

# ERLANG C FORMULA AND ITS USE IN THE CALL CENTERS

Erik CHROMY<sup>1</sup>, Tibor MISUTH<sup>1</sup>, Matej KAVACKY<sup>1</sup>

<sup>1</sup> Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic

chromy@ktl.elf.stuba.sk, misuth@ktl.elf.stuba.sk, kavacky@ktl.elf.stuba.sk

**Abstract.** This paper deals with calculation of important parameters of the Call Center using the Erlang C formula and then results are verified through simulations. Erlang C formula is defined as a function of two variables: the number of agents  $N$  and the load  $A$ . On the base of their values it is possible to determine the probability  $P_C$ , that the incoming call will not be served immediately, but it will have to wait in the waiting queue. Simulations satisfy the assumption of Markov models.

## Keywords

Call Center, Erlang C Formula, Markov Models, Quality of Service.

## 1. Introduction

Call Center is dynamical, technical system (package of technical equipments – hardware, software and human sources) designed for effective connecting people with the requirements for service with operator or with systems able to satisfy their requirements. The core of the Call Center is Automatic Call Distribution (ACD).

Each of the components of the ACD can be described with some precision by means of mathematical tools and causalities. Since the ACD systems process a large number of incoming requests, the majority of models is based on the principles of mathematical statistics. The right choice of a statistical model is able to ensure the sufficient accuracy of the results. It is essential to describe the dependency of input variables and parameters that can greatly affect the accuracy of the results. The modeling of Call Center parameters is possible through Markov models, but also through Erlang formulas.

This paper deals with calculation of important parameters of the Call Center (which affect proper operation of such queuing system) using the Erlang C formula and then results are verified through simulations. These simulations satisfy the assumption of Markov models.

### 1.1 Erlang C formula and M/M/m/∞ model

Immediate rejection of call by reason of occupation of all agents (as expected in Erlang B formula) is in terms of provided services by Call Center inappropriate solution. This shortness is eliminated in second Erlang formula – Erlang C. In the case that the call cannot be served immediately, the call is placed into the waiting queue with unlimited length. If the release of one of the agents happens, it is automatically assigned to the following call from the queue. If the waiting queue is empty, the agent is free and he waits for next call.

Erlang C formula [5] is in the original form defined as a function of two variables: the number of agents  $N$  and the load  $A$ . On the base of their values it is possible to determine the probability  $P_C(1)$ , that the incoming call will not be served immediately, but it will have to wait in the waiting queue.

$$P_C(N, A) = \frac{A^N N}{N!(N-A)}, \quad (1)$$

$$\sum_{i=0}^{N-1} \frac{A^i}{i!} + \frac{A^N N}{N!(N-A)}$$

where

$$A = \frac{\lambda}{\mu}. \quad (2)$$

Now, we use the relationship between load  $A$  (2), the average number of calls per time  $\lambda$  and the average number of requests processed per time  $\mu$ . Next, we define the variable  $\eta$ , which represents the load of 1 agent as [2, 3, 7]:

$$\eta = \frac{\lambda}{N\mu}. \quad (3)$$

By substituting (2) and (3) into equation (1) we have:

$$P_C(N, \eta) = \frac{\left(\frac{\lambda}{\mu}\right)^N N}{N! \left(N - \left(\frac{\lambda}{\mu}\right)\right)} = \frac{(N\eta)^N}{N!(1-\eta)} \quad (4)$$

$$= \frac{\sum_{i=0}^{N-1} \frac{\left(\frac{\lambda}{\mu}\right)^i}{i!} + \frac{\left(\frac{\lambda}{\mu}\right)^N N}{N! \left(N - \left(\frac{\lambda}{\mu}\right)\right)} = \sum_{i=0}^{N-1} \frac{(N\eta)^i}{i!} + \frac{(N\eta)^N}{N!(1-\eta)}$$

what corresponding with the relation for the probability that the request in queuing system M/M/m/∞ will be placed into the queue, i.e. in the system is more than  $m$  requests [2, 3, 7]. Now it is analytically derived, that the Erlang C model (1) and the Markov model M/M/m/∞ (4) are identical.

It is possible by using of the basic form for the Erlang C formula (1) to calculate the value of parameter  $A$  (maximum load) at a known number of agents  $N$  and the probability of waiting  $P_C$ . Due to complexity of the analytical expression of these unknown parameters, numerical method for solving is used.

By adding the waiting queue into the system, many other parameter variables that can be monitored and also affected will appear. An important factor in terms of caller is queue waiting time. This value is random variable described by distribution function [7]:

$$F_W(\tau) = 1 - P_C \cdot e^{-\mu(N-A)\tau}, \tau > 0. \quad (5)$$

Then it is possible to calculate the average queue waiting time  $W$  (average call waiting time in the queue before assigning a call to agent):

$$W = \frac{P_C}{\mu(N-A)}. \quad (6)$$

and by applying of Little theorem [7] and formula (2) we get the average number of requests in waiting queue as follows:

$$Q = \lambda W = \frac{\lambda}{\mu(N-A)} P_C = \frac{A}{(N-A)} P_C. \quad (7)$$

By using of general definition of distribution function of any statistical distribution [4] and by applying its properties on the distribution function (5), we can derive the formula for calculation of the parameter GoS (Grade of Service) (percentage of calls, that are answered, or assigned to the agent before the defined threshold AWT – Acceptable Waiting Time) by known value of AWT [2, 6]:

$$GoS = 1 - P_C \cdot e^{-\mu(N-A)AWT} \quad (8)$$

The average number of requests in the system  $K$  (and also the average number of occupied lines) is [7]:

$$K = N\eta + \frac{\eta}{1-\eta} P_C = A + \frac{A}{(N-A)} P_C = A + Q, \quad (9)$$

where we get from Little theorem a value of the average time  $T$ , that the requirement spend in the system:

$$T = \frac{K}{\lambda} = \frac{A+Q}{\lambda} = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{P_C}{\mu(N-A)}. \quad (10)$$

Other important parameter is the average utilization of agents  $\eta$  [2]:

$$\eta = 1 - \frac{\sum_{k=0}^{N-1} \left[ \frac{N-k}{n \cdot k!} A^k \right]}{\sum_{i=0}^{N-1} \frac{A^i}{i!} + \frac{A^N N}{N!(N-A)}}. \quad (11)$$

Erlang C formula is the basis for the analysis of parameters and simulation of the call center. Its shortness is the assumption of the unlimited waiting queue. This is not a problem in terms of available storage capacity, and therefore the waiting queue could be potentially unlimited, but no real caller will wait too long. Therefore, the limitation of waiting period represents other parameter that is under consideration in the special Markov models.

## 1.2 M/M/m/M model with limited length of waiting queue

A limited number of requirements placed in the queue of queuing system can be described by Markov model M/M/m/M, where the maximum number of requests in the system  $M$  is greater or equal to the number of servers  $m=N$  [2].

The probability  $p_0$  [2], that in the system occurs exactly 0 requirements (i.e. empty) is defined as:

$$p_0 = \left\{ \sum_{k=0}^{N-1} \frac{A^k}{k!} + \frac{A^N \left[ 1 - \left( \frac{A}{N} \right)^{M-N+1} \right]}{N! \left( 1 - \frac{A}{N} \right)} \right\}^{-1} \quad (12)$$

and then we can define the probability of the call rejection  $P_B$  [2] as:

$$P_B = \frac{A^M}{N^{M-N} N!} p_0 \quad (13)$$

and the probability, that the call will be assigned into the waiting queue  $P_C$  [2] as:

$$P_C = \frac{A^N}{N!} p_0. \quad (14)$$

Furthermore, for the values  $Q$  [2] and  $K$  [2] can be calculated as

$$Q = \frac{p_0 A^{N+1}}{(N-1)!(N-A)^2} \left\{ 1 - \left( \frac{A}{N} \right)^{M-N+1} - (M-N+1) \cdot \left( \frac{A}{N} \right)^{M-N} \left( 1 - \frac{A}{N} \right) \right\} \quad (15)$$

$$K = Q + A \left( 1 - \frac{A^M}{N^{M-N} N!} p_0 \right). \quad (16)$$

The time characteristics can be calculated on the basis of Little theorem [7].

These relations are relatively complicated. It is therefore possible a consideration, if the approximation using standard Erlang C formula is not sufficient, or what are the conditions, that such approximation is sufficiently exact. If the average number of requests in the system  $K$  will be far less than the limit  $M$ , it is possible to apply Erlang C formula without thinking of the maximum capacity. The more closely will be the value  $K$  to the limit, the less accurate results Erlang C formula will provide.

## 2. The principle of realized simulations

The basic of the simulated algorithm consists of three blocks:

- set up of inputs,
- traffic simulations,
- processing of measured values and their presentation.

### 2.1 Modeling of inputs

The basic parameters are:

- average number of incoming calls into the call center per time  $\lambda$ ,
- average time of call processing by agent  $1/\mu$ ,
- number of agents  $N$ .

The group of time parameters consists of:

- the total length of simulated traffic in seconds  $T_{SIM}$ ,
- time step of simulation  $T_{STEP}$  (by default 1 second),
- time to steady state  $T_{NAB}$ .

Vector with arrival times of each call based on the average number of requests per time unit and length of simulation is created by random generator. There is used a property about exponential distribution of length of the intervals between arrivals. Generated random variables are then tested by the statistical chi-quadrade [4]. There is test of vector consistency with exponential distribution on significance level  $\alpha = 0,05$ . In the case of unsuccessful test, the entire vector is randomly generated once again. Number of generated calls is by 20 % higher than the average number of calls that should enter into the call center through the simulated period.

Furthermore, vector with service times of individual calls that will be allocated during the simulation, is generated for each agent based on his average service time.

All these operations are performed in advance by reason of high rate and efficiency of MATLAB by working with vectors and matrices. The generation of set of values

is then faster than a gradual generation of individual values during the simulation.

### 2.2 Traffic simulation

The core of simulation is realized as a cycle, in which the each iteration represents one time step in simulation. In the each iteration is the vector of incoming calls compared with actual time and if it is necessary, the call enters into the simulated call center in proper timer. Concurrently the time of incoming call is recorded for later calculations. Call is placed into the waiting queue or directly assigned to a free agent. The time index is stored for later calculations (average waiting time) at the moment of call assignment.

Furthermore, in the each iteration the status and the occupancy of all agents are checked. If any of the agents is free, it will be assigned the first call from the waiting queue. If more agents are free, the call is assigned to agent that didn't work for the longest time. If there is not a call in the waiting queue, the agent remains as free and expects the arrival of another request into the system.

During the one iteration there is possibility of entry into the system and also assignment to the agent of more than one call simultaneously. The number of calls in the system is stored in each step so then it is possible to determine the number of calls in the waiting queue.

The mathematical model of queuing system is defined for the steady state. It means, that in the moment of parameters monitoring, the simulation runs infinitely long time.

The run of the simulation is terminated after predefined simulation time. Only the calls that are terminated (and are thus served by agent) before expiration of the simulated time are included into the statistics.

### 2.3 Processing of the simulation results

This phase of the algorithm ensures the processing of all measured data during the simulation (time of events generation, number of calls in the system, agent occupancy, ...). Based on these the particular parameters monitored in the call center are calculated. Thus obtained simulation results can be then compared with the expected values obtained by calculations through the mathematical model.

The general variables available from the results of the simulated model are:

- real length of the simulation (calculation time),
- number of simulated calls,
- average service time per one call ( $1/\mu$ ),
- average number of calls in the system ( $K$ ),
- average time spent by user in the system ( $T$ ),

- number calls placed in the waiting queue,
- probability of blocking  $P_B$ , respectively probability of waiting  $P_C$ ,
- average utilization of agents  $\eta$ .

### 3. Calculations using the Erlang C model

Erlang C model works in the basic form with 3 input parameters ( $A, N, P_C$ ). By use of relation (2) it is possible to divide the value of load  $A$  (generated by incoming calls) into 2 components:  $\lambda$  and  $1/\mu$ . Calculator thus always works with 4 values, while 3 of them act as input parameters and the last parameter is the output parameter. By adding of the waiting queue we can obtain a set of new parameters that can be calculated and then compared with the simulation results:

- average number of requests in the waiting queue  $Q$  (using (7)),
- average number of requests in the call center  $K$  (using (9)),
- average waiting time in the waiting queue  $W$  (using (6)),
- average time spent in the call center  $T$  (using (10)),
- value of GoS for AWT=20 s (using (8)),
- average utilization of agents  $\eta$  (using (11)).

Probability of insertion into waiting queue  $P_C$  can be entered in three different variants:

- direct entry of value  $P_C$ ,
- entry of average waiting time  $W$ ,
- entry of GoS.

When using a  $P_C$  as input variable, this variable is considered to the upper limit.

#### 3.1 Calculation of the parameter $P_C$

The calculation of the unknown value of probability of waiting can be easily realized by basic relation for Erlang C model (1). The dividing of two large number could lead to numerical errors and the obtained result might not be accurate. Therefore we derived an alternative relation, that is identical to the original (1) (in terms of result):

$$P_C(N, A) = \frac{1}{1 + \frac{N-A}{N} \left( \sum_{i=1}^N \prod_{k=0}^{i-1} \frac{N-k}{A} \right)} \quad (17)$$

Similarly, we can use Horner scheme [8] also for more efficient calculations.

#### 3.2 Calculation of the parameter $N$

Analytical calculation according to (1) respectively (17) would be very difficult. The solution is therefore through a numerical method. The easiest technique is to gradually increase of the number of agents and continuous checking the stop conditions. This depends on the form, in which the input value  $P_C$  is inserted (thus direct, or as  $W$ , or as  $GoS$ ). At the moment, where the current value in the calculation is less or equal to the required value, the necessary number of agents is found. The implementation uses this idea, but applying of the relation (17) in every step of the calculation it is possible to use the current result obtained from the previous iteration cycle for the lower value of  $N$ . So, this is very quickly method to find the unknown value  $N$ .

#### 3.3 Calculation of the parameter $\lambda$ or $1/\mu$

The calculation of one of these unknown quantities in terms of Erlang C model means the find out of the load  $A$ , that can the system process at the specified parameters  $N$  and  $P_C$  (respectively  $W$  and  $GoS$  parameters). Consequently, we can calculate the second one by applying the formula (2) and one of the value  $\lambda$  and  $1/\mu$ .

In terms of implementation, the finding of the unknown value  $A$  is the most difficult of all three combinations. As the solution we can use the feature, that allows to search the value of unknown  $x$ , for which it holds  $f(x) = 0$ . In this case, the function  $f(x)$  for input  $P_C$  is:

$$f(x) = P_C(x, N) - P_{C\_INPUT} = 0 \quad (18)$$

If the input value is defined as the average waiting time  $W$ , respectively  $GoS$ , we can use the following substitutions by (6) and (10):

$$P_C = W\mu(N - A), \quad (19)$$

$$P_C = \frac{1 - GoS}{e^{-\mu(N-A)AWT}} \quad (20)$$

In both cases we have the value  $\mu$ . Therefore, if we need the solution for the unknown value  $1/\mu$ , we must find the solution for the value  $\mu$  and only then to calculate the load  $A$  by (2). The functions  $f(x)$  for input  $W$  (by use of substitution (19)) are:

$$f(x) = P_C(x, N) - W \cdot \mu \cdot (N - x) = 0, \quad (21)$$

respectively, if  $\mu$  is unknown, then:

$$f(x) = P_C\left(\frac{\lambda}{3600 \cdot x}, N\right) - \frac{W}{x} \cdot \left(N - \frac{\lambda}{3600 \cdot x}\right) = 0 \quad (22)$$

If the input parameter is  $GoS$ , the substitution (20) is used and the function for parameter  $\lambda$  is:

$$f(x) = P_C(x, N) - \frac{1 - GoS}{e^{-\mu(N-x)AWT}} = 0, \quad (23)$$

respectively, if  $\mu$  is unknown, then:

$$f(x) = P_C\left(\frac{\lambda}{3600 \cdot x}, N\right) - \frac{1 - GoS}{e^{-x\left(N - \frac{\lambda}{3600 \cdot x}\right) \cdot AWT}} = 0 \quad (24)$$

### 4. Simulations

The whole process of simulation for Erlang C model is shown in the following fig. 1:

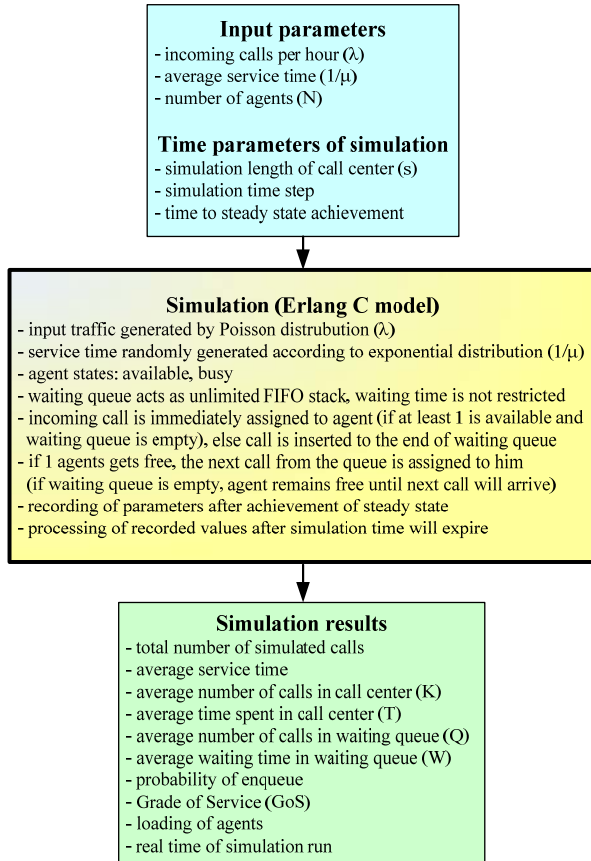


Fig. 1. Process of simulation for Erlang C model

#### 4.1 Simulation results

The parameters of simulated model are:

- 667 incoming calls per 1 hour,
- average call service time is 150 seconds,
- simulation time of the call center work is 30 hours with 1 second step (run to steady state is 1 hour).

The calls that cannot be processed immediately are placed into the waiting queue (according to Erlang C formula). For the system stability it is necessary to satisfy the condition  $A < N$ . Therefore the calculations and simulations are realized for the number of agents in the range of 28 to 37. In this range are the most notable changes observed in output parameters. The table 1 and table 2 shows the results obtained by calculation using the

mathematical model and results also obtained by simulations.

Tab. 1. The results of calculations

N	Pc [%]	K	T [s]	Q	W [s]	GoS [%]	η [%]
28	95,4	155	836,6	127,2	686,6	7,2	99,3
29	75,3	45,1	243,5	17,3	93,5	35,9	95,8
30	58,7	35,2	189,9	7,4	39,9	56,3	92,6
31	45,1	31,7	171,1	3,9	21,1	70,6	89,7
32	34,1	30	162,1	2,2	12,1	80,6	86,8
33	25,3	29,1	157,3	1,4	7,3	87,3	84,2
34	18,5	28,6	154,5	0,8	4,5	91,9	81,7
35	13,3	28,3	152,8	0,5	2,8	94,9	79,4
36	9,4	28,1	151,7	0,3	1,7	96,8	77,2
37	6,5	28	151,1	0,2	1,1	98,1	75,1

Tab. 2. The results of Erlang C model simulations

N	Pc [%]	K	T [s]	Q	W [s]	GoS [%]	η [%]
28	91,9	88,5	478	60,9	328	12,3	98,8
29	74	43,2	233	15,4	83,4	38	95,6
30	59,3	35,1	189	7,2	38,8	55,9	93
31	46,2	32,4	174	4,5	24,1	69,3	90
32	33,7	30	162	2,2	11,9	81,3	86,9
33	25,1	29,2	158	1,4	7,3	87,5	84,4
34	19,2	28,8	155	0,9	4,7	91,5	82,1
35	13	28,3	153	0,5	2,5	95,5	79,4
36	8,7	28	151	0,3	1,5	97,4	77,1
37	6,4	28,1	151	0,2	1	98,2	75,3

Obtained simulation results and calculations are very similar. The differences exist only in the case of the minimum number of agents. However, it is probably that any company will not carry on the call center with extremely poor quality of service [9, 10, 11, 12] delivering (mainly the very long waiting time).

From the caller point of view, the most important parameters are the average waiting time in the waiting queue  $W$  and parameter  $GoS$  (this case is evaluated for  $AWT=20$  seconds). The caller expects the lowest value of  $W$  and also the value  $GoS$ , which is close to 100%. From the call center point of view, the most important parameters are the number of agents  $N$  and their utilization  $\eta$ , because these two variables significantly affect the financial demands of service. The aim of the operator is to minimize the number of agents and to maximize their utilization. The aim of the analysis is therefore to find such a minimum number of agents  $N$ , when the operation parameters are yet on the sufficient level. As suitable we can consider the  $GoS$  parameter on level 80% and the average waiting time about 10seconds.

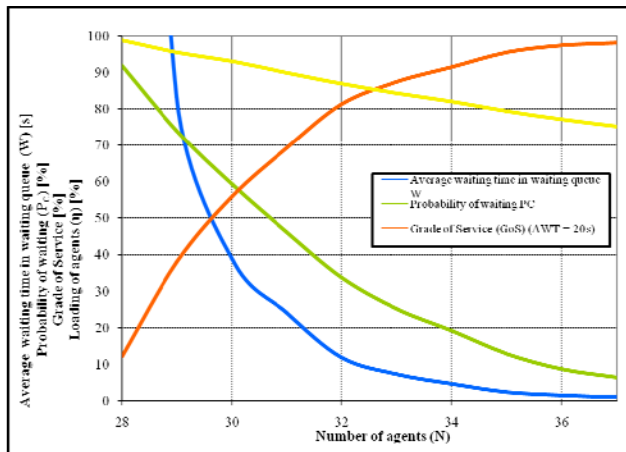


Fig. 2. Simulation results

The fig. 2 shows the characteristic curve of the important call center simulation results according to Erlang C model assumptions in relation with the number of agents  $N$ . The characteristic curve of the average waiting time has a very strong exponential character and thus a slight increase of the number of agent (about 1 to 2 agents) can bring a significant improvement of this parameter. A similar, even though less aggressive, is the characteristic curve of the probability  $P_C$ .  $GoS$  parameter also exponentially converges to the level 100% and we can see that a small change in the number of agents can bring significant improvement. The characteristic curve of agents load  $\eta$  is in the displayed range almost linear.

According to the above mentioned requirements on the provided quality of service by the call center is in this case possible to consider 32 agents as sufficient. By adding two agents it is possible to shorten the average waiting time by half and to increase the value of  $GoS$  parameter at 10% on very decent level (90%). The utilization of agents does not decrease below 80% and therefore it does not create unnecessarily long pauses, when the agents were redundant.

## 5. Conclusion

Based on calculations and simulations it can be stated, that in term of simplicity and accuracy of obtained results Erlang C formula is applicable for call center simulations. However, its shortness is the possibility of calculations only for one service group, and also the need to define for all agents the same service time.

Despite these limitations, it is possible to use the basic Erlang C formula also for calculations for call center with several service groups. In this case when it is possible to determine the probability that calls are routed to the individual service groups, then it is possible to use the basic Erlang C formula for calculations for each service group individually. The number of incoming calls per time unit is the aliquot portion of the total number of incoming calls into the call center. The Erlang C formula calculation

can be used in the case of different performance of agents. It is possible to determine the value of the average call processing time or average number of the processed calls per time unit by dividing the total number of processed calls per time unit of all agents and the total number of agents in service group.

For the purpose of further study it would be interesting to expand the simulations by more independent service groups at the same time and the random distribution of call between them according to defined probabilities. Another interesting possibility could be different average service time for individual agents. Finally there is possibility to simulate the impact of unequal performance of agents on the results of the whole call center.

## Acknowledgments

This work is a part of research activities conducted at Slovak University of Technology Bratislava, Faculty of Electrical Engineering and Information Technology, Department of Telecommunications, within the scope of the projects VEGA No. 1/0565/09 „Modeling of traffic parameters in NGN telecommunication networks and services“ and ITMS 26240120029 “Support for Building of Centre of Excellence for SMART technologies, systems and services II”.

## References

- [1] BERGEVIN, R., WYATT, A. *Contact centers for dummies*. [online]. Indianapolis (IN) (USA): Wiley Publishing, 2005, 80 p. ISBN 0-471-75819-1. Available from: <<http://www.algotech.sk/files/ContactCentersforDummies.pdf>>.
- [2] UNČOVSKÝ, L. *Stochastické modely operačnej analýzy*. 1<sup>st</sup> ed. Bratislava: ALFA, 1980, 416 p. ISBN 63-557-80.
- [3] POLEC, J., KARLUBÍKOVÁ, T. *Stochastické modely v telekomunikáciách*. 1<sup>st</sup> ed. Bratislava: Fond Jozefa Murgaša pre telekomunikácie n.f. vo vydavateľstve FABER, 1999, 128 p. ISBN 80-968125-0-5.
- [4] RIEČANOVÁ, Z. et al. *Numerical methods and mathematical statistics*. 1<sup>st</sup> ed. Bratislava: ALFA, august 1987. 496 p. ISBN 063-559-87.
- [5] Diagnostic Strategies. *Traffic Modeling and Resource Allocation in Call Centers*. Needham, USA: Diagnostic Strategies, 2003. Available from: <[http://www.fer.hr/\\_download/repository/A4\\_1Traffic\\_Modeling.pdf](http://www.fer.hr/_download/repository/A4_1Traffic_Modeling.pdf)>.
- [6] KOOLE, G. *Call Center Mathematics: A scientific method for understanding and improving contact centers*. Amsterdam: Vrije Universiteit, 68 p. [cited 2008-01-26]. Available from: <<http://www.math.vu.nl/~koole/ccmath>>.
- [7] BOLCH G., et al. *Queueing Networks and Markov Chains*. 2<sup>nd</sup> ed. Hoboken, New Jersey, USA: John Wiley, 2006, 878 p. ISBN 0-471-56525-3.
- [8] FORSYTHE, G. E., MALCOLM, M. A., MOLER, C.B. *Computer Methods for Mathematical Computations*. Upper Saddle River, New Jersey, USA: Prentice Hall, 1977, 259 p. ISBN 0131653326.
- [9] POSOLDOVÁ, A., BAROŇÁK, I. *AC Methods in ATM and IP Networks*. In New Information and Multimedia Technologies. NIMT

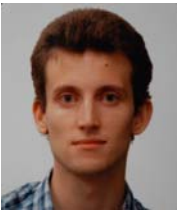
2010: Brno, Czech Republic, 16. - 17. 9. 2010, Brno University of Technology, pp. 20-23, ISBN 978-80-214-4126-2.

- [10] BALOGH, T., LUKNÁROVÁ, D., MEDVECKÝ, M.: *Performance of Round Robin-Based Queue Scheduling Algorithms*. In: CTRQ 2010: The Third International Conference on Communication Theory, Reliability and Quality of Service, Athens, Greece, 13.-19. 6. 2010, IEEE Computer Society, 2010, pp. 156 - 161, ISBN 978-0-7695-4070-2.
- [11] BAROŇÁK, I., MIČUCH, J.: *Preventive Methods Supporting QoS in IP*. In: New Information and Multimedia Technologies. NIMT 2009, Brno, Czech Republic, 17. - 18. 9. 2009, Brno University of Technology, pp. 15-23, ISBN 9778-80-214-3930-6.
- [12] VOZNAK, M., ROZHON, J.: *Methodology for SIP Infrastructure Performance Testing*, WSEAS TRANSACTIONS on COMPUTERS, Issue 11, Volume 9, pp. 1012-1021, September 2010, ISSN 1109-2750.

## About Authors ...



**Erik CHROMÝ** was born in Veľký Krtíš, Slovakia, in 1981. He received the Master degree in telecommunications in 2005 from Faculty of Electrical Engineering and Information Technology of Slovak University of Technology (FEI STU) Bratislava. In 2007 he submitted PhD work from the field of Observation of statistical properties of input flow of traffic sources on virtual paths dimensioning and his scientific research is focused on optimizing of processes in convergent networks. Nowadays he works as assistant professor at the Department of Telecommunications of FEI STU Bratislava.



**Tibor MIŠUTH** is a student of PhD. study at Department of Telecommunications, Faculty of Electrical Engineering and Information Technology of Slovak University of Technology Bratislava. He focuses on application of Erlangs' equations both in classic telecommunication networks and modern IP networks.



**Matej KAVACKÝ** was born in Nitra, Slovakia, in 1979. He received the Master degree in telecommunications in 2004 from Faculty of Electrical Engineering and Information Technology of Slovak University of Technology (FEI STU) Bratislava. In 2006 he submitted PhD work "Quality of Service in Broadband Networks". Nowadays he works as assistant professor at the Department of Telecommunications of FEI STU Bratislava and his scientific research is focused on the field of quality of service and private telecommunication networks.