], the queuing theory asked the questions of interest for actual computer system designers; however very severe difficulties problems were experienced to find the answers. The basic drawbacks of the theory were the missed real-time operation condition considerations and the assumption of independency of the input and output streams, even in nice works devoted (at least, in theory) to computer applications [3-4].

The Team do not know the general theory for investigation of real-time problems in computer systems /networks. However, a part of the stream independence assumptions were removed in the closed queuing network theory, including the first general computation method, MVA [5], [6]. Unfortunately, the assumptions of the method can hardly be met by actual, especially large-scale computer networks.

Later on, no, in practice, calculation method was developed in the area. Some "good advices" can be heard or read: apply Petri nets, or stochastic Markov chains. The first induces shame that such incompetent peoples may be referees for prestigious events in computer networks (Petri nets are a presentation tool that, by itself, may do nothing in solving difficult performance evaluation/prediction problems [7]). The other method is a theoretical one and is useless for actual network designers [8].

There is also a problem of usefulness of the analytical methods proposed: application of the methods (assuming that they are approximate ones) does not teach the computer system and network designers too much concerning the solutions adapted since it is very difficult to understand the isomorphism between the computer network and the queuing network (if there is any) under analysis. Such clear isomorphism exists e.g. between the computer network and the simulation model [9] what has made it possible for the first Polish computer automation system designers to learn a lot about the product designed by them from developing the simulation model.

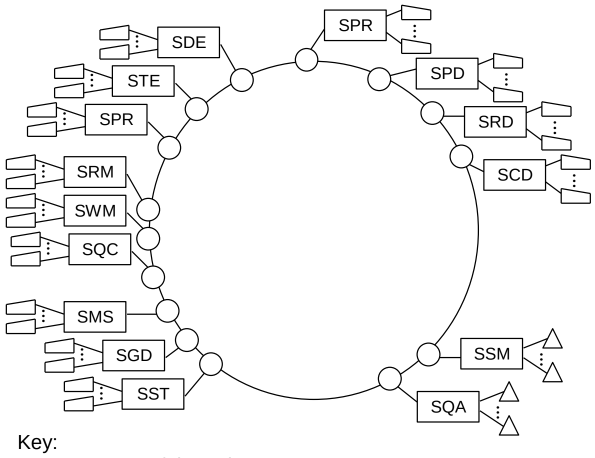
To cope with the actual performance problems, the Team developed a computer automation system simulation tool (event-driven) [9], [10] and used it for performance investigation of a dozen or so pioneering computer automation systems/networks in the case study country. Nevertheless, the team did not stop trying to find references on possible analytical methods that could be used for performance prediction for actual computer networks and e.g. entered into cooperation with a mathematician commonly recognized then as the best Polish one involved in investigation of queuing networks. The aim was to predict the performance of the first-Polish WAN of the public network character, MSK, quite well measured by the team [11]. The mathematician proposed an analytical method, the assumptions were compared with the reality of the network and the method was rejected as an incompatible one. After a year, the mathematician gave up and said: you have to go your own way [12]. And the Team worked out an approximate actual network performance evaluation method (Method) intended for investigation of MSK of the WAN type, so the method covers also the detailed algorithms for star-networks and LANs.

The development of novel systems and networks in the case study country was nearly blocked in the nineties [13], [14] and the applications of the own method was suspended; it was resumed at early years of the XXI-st Century when it became needed for investigation novel computer systems and networks designed by the team, including the Computer Integrated Manufacturing and Management systems (CIMMs) severely needed and expected by many manufacturing enterprises [15], [16], [17].

In the following sections, the Method will be discussed for a general case of the metropolitan Wide Area Network (WAN)

II. Exemplary WAN description

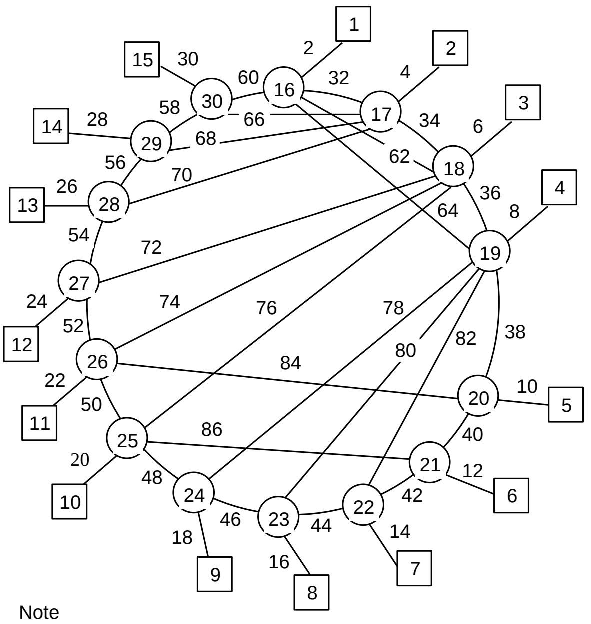
The case study description is based on the initial design work done for the biggest then Polish manufacturer of household appliances (the Enterprise) of the target network architecture shown in Fig. 1. The architecture is based on the infranet metropolitan network operating under the TCP/IP protocol and on the process control and monitoring network.



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

Figure 1. Hardware architecture of the CIMM system

Assuming that the delays in transmission links are insignificant, the hardware structure of the CIMM system may be presented as in Fig. 1. The exemplary network is composed of 30 switches (15 routers and 15 servers). Let the individual switches and data links be numbered as shown in Fig. 2 (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.

Fig. 2. Numbering of links and nodes

Upon the network of Fig. 2, there is stretched a set of closed routes (calls, connections). Exemplary closed routes are the connections between a Marketing & Service Department worker and the general Enterprise database (the President's database) (closed route No. s = 166; ref. Fig. 3a) and between a technologist and the system Media database (closed route No. s = 279; ref. Fig. 3b)

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

III. The approximations

In addition to the denotations defined in Fig. 2, the following will be used:

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:

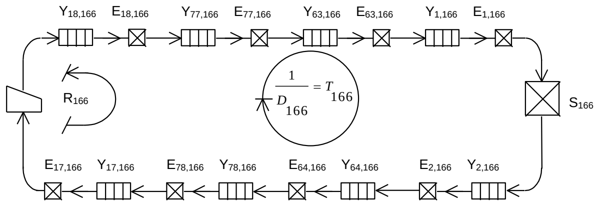
$$A_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,a_i}\} \qquad (1)$$

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

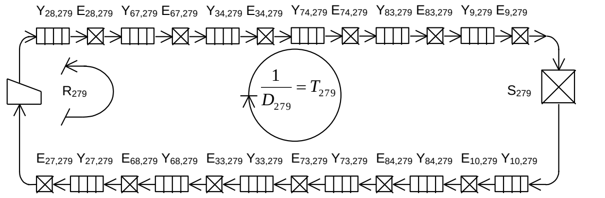
$$B_i = \{z_{i,1}, z_{i,2}, \dots, z_{i,b_i}\} \qquad (2)$$

The link preceding the *i*-th link in the *s*-th closed route is denoted by $h = h_s(i)$, 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).



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:

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

R_s= 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 3. Exemplary closed routes

$$C_i = B_i \cap (A_h \cup B_h) \quad (3)$$

The power of C_i is given by (4).

$$\overline{C_i} = c_i \quad (4)$$

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

$$\bar{t}_{i,s} = \frac{a_i + b_i - 1}{\sum_{l\in\{A_i\cup B_i\}/\{s\}} \frac{1}{\bar{t}_i}}; \frac{0}{0} = 0; \quad (5)$$

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

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

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

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

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

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

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:

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

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

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

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

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 Fig. 3).

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

$$\bigcap_{s\in S}(t_s = \sum_{i\in I_s}(y_{i,s} + e_{i,s})); \qquad (13)$$

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

$$r_s = t_s - m_s.$$

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

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

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

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

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

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

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. $\frac{\overline{e}_{i,s}}{2}$ 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-$\frac{\overline{m}_{i,s}}{\overline{t}_{i,s}-\overline{e}_{i,s}}$)).

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 ($\frac{\overline{y}_{i,s}}{\overline{t}_{i,s}-\overline{e}_{i,s}}$) provided that it is not in the state of thinking $(1-\frac{\overline{m}_{i,s}}{\overline{t}_{i,s}-\overline{e}_{i,s}})$.

The third equation set in (13) describes the routes passing via the i-th link but not beginning there.

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, $d_s = \frac{1}{t_s}$, and the round-trip delay,

IV. Validation of the Approximations

A. Validation with accurate results for cyclic queuing systems

The Method was validated with available data for cyclic queuing networks. The 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, $t_{s,}$, never exceeded 0.03..

B. Validation with simulation

In order that the approximations may be validated, a fast event-driven WAN simulator was developed [9], [10]. 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 of the cycle time (and throughput) was lower than 3.5.

C. Validation with measurements

For MSK, it was possible to design and implement and internal traffic generator and measuring tool Sitwa and to take many performance measurements. The first Polish Interuniversity Computer Network project (an academic one) included expressly research work on computer networks that resulted in many useful results such as internal measurements that could be much more complex and useful than external measurements [18], [19], [20]. For several hundred comparative experiments for the Method and internal measurements, the maximum relative error did not exceed 0.05.

Final remarks

The case study investigation are an evidence that the method may be successfully applied for investigation of actual computer networks. Its simplicity, in comparison with the well-known stochastic queuing networks makes it a possible tool for network designers. The low relative error measured in the validation work is better than needed for the design work. The low calculation complexity (the number of iterations needed to obtain performance results within the interval of 1% never reached 10; computer time was below 10% that needed for simulation investigation for the same scenarios) prove that the Method may be successfully applied for computer networks much larger than MSK.

Nevertheless, is must be remembered that the Method was developed by actual computer system/network designers and not researchers, and it needs further investigations and tuning. It is presented here as a proposal for wireless, satellite and photonic network designers and researchers to encourage them to apply it or to develop their own, similar self-made-man approach to performance evaluation of their networks since the support from performance evaluation theoreticians seems to be rather unreasonable.

A frequent and fast response from theoreticians is: “Stupid engineers”. The leading designer of the Team likes to have such pronoun: An engineer is someone who can do useful things and stupid, in the scientific circles, often means the one who does not use a heave cannon to shoot a housefly. But, except of Electronics, he graduated in Mathematics with high honour (in the times of Prof. H. Steinhaus times when the Wroclaw School of Mathematics was ranked on the very top of the world Mathematics). This, plus his engineering achievements as the leading designer of a dozen or so successful large-scale Polish pioneering applications of computer systems/networks plus the more than 40-year long experience in studying references in performance evaluation makes it worth to think before you use this nickname.

9 References

[1] W. Wojsznis and J. Lewoc, Sistema uprawlenija prodwizeniem metalla w prokatnom ceche, IV-th International Conference of Comecon Countries on Production Process Automation in Black Metallurgy, Publication I/33, Zaporoze 1971 (In Russian).
[2] Takacs L., Introduction to the theory of queues, Oxford University Press, 1962.
[3] L. Kleinrock, Queuing Systems. Volume 1. Theory, John Wiley and Sons, 1975.
[4] . Kleinrock, Queuing Systems. Volume 2. Computer Applications, John Wiley and Sons, 1977.
[5] Reiser, M., Mean-value Analysis and Convolution Method for Queue-dependent Servers in Closed Queueing Networks, Performance Evaluation 1, North Holland, 1981.
[6] T.G., Robertazzi, Computer Networks and Systems. Queueing Theory and Performance Evaluation, Springer-Verlag, 2000.
[7] German R., Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets, Wiley, 2000.
[8] R. Surfozo, Introduction to Stochastic Networks, Springer-Verlag, 1999.
[9] Lewoc J., Simulation of a Computer Control System, Applicationes Mathematicae, XII, 3/1971/
[10] Lewoc J.B., Analysis of operation of digital control systems through simulation on digital computers (Ph.D. Thesis), Warsaw Technical University, 1972 (In Polish).
[11] J.B. Lewoc, Performance Evaluation Problems for Actual Computer Networks, AMSE Monographs, Tassin, 1990.
[12] T. Rolski, Private message, Wroclaw 1988.
[13] J.B. Lewoc, A Case Study: Practical Ethical Aspects for Technology Transfer and Change in Computer Control System Development. In: IFAC Congress, Prague 2005.
[14] J.B. Lewoc J.B. et al., A Case Study: History of Polish Computer Applications in Power System Control, Submitted to IFIP Congress, Milano 2008 (unpublished).
[15] Jozef B. Lewoc, Antoni Izworski, Slawomir Skowronski, Antonina Kieleczawa and Georgi Dimirovski, Computer Integrated Manufacturing & Management Systems - An Emerging Application of Control and Management, ETAI, Ohrid, 2009..
[16] A. Izworski, J.B. Lewoc and S. Skowronski, Development of Computer Integrated Manufacturing and Management Systems, In: Information Control Problems in Manufacturing, Saint Etienne, 2006.
[17] 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.
[18] W. Czajkowski, J. B. Lewoc, M. Slugocki and W. Strzalkowski, A communication subnetwork simulation and performace evaluation tool, Modelling Techniques and Tools for Performance Analysis (INRIA, ACM), Paris 1985.
[18] W. Czajkowski, J. B. Lewoc, M. Slugocki and W. Strzalkowski, A communication subnetwork simulation and performace evaluation tool, Modelling Techniques and Tools for Performance Analysis (INRIA, ACM), Paris 1985.
[19] W Czajkowski et al., Initial Performance Evaluation for a Communication Subnetwork Node, International Conference on Modelling Techniques and Tools for Performance Evaluation (INRIA, ACM), Paryz, 1984 (North Holland).
[20] Lewoc J.B, A. Izworski and S. Skowronski, An Internal Internet Traffic Modelling / Performance Measuring Tool. In: Mathematical Modelling (IMACS). Vienna 2006.