Computer Simulation as a Tool for Calculation of the Paging System Capacity

Miroslav Belošević
Research & Development Department of Paging Systems
Swissphone Telecom
Fälmisstrasse 21
CH-8833 Samstagern, Switzerland
E-mail: miroslav.belosevic@swissphone.ch

Mladen Kos
Department of Telecommunications
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3
HR-10000 Zagreb, Croatia
E-mail: mladen.kos@fer.hr

KEYWORDS
Paging system capacity, Pocsag, simulation, queuing system, random number generation, optimisation

ABSTRACT

To calculate the capacity of a paging system two different models can be applied: a theoretical and a simulation model. This paper presents factors that affect the capacity of a paging system, shows how the capacity can be calculated by use of mentioned models and compares obtained results. The simulation process is described and impact of different queuing algorithms on system capacity is presented. The paper discusses also the relationship between models and shows that the capacity of a paging system can be determined by use of a theoretical model only if a simulation model has been applied before.

INTRODUCTION

The capacity of a paging system is influenced by many different factors (Ball et al. 1985). These factors can be classified and divided on three different levels: physical, transport and presentation level. The factors on the physical level affect and reduce the time that is available for the transmission of paging messages. These factors are wide-area coverage techniques (time and/or frequency divided adjacent paging areas) and organisation of distribution network (point-to-point or point-to-multipoint) as well as network synchronisation and operation and maintenance policy. All of mentioned factors require some amount of time that is therefore not available for the transmission of a real paging traffic.

The second, transport level is the level of a paging protocol. This level will be in more detail described in this paper. Given the time disposable for the transfer of paging messages on the first level, a paging protocol on the second level affects the amount of bits available for transfer of messages.

Table 1: Effective speed of paging protocol

Protocol	Nominal [bit/s]	Effective [bit/s]	Ratio E./N.
Pocsag	512	286	0.56
Pocsag	1200	696	0.58
Pocsag	2400	1392	0.58
Ermes	6250	3500	0.56

Because of its nature (one-way radio protocol and quasi-synchronous operation), a paging protocol uses relatively small transfer speeds and a great amount of redundant comparing to the number of information bits. Table 1 shows nominal and effective speeds for the most common paging protocols. Other factors on this level such are a preamble, a synchronisation codeword, a packet size, battery saving techniques and a queuing algorithm diminish the number of the information bits too. Taking all of these factors into account we can say that an effective speed of a paging protocol is approximately a half of its nominal speed.

Given the number of available information bits on the transport level, different presentation methods on the third level determine the amount of information that can be transferred. For example, alphanumeric messages are usually encoded with standard ASCII block code assuming 7 bits per character. But if the paging operator agrees to use capital letters only, same message can be encoded with only 6 bits per character. In such a way 14% of the channel capacity can be saved without loosing of any information. Information can be also encoded using Huffman codes or some other methods that can reduce the amount of transferred bits needed to present same information. Those methods, as well as factors influencing the capacity of a paging system on the first level are detailed explained in (Belošević 1999). Discussion in this paper is concentrated on the second level and on theoretical and simulation model for calculation of the paging system capacity.

THEORETICAL MODEL

The capacity of a paging system can be expressed either as a total number of messages n_{msg} sent on a channel during the busy hour or a total number of subscribers n_{user} that a paging channel can serve. The latter can be obtained by dividing the total number of messages by the busy hour call rate *bhcr*:

$$n_{user} = \frac{n_{msg}}{bhcr} \qquad (1)$$

Having in mind that the busy hour call rate is an empirical value different from one to other paging system, the capacity of a paging system calculated with this method can be considered as an empirical value too. Because of that the term "maximal number of messages" will be used instead of "maximal number of users" to denote the capacity of a paging system.

The main characteristics of applied paging protocol and of carried traffic, as well as some transmission parameters have to be known to calculate the capacity of a paging system by use of a theoretical model. For the calculation of the paging system capacity the Pocsag protocol will be used further in this paper while almost 80% of the whole world subscriber base still use this protocol. Given the protocol speed, the preamble and batch length as well as transmission packet length and number of codewords per one Pocsag batch, the total number of information codewords per hour can be calculated. This value divided by an average number of codewords per message equals the total number of messages n_{msg} that can be sent on a Pocsag paging channel per hour.

$$n_{msg} = \frac{Int\left(\frac{v \cdot t_{packet} - l_{preambel}}{l_{batch}}\right) \cdot \frac{t_{hour}}{t_{packet}} \cdot m_{batch}}{\sum_i \left(Round\left(\frac{c_{msg} \cdot l_{symbol}}{l_{CW}}\right) + 1 \right)_i \cdot p_i} \qquad (2)$$

Following parameters are variables:

v	protocol speed (bit/s)
t_{packet}	transmission packet length in seconds
c_{msg}	number of characters per message
p_i	message type distribution

$$\sum_i p_i = 1; i \in \{ToneOnly, Numeric, Alphanumeric\}$$

Following parameters are constants:

t_{hour}	3600	number of seconds per hour
$l_{preamble}$	576	preamble length (bits)
l_{batch}	544	batch length (bits)
l_{CW}	20	codeword length (bits)
l_{symbol}	0,4,7	symbol length (bits) 0 - Tone Only 4 - Numeric 7 - Alphanumeric
m_{batch}	16	number of codewords in batch

This equation assumes that the transmission packet length t_{packet} is given in number of seconds rather than in number of batches what would be the other possibility. One Pocsag transmission packet consists of preamble followed by certain number of batches, which depends on a packet length. Each batch comprises one synchronisation and 16 information codewords grouped in 8 frames (figure 1). The expression in the denominator of the equation (2) denotes a required number of codewords per message taking into account all three message types supported by the Pocsag protocol. The values p_i give the probability of appearance of particular message type. The length of a message is given by a parameter c_{msg}. It should be 0 (zero) for tone only messages. "+1" stands for an address codeword because each message starts with an address codeword followed by a number of message codewords in case of numeric and alphanumeric messages. Tone only messages consists only of an address codeword. Equation (2) will be used later in this paper to compare the calculated capacity with the capacity obtained by use of a simulation model.

Although is this equation mathematically absolutely correct, it wouldn't give correct results in practice. Because of battery saving techniques the beginning of a message for particular subscriber has always to be placed in the same frame as dictated by the last three bits (modulo 8) of the pager address. The effect of this rule is much longer battery life (RCSG 1986) but also a reduced efficiency with which a message can be placed into a packet. That means that some of available information codewords will not be filled with a message data. There will be gaps between messages. These gaps are filled with idle codewords, which carry no information.

The ratio of information (address and message) codewords vs. total number of codewords is known as batch utilisation factor. This value has considerable impact on the capacity of a paging system, but cannot be calculated with any mathematical method. The figure 1 shows the 5-batch Pocsag packet without preamble. The packet is filled with four messages.

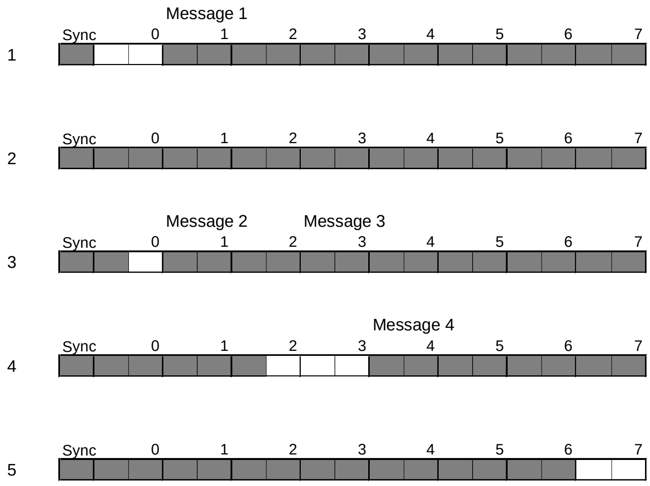

Figure 1: Pocsag packet

Actually there is enough free space for additional eight tone only messages. But in the moment of packet creation there were no such messages in the queue, which would have replaced the idle codewords with message codewords. The result is a batch utilisation factor of bu = 0.9. Now, it can be seen that the equation (2) is valid only in idealistic case, assuming that all of available codewords are filled with message data. Because of that this equation have to be modified and multiplied by a batch utilisation factor bu to get a useful equation for the calculation of the paging system capacity.

$$n_{msg}' = n_{msg} \cdot bu \qquad (3)$$

But the problem is that a batch utilisation factor bu cannot be calculated by any theoretical model. As well as busy hour call rate, it can be determined only by a simulation model or by an observation over a period of time in a real paging system.

SIMULATION MODEL

The second drawback of a theoretical model is that a time dimension of a paging system (message delay distribution) cannot be expressed with this model. The message delay denotes the time between arrival of a message in a queuing system and start of its Pocsag transmission. The message delay distribution is one of the main factors that determine the quality of service of a paging system and therefore of great importance for its operator. Like a batch utilisation factor, it can be calculated only by a simulation.

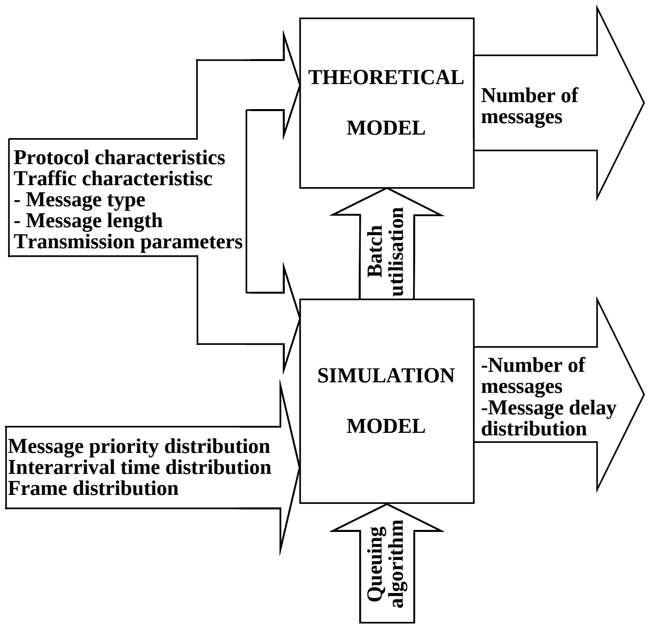

Figure 2: Relationship between models

Figure 2 shows inputs and outputs of a theoretical and a simulation model, as well as a relationship between them. A simulation model has three input parameters more: a message priority, an inter-arrival time and a message starting frame distribution.

There are three levels of a message priority: normal, priority and urgent. Only a small part of a total message number (5-15%) may have the priority higher then normal, to guarantee the fast distribution of priority messages even if the queuing system is overloaded. The message inter-arrival time is distributed according to exponential distribution and the message frames are (in a normal case) distributed uniformly over the eight frames within a Pocsag batch. Additional parameter that has impact on results of a simulation model is organisation of a queuing system and algorithm used for building of a transmission packets. Above-mentioned batch utilisation factor is shown as an output result of a simulation and at the same time as an input parameter of a theoretical model.

Taking all these parameters into account a computer simulation program has been developed to simulate traffic flows in a real Pocsag paging system and to make possible the calculation of its capacity. The structure of the simulation program is depicted on figure 3.

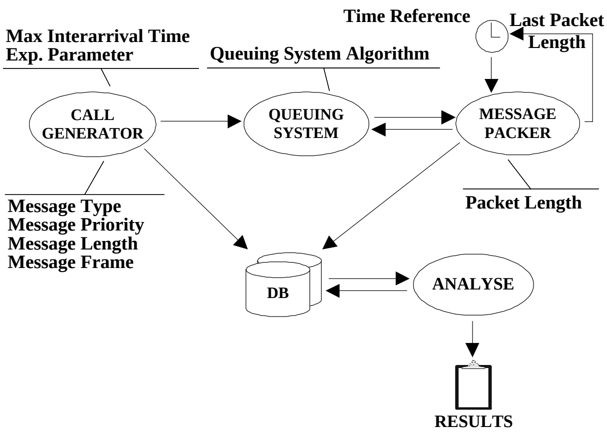

Figure 3: The simulation program structure

It comprises same modules as a real paging system plus call generator, which is specific for the simulation process. In the following text each module of the simulation program is explained in more detail.

Call Generator

The main task of the call generator is to simulate an input traffic of a real paging system which is considered to be a poisson process. It generates random calls according to SDL diagram depicted on figure 4.

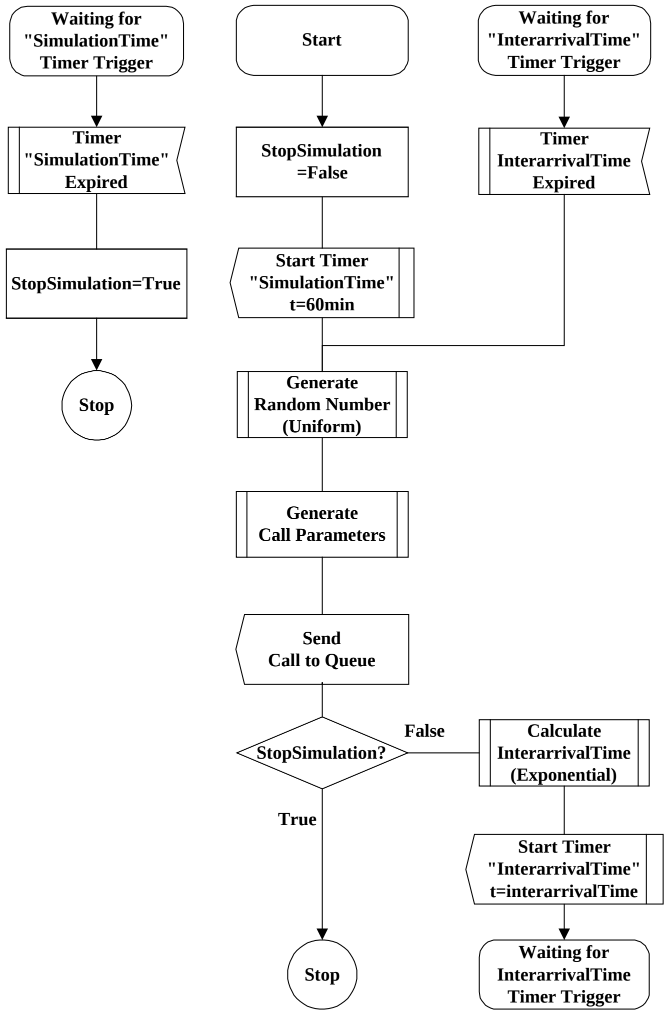

Figure 4: Call generator policy

The call generator policy is explained step-by-step in following text:

1. After the program start the "StopSimulation" variable is set to False.
2. Timer "SimulationTime" is started. This timer will expire after 60 minutes and will set the "StopSimulation" variable to True. This will cause the end of the simulation process.
3. Uniformly distributed random number is generated.
4. According to this number and parameters of the simulation following message attributes are calculated:
   - message type (tone only, numeric, alphanumeric)
   - message priority (normal, priority, urgent)
   - message length
   - message frame (0…7)
5. Call is sent to queue.

6. If the "StopSimulation" variable is set to True the simulation process stops. Otherwise follows next step.
7. According to the number from the third step, an exponentially distributed random number is calculated. This number will denote the message inter-arrival time.
8. A timer is started taking inter-arrival time calculated in previous step as timeout value.
9. After timer expiration the whole cycle is repeated starting from step 3.

These steps are repeated until the end of the simulation. The simulation will be stopped after an hour, because all values, which express the capacity of a paging system are normalised to the time period of one hour.

For the calculation of the paging system capacity is necessary that a paging system works close to his maximum capacity. That means that output packet has to be filled up as much as possible. This can be done only if there are enough messages waiting in the queue. On the other hand, if there is a lot of messages in the queue, the average message delay and the quality of service may start becoming unacceptable. That means that the call generator parameters have to be carefully selected to enable the operation of a paging system at very close to its maximum capacity with very little delay. Given the message type distribution and maximum message length, an ideal message inter-arrival time distribution has to be found, which will force a paging system to work close to its maximum.

Queuing System

Generated calls are sent directly into a queuing system. Two different organisations of a queuing system will be evaluated in this paper:

1. The first organisation assumes eight groups of queues, one for each frame. Each group consists of three queues, one for each priority level (Urgent, Priority, Normal), so there are total 24 FIFO queues (figure 5). Message is placed in the particular queue according to its starting frame and priority level.

2. The second organisation assumes only three FIFO queues, one for each priority level (figure 6). Message is placed in the particular queue according to its priority level.

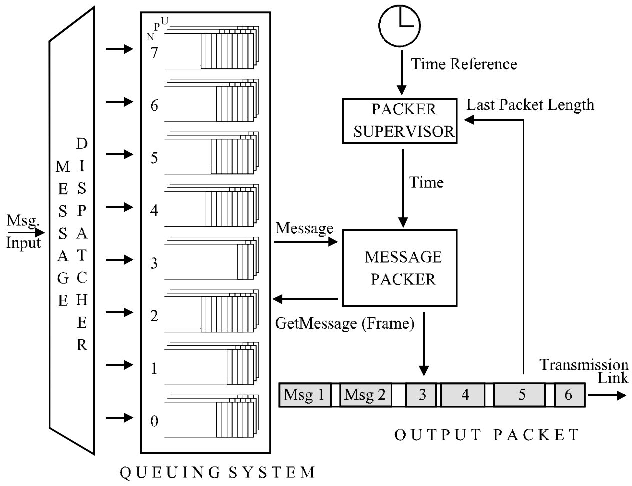

Figure 5: Organisation of queuing system Nr. 1

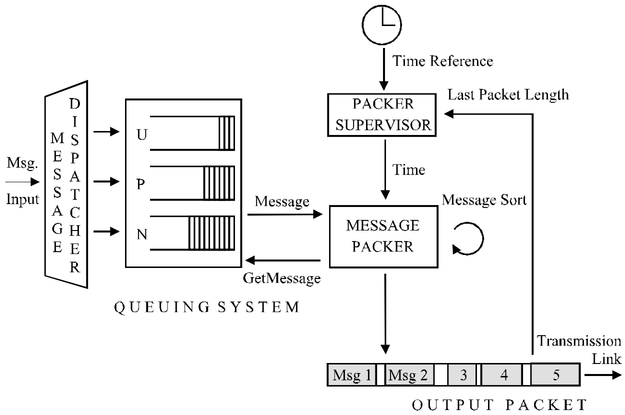

Figure 6: Organisation of queuing system Nr. 2

Messages are waiting in the queue for the message packer process, which will take them out according to appropriate algorithm and place them in the output packet.

Message Packer

Message packer waits for an external signal from the time reference, which denotes the start of a packing procedure. This signal is generated every t_{packet} seconds. Depending on organisation of a queuing system, message packer applies corresponding algorithm to take the messages out of the queue and to pack them into output packet:

Algorithm A1. This algorithm starts and looks for a message in the queue group for the frame 0. A call with a highest priority level will be packed. Let us assume that this call ends in frame 5. The program will look further for a call (with a highest priority level) in the queue group for the frame 6. If such a call exists it will be taken out and packed. Otherwise, message packer will look for a call in the queue group for the next frame and so on. The algorithm stops if the whole packet is filled up with messages or there is no more messages waiting in the queuing system.

Algorithm A2. This algorithm always looks for a message of highest priority waiting in the queuing system regardless of its starting frame. The message is placed on the first possible position (frame) in an output transmission packet. Let us assume the last message placed in the packet ends in frame 5. New message, which is still in the queue, starts in frame 4. So a new batch has to be appended to the packet and the new message will be placed in its frame 4. If there is not enough free space to place the message (packet length exceeds limit), a separate algorithm tries to reorganise the packet and to minimise the number of idle codewords. It changes the order of the messages and tries to optimise number of batches in packet and to create enough free space at the end of the packet for the new message. If the new message cannot be placed into the packet even after reorganisation, the next message is taken out from the queue. The algorithm ends if there is no more messages in the queue or if there is not enough free space in the output packet for any of the waiting messages.

Although the real efficiency of each algorithm can be tested only by use of a simulation process, some statements on its characteristics can be made even before the start of a simulation. It is obvious that the first algorithm (A1) is faster than the second one (A2). The algorithm A2 requires some extra time to reorganise the message order in the transmission packet. The advantage of the algorithm A1 is that the messages are automatically sorted during its arrival at the system according to their starting frame. But here we can observe a certain drawback of the algorithm A1. Namely, the messages are taken out from the queue according to their starting frame (on the first place) and than according to their priority level. In the example above for the algorithm A1, the no-priority message in frame 6 has precedence over all priority messages in other frame groups. It also has precedence over all other messages (from other frames) that have come in the queuing system before it. That is the reason why the algorithm A2 has to be investigated. The algorithm A2 assumes that only priority messages have precedence over other messages. Whether this drawback of algorithm A1 has impact on the message delay distribution and batch utilisation factor or not, can be tested only with a simulation program.

Database

All calls generated during the simulation process are logged in the database. First, the call generator logs all generated calls with their attributes and absolute time of generation. Second, the message packer scans whole output packet and writes the times of call transmission in the database record for that call. Once those data are available, the analysis of the simulation results can be started.

SIMULATION RESULTS

Following parameters have been chosen for the simulation program:

1. message type distribution:
   - 10% tone only
   - 20% numeric
   - 70% alphanumeric messages
2. maximum message length:
   - 10 numeric
   - 80 alphanumeric characters
3. message priority distribution
   - 85% normal
   - 10% priority
   - 5% urgent messages
4. maximal/minimal inter-arrival time of 2s/1ms
5. protocol speed 1200 bit/s
6. output packet length 30 seconds

Given these parameters, optimal parameters for the call generator have to be found. As mentioned before in this paper an optimal message inter-arrival time distribution has to force a paging system to work close to its maximal capacity. At the same time an acceptable quality of service has to be achieved. An inter-arrival time distribution *f(t)* depends on maximal inter-arrival time t_{max} and on a parameter of an exponential function *a*.

$$f(t) = t_{\max} \cdot e^{-at} \qquad (4)$$

t is uniformly distributed random number. Different values of parameter *a* have yield to results showed in table 2.

It is interesting to observe that a change of parameter *a* of an exponential function from 0.00175 to 0.003 generates 6.7% more calls, but also more then eight times longer delay. On the other side the batch utilisation factor is for 3.2% better because of much larger assortment of different messages, which are waiting in the queuing system.

Table 2: Influence of parameter *a* on the call generator performance

Parameter *a*	**0.001**	**0.00175**	**0.002**	**0.003**
total calls	3794	5605	5861	5980
bu [%]	93.1	96.7	99.4	99.9
max delay [s]	52	200	666	1693

The message delay distribution functions for no-priority messages and for the values from table 2 are shown on figure 7.

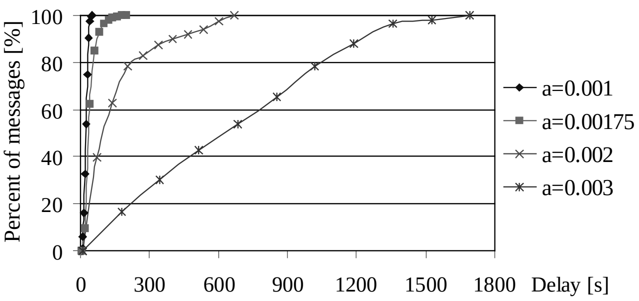

Figure 7: Message delay distribution, no priority, Algorithm A1

A value of a parameter *a* = 0.00175 has been chosen for the simulation program, because it gives the optimal trade-off between the number of served calls and the batch utilisation factor. It produces also the message delay distribution function that is still acceptable.

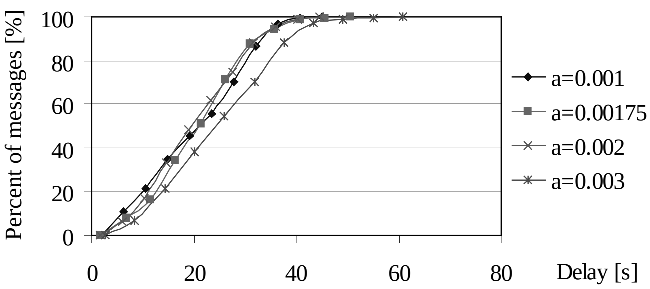

Figure 8: Message delay distribution, urgent priority, Algorithm A1

It is interesting to compare the end results for the same values shown in table 2, but for messages with urgent priority (figure 8). It can be seen that message delay distribution functions are in this case almost identical. Messages with priority can be practically served immediately after an arrival at the queuing system, regardless of the current traffic load and possible queue overload, because only relatively small part of all generated messages have the highest priority level.

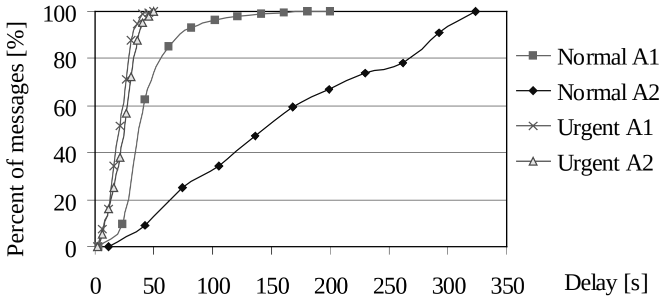

Figure 9: Delay distribution, Algorithm A1/A2, uniform frame distribution

The efficiency of both algorithms is compared on figure 9 for the same traffic conditions (*a* = 0.00175) and for the uniform frame distribution. The algorithm A1 has served total 5605 calls and has achieved the batch utilisation factor of 96.7% compared to algorithm A2, which has served only 5256 calls during the same time period with batch utilisation of 90.2%. The message delay distribution function is also much more acceptable for the algorithm A1 in spite of assumptions made in previous chapter.

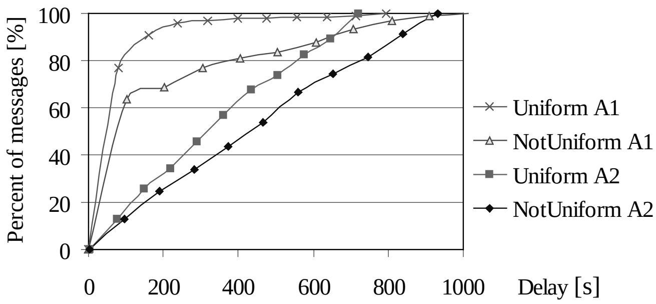

Figure 10: Delay distribution, Algorithm A1/A2, uniform/not-uniform frame distribution

Figure 10 shows an influence of a message frame distribution function on the algorithm efficiency. A not-uniform frame distribution has caused a general degrade of message delay distribution for both queuing algorithms. But also in that case the algorithm A1 has produced better results than other algorithm and so it has approved its superiority in comparison with algorithm A2.

THEORETICAL VS. SIMULATION MODEL

Once we have done the computer simulation and obtained the batch utilisation factor, the theoretical model can be applied. The parameters mentioned in previous chapter and batch utilisation factor *bu* = 96.7% have to be put in equations (2) and (3). The parameters for the message type distribution have to be slightly modified. Namely the input parameters of the simulation program were 10% tone only, 20% numeric and 70% alphanumeric. But the call generator has really generated 9,4% tone only, 20% numeric and 70,6% alphanumeric calls. The theoretical model will give 5701 calls compared to 5605 calls obtained by the simulation model. The difference of 1,7% is consequence of limited accuracy of a computer software timer used during the generation of random calls, but the difference is so small that it can be neglected.

FURTHER RESEARCH

The simulation program presented in this paper has been approved on the real paging system. During design of new Pocsag paging system, queuing algorithms presented in this paper have been compared. According to results shown in previous chapter the algorithm A1 has been chosen for new system. The simulation program turned out to be useful during introduction of a new broadcasting service in an existing paging system too. New service, means new characteristic of an input traffic. Only by use of a simulation program can be tested whether this new traffic characteristic would influence and degrade the existing quality of service of a paging system.

The research shown in this paper relates to paging systems with Pocsag protocol. Further research will be directed to evaluation of queuing algorithms and overall system performance of an Ermes paging system. For this purpose a lot of work already done for this research can be reused. Only a few modifications have to be made on call generator process and on database format. Queuing algorithms and message packer have to be adapted on Ermes protocol. Some of this work is already made and is described in (Belošević 1999).

CONCLUSION

Two different models for calculation of the paging system capacity have been evaluated in this paper. The theoretical model has been presented including the universal mathematical formula for calculation of the capacity of the Pocsag paging system. It has been shown that a theoretical model can give useful results only if a simulation model has been applied before. That is because of a batch utilisation factor, which cannot be obtained by a theoretical model, but only by use of a computer simulation. The structure of the simulation program has been presented, as well as influence of a queuing system policy on the simulation results. We have seen that the computer simulation can be used for the calculation of the paging system capacity as well as for the evaluation and comparison of different queuing algorithms. Besides that, the computer simulation has been approved as a unique tool for calculation of the message delay distribution function, which is also one of the main advantages in comparison with a theoretical model.

REFERENCES

Ball, D. M., Mabey, P. J.,"Traffic handling capacity of CCIR Radiopaging Code No. 1", *Land Mobile Radio*, Publication No. 65, pp 7-13, December 1985.

Belošević, M., "Optimisation of information flows and local access in mobile telecommunication networks" (Master science thesis), Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, 1999.

Hillier, F. S., Liebermann, G. J., *Operations Research - Einführung*, 5. Auflage, München, Wien, Oldenbourg, 1997.

Radiopaging Code Standards Group, *The book of the CCIR Radiopaging Code No. 1*, Secretary RCSG, British Telecom, 1986.

BIOGRAPHY

Miroslav Belošević was born in Glina, Croatia in 1966. He received the B.S. and M.S. degrees in electrical engineering from the University of Zagreb in 1989 and 1999, respectively. From 1990 to 1994 he was working in Ericsson Nikola Tesla, Zagreb, Croatia on a software design for a real-time systems dealing with operation and maintenance of a telecommunication network. Since 1994 he has been working in Swissphone Telecom, Switzerland, Research & Development Department of Paging Systems. His current research interests are related to optimisation of the capacity and information flows in a paging system and to evaluation of different queuing algorithms by use of a computer simulation.