

# Performance evaluation of the multiple output queueing switch with different buffer arrangements strategy

Grzegorz Danilewicz, Wojciech Kabaciński, Janusz Kleban, Damian Parniewicz,  
and Patryk Dąbrowski

**Abstract**— Performance evaluation of the multiple output queueing (MOQ) switch recently proposed by us is discussed in this paper. In the MOQ switch both the switch fabric and buffers can operate at the same speed as input and output ports. This solution does not need any speedup in the switch fabric as well as any matching algorithms between inputs and outputs. In this paper new performance measures for the proposed MOQ switch are evaluated. The simulation studies have been carried out for switches with different buffer arrangements strategy and of capacity  $2 \times 2$ ,  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$  and  $32 \times 32$ , and under selected traffic patterns. The simulation results are also compared with OQ switches of the same sizes.

**Keywords**— *packet switching, switching fabric, multiple output queueing.*

## 1. Introduction

The tremendous increase in the speed of data transport on optical fibers has caused a need of deploying next generation network nodes (switches/routers) with high-speed interfaces and large switching capacity. The challenges of building new network nodes include: implementing a large capacity switch fabric providing high-speed interconnection and devising a fast arbitration scheme resolving output contention problems. One of constraints that limits the switching capacity is the speed of memories used for buffering packets to resolve contention resolution in packet switches. Buffers can be placed on inputs, outputs, inputs and outputs, and/or within the switch fabric. Depending on the buffer placement respective switches are called input queued (IQ), output queued (OQ), combined input and output queued (CIOQ) and combined input and crosspoint queued (CICQ) [1].

In [2] we have proposed a new switch architecture which uses multiple output queueing (MOQ). In this architecture buffers are located at output ports and are divided into  $N$  separate queues. Each of  $N$  queues in one output port stores packets from one input port. We assume, that fixed-length switching technology is used, i.e., variable-length packets are segmented into fixed-length packets, called time slots or cells, at inputs and reassembled at the outputs. We will use terms cell and packet interchangeably further on. In the proposed architecture at most one packet is to be written to the one output queue in one time slot. There-

fore, the memory speed is equal to the line speed, but the performance of the switch is very similar to those of OQ switch. The proposed architecture is very promising and looks attractive for constructing high-capacity high-speed packet switches. In this paper new results of simulation studies carried out for the MOQ switch with different buffer arrangements strategy and selected traffic patterns are presented.

The rest of the paper is organized as follows. In Section 2 the general switch architecture proposed in [2] is reminded. In the next section the parameters of simulation researches are explained. Then performance measures for the proposed switch architecture with different buffer arrangements strategy under different traffic patterns are presented and compared with results obtained for OQ. Then we come to conclusions.

## 2. The switch architecture

The detailed description of MOQ switch architecture is given in [2]. In this switch output queues are located at output side of the switch. To reduce the memory speed an output buffer at each output port is divided into  $N$  separate queues. Each queue stores packets directed to the output only from one input. In this way this architecture is similar to the virtual output queueing (VOQ) switch, but multiple buffers are located at output ports not at input ports. The general architecture of the switch is shown in Fig. 1. The switch consists of  $N$  input ports,  $N$  output ports and the switch fabric. Input and output ports can be implemented on separated ingress and egress cards or they may be placed on one line card. At the output port the buffer memory is divided into  $N$  separate queues. Each queue stores packets directed from one input port. The output queue denoted by  $OQ_{j,i}$  at the output port  $j$  stores packets directed to this output port from input  $i$ . At one time slot each input port can send at most one packet and each output port can receive up to  $N$  packets, each from different input ports. Therefore, these packets can be simultaneously written to  $N$  queues.

The main advantage of these architecture is that it can operate at the same speed as input and output ports, and the lack of arbitration logic, which decides which packets from inputs will be transferred through the switch fabric to output ports (this arbitration mechanism is needed

in VOQ switches). However, since we have  $N$  queues in each output port, it is necessary to use an output arbiter, which chooses a packet to be sent to the output line. We propose to use round-robin scheme, which is widely used because of its fairness. The switch fabric in the proposed switch should have a capacity of  $N \times N^2$  and should be nonblocking at the packet level.

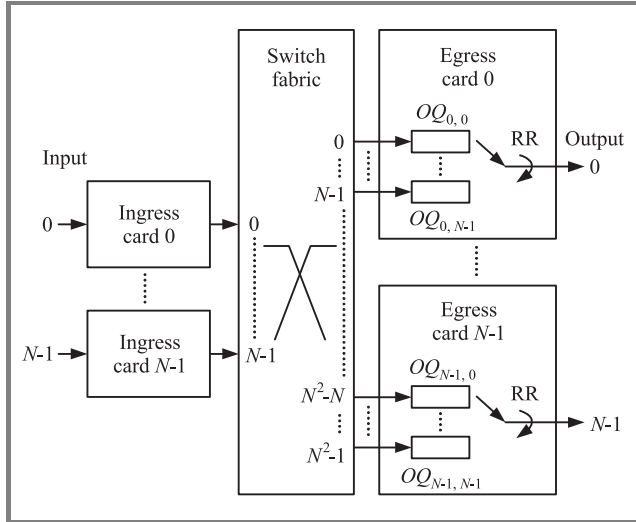


Fig. 1. The switch architecture with multiple output queuing.

In [2] we have shown that the MOQ switch has the hardware and wiring complexity similar to the VOQ switch, but for uniform traffic its performance is very similar to the OQ switch. In this paper more research results are given for uniformly and non-uniformly distributed traffics.

### 3. Buffer arrangements

Buffers in output ports are arranged into  $N$  separate queues. When  $N$  packets from  $N$  input ports are directed to one output port in the same time slot, each packet is written to the different queues. Therefore, the memory speed is the same as the line speed. Buffers may be arranged as separate queues with independent write pointers or as a memory bank with one pointer which points the same memory cells in each queue. The implementation of this two buffer arrangements strategies in hardware can influence the costs of MOQ switch.

Packets from  $N$  queues in each output port are read out using the round-robin algorithm. When independent write pointers are used, the round-robin pointer, denoted by RR (Fig. 1), is moved to the queue next to those read out in the previous time slot. When packets are written to the same position of the buffers (one write pointer is used), the operation of RR is modified in such a way, that when all packets from the same position are already read out, the RR is set back to 0. The operation of these two arrangements will be described by means of the following example.

In the first case the separate pointer is assign to each queue. This pointer, denoted by  $MP_{j,i}$ , points the end of queue  $OQ_{j,i}$ , where the next incoming packet to output  $j$  from input  $i$  will be written to. The example for output  $x$  is shown in Fig. 2. It is assumed that all queues are empty at the beginning of the first time slot. Pointers are shown by arrows which show the state of the pointers at the end of respective time slots. In the first time slot two packets (numbered 1 and 2) from inputs 0 and 1 arrive to the considered output  $x$ . The round-robin pointer (RR) is set to 0 (the head of line blocking (HOL) packet from  $OQ_{x,0}$  has the highest priority). Since buffer  $OQ_{x,0}$  is empty, the packet from input 0 is immediately directed to the output, the RR pointer is set to 1, and packet 2 is stored in  $OQ_{x,1}$ . The state of RR at the end of the time slot is shown in Fig. 2. The pointer of  $OQ_{x,1}$  is moved to the next memory cell. In the next time slot packets from inputs 0, 1 and 3 arrive (numbered 3, 4, and 5, respectively). They are stored in respective queues, while packet 2 from  $OQ_{x,1}$  is sent out. During the third time slot packets 6, 7 and 8 arrive from inputs 1, 2, and 3, respectively. Since RR is now set to 2 and buffer  $OQ_{x,2}$  is empty, packet 7 is sent directly to the output, while packets 6 and 8 are stored in  $OQ_{x,1}$  and  $OQ_{x,3}$ . In the next time slot packet 5 will be sent out from  $OQ_{x,3}$ . The sequence of packets from the same input port is preserve.

In the second case there is one pointer for all queues. This pointer, denoted by  $MP_j$ , points to the memory cells in all queues of output  $j$ , where the next incoming packets will be written to. The example is shown in Fig. 3. In the first time slot two packets (numbered 1 and 2) from inputs 0 and 1 arrive to the considered output  $x$ . The round-robin pointer (RR) is set to 0 (the HOL packet from  $OQ_{x,0}$  has the highest priority). Since buffer  $OQ_{x,0}$  is empty, the packet from input 0 is immediately directed to the output, packet 2 is stored in  $OQ_{x,1}$ , the  $MP_x$  is moved to the next memory cells in all queues (shown by arrows in Fig. 3), and the RR pointer is set to 1 (here also the state of RR is shown et the end of the time slot). In the next time slot packets from inputs 0, 1 and 3 arrive (numbered 3, 4, and 5, respectively). They are stored in the second memory cell of respective queues, while packet 2 from  $OQ_{x,1}$  is sent out. After this packet is read out, there is no any packet in the first memory cell in all queues. Therefore, the next cells in the queues are moved to the HOL position, and the RR is set to 0. During the third time slot packets 6, 7 and 8 arrive from inputs 1, 2, and 3, respectively. Since RR is now set to 0, packet 3 from  $OQ_{x,0}$  is sent to the output, while new packets are written to the buffer. In the next three time slots packets 4, 5, and 6 will be sent out from  $OQ_{x,1}$ ,  $OQ_{x,3}$ , and  $OQ_{x,1}$ , respectively.

In this second approach all packets which arrive to the given output are written in the same position of each buffer. So we can use only such positions where all memory cells are empty. When in the given time slot less than  $N$  packets arrive to the output, some memory cells will be empty and they could not be used to store packets until all packet in the same position of all buffers are read out. There-

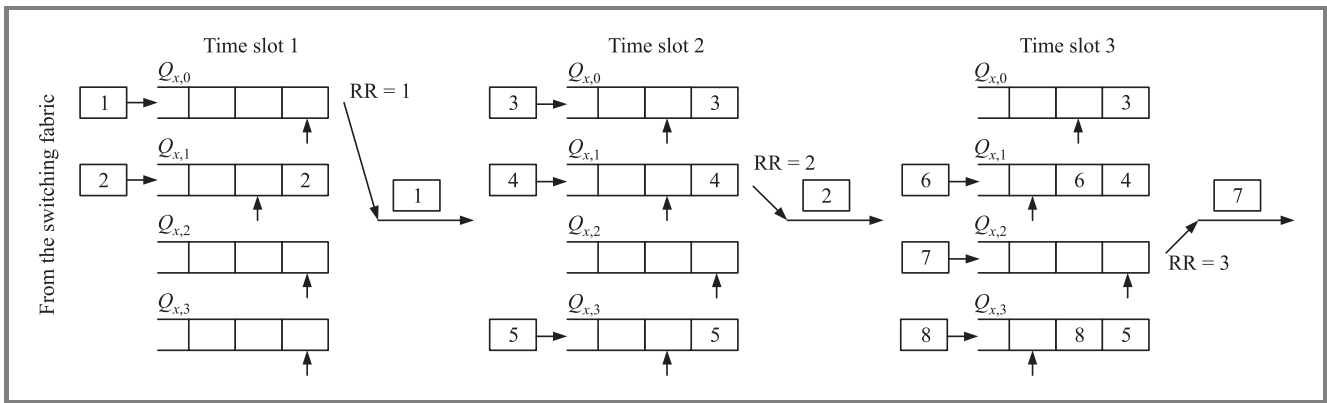


Fig. 2. The example of buffer operation with separate pointers.

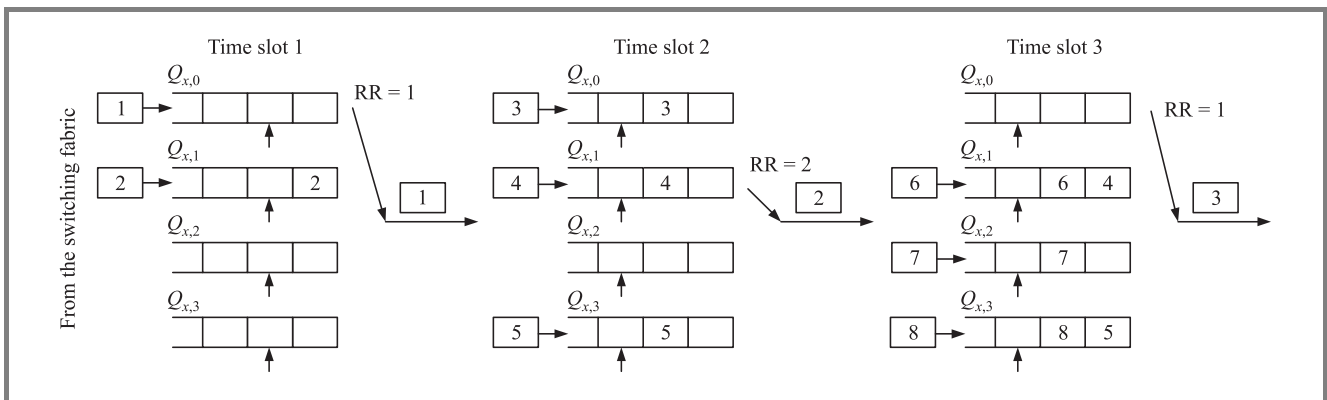


Fig. 3. The example of buffer operation with one pointer.

fore, the memory is not used as efficiently as in the first approach.

## 4. Simulation experiments

### 4.1. Packet arrival models

The Bernoulli arrival model is considered in the paper. In this arrival model cells arrive at each input in a slot-by-slot manner. Under Bernoulli arrival process, the probability that there is a cell arriving in each time slot is identical and is independent of any other slot. The probability that cell may arrive in a time slot is denoted by  $p$  and is referred to as the load of the input [3]. This kind of traffic defines a memoryless random arrival pattern.

### 4.2. Traffic distribution models

We consider several traffic distribution models which determines the probability that a cell which arrive in an input will be directed to the certain output. The considered traffic models are:

**Uniformly distributed traffic** – this type of traffic is the most commonly used traffic profile test in the literature [4–6]. In a uniformly distributed traffic probability  $p_{ij}$

that packet from input  $i$  will be directed to output  $j$  is uniformly distributed through all outputs, i.e.,

$$p_{ij} = p/N \quad \forall i, j. \tag{1}$$

**Non-uniformly distributed traffic** – in this traffic model some outputs have a higher probability of being selected, and respective probability  $p_{ij}$  was calculated according to the following equation [1]:

$$p_{ij} = \begin{cases} \frac{p}{2} & \text{for } i = j \\ \frac{p}{2(N-1)} & \text{for } i \neq j \end{cases} \tag{2}$$

**Diagonally distributed traffic** – in this model the traffic is concentrated in two diagonals of the traffic matrix, and the probability that a packet will be directed to any of the two outputs is equal to  $p/2$  [4–7]. This loading is skewed in the sense that input  $i$  has packets only for outputs  $i$  and  $|i + 1|$ , where  $|k|$  denotes the modulo  $N$  operation ( $|k| = k \bmod N$ ).

**Log-diagonally distributed traffic** – for a log-diagonally distributed traffic, the traffic matrix is defined by equation [4, 6]:

$$p_{ij} = 2p_{i|j+1|} \tag{3}$$

and

$$\sum_i p_{ij} = p. \tag{4}$$

For example, the load distribution at input 1 across outputs is

$$p_{1j} = 2^{N-j}p / (2^N - 1). \quad (5)$$

**Lin-diagonally distributed traffic** – this traffic is a further modification of diagonally distributed traffic. Lin-diagonally distributed traffic can be defined as

$$\bar{p}_d = p(N - d) / (N(N + 1) / 2) \quad (6)$$

with  $d = 0, \dots, N - 1$ , then  $p_{ij} = \bar{p}_d$  if  $j = |i + d|_N$ . This traffic model is an intermediate case between the uniformly and log-diagonally distributed traffics in which the load decreases linearly from one diagonal to the other [8].

### 5. Performance evaluation

In this section performance evaluation of the MOQ switch will be presented and compared with OQ switches. Simulation results indicate that mean time delay (MTD in time slots) and cell lose probability for MOQ switches with different buffer arrangements strategy are very similar and differences are unnoticeable. Thus only results for separate pointers are presented. The results have been obtained for switches of different sizes:  $2 \times 2$ ,  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$ . The results for switches of different sizes are very similar in shapes. The adopted buffer size assures — for each value of traffic load — stable values of MTD (the application of larger buffers do not lead to increase this

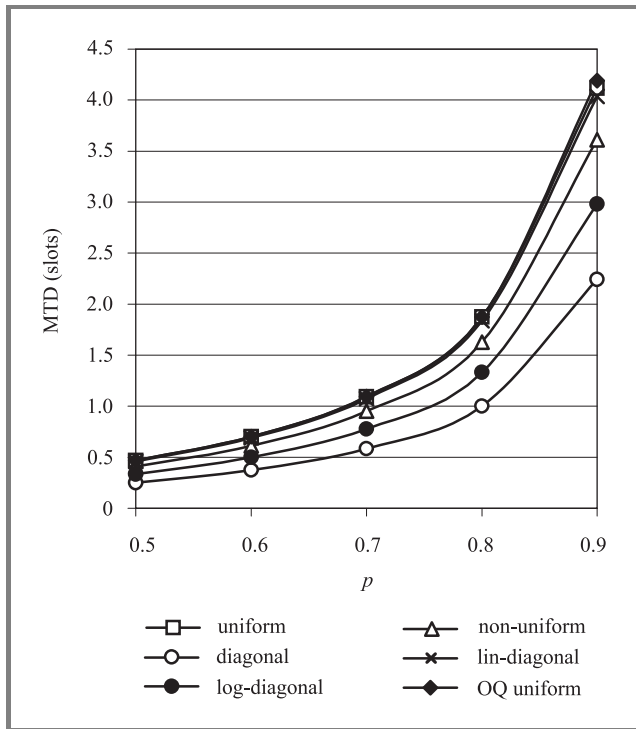


Fig. 4. The MTD for Bernoulli arrivals with different distributed traffic in a  $16 \times 16$  switch with MOQ ( $L = 16$ ) and OQ ( $L = \infty$ ).

waiting time). For OQ switch we have assumed that the buffer size is infinity. Results for OQ will be presented only for Bernoulli arrivals and uniformly distributed traffic and will be obtained from calculations based on formulas given in [3].

In Fig. 4 different traffic distribution models are compared in  $16 \times 16$  MOQ switch, where buffer size is equal to 16. The highest MTD in MOQ switch is observed for uniformly distributed traffic. This kind of traffic gives similar MTD values for MOQ and OQ switches but MOQ is slightly better.

The MTD in MOQ switches of different sizes versus  $p$  for uniform traffic and  $L = 16$  is plotted in Fig. 5. It can be seen that when the switch size is growing the MTD also grows but very slowly. This delay is almost the same as the theoretical MTD calculated for OQ switches of the same capacities.

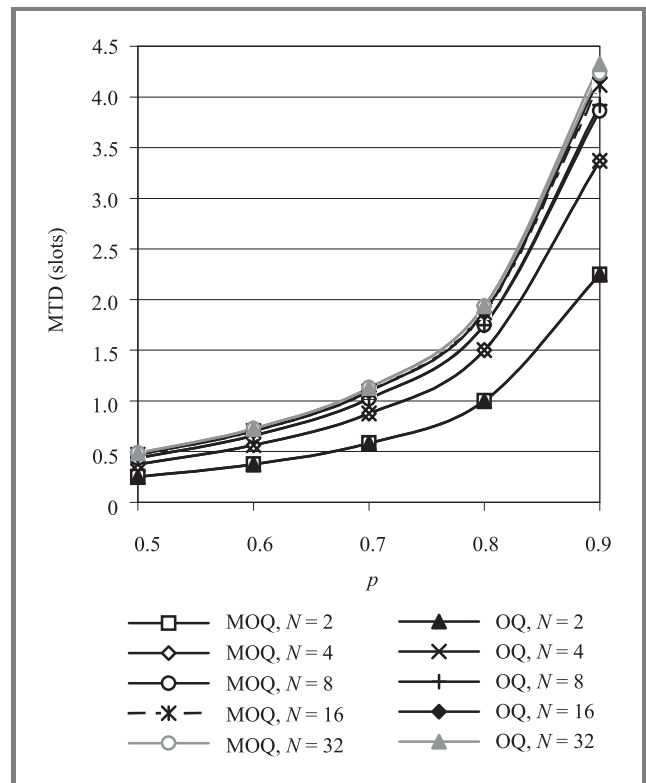


Fig. 5. The MTD for Bernoulli arrivals with uniformly distributed traffic and for different capacities of MOQ ( $L = 16$ ) and OQ switches ( $L = \infty$ ).

Another important performance measure for packet switches is the cell loss probability (CLP). Figure 6 compares CLP obtained for MOQ switch with the results calculated for the OQ switch. CLP for OQ switch is calculated from formula  $CLP = 1 - (\rho_0/p)$ , where  $p$  is the offered load. Proof of this formula can be found in [3]. It is intuitively clear, that the proposed switch architecture requires greater total number of memory cells ( $N$  buffers for each output port) in order to keep the same value of CLP parameter as in the case of switches with single output queue for

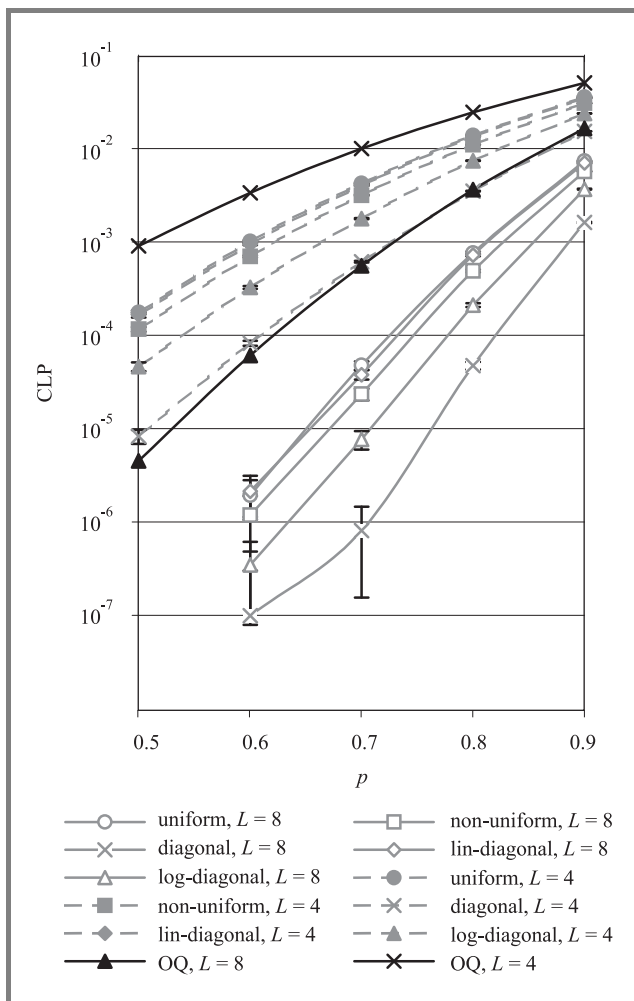


Fig. 6. The CLP for Bernoulli arrivals with different distributed traffic and for different lengths of buffers in MOQ and OQ  $16 \times 16$  switches.

each output port. From Fig. 6 it can be seen that when the length of each buffer is equal to the same value (it means that in OQ we use  $L$  memory cells for one output while in MOQ we use  $N \times L$  memory cells for one output) then CLP in MOQ switch is about one order of magnitude better than CLP in OQ switch. CLP for long buffers ( $L = 16$ ) is practically unnoticeably in our simulations.

## 6. Conclusions

We have presented the packet switch architecture which uses multiple output queueing and its performance under different traffic patterns. Our studies lead us to conclusion that buffer arrangements strategy is only important for the practical switch implementation not for performance of the switch fabric. The hardware complexity of MOQ architecture is very similar to VOQ switch but its performance is very comparable to OQ switch. The MTD is the same for both MOQ and OQ architectures for uniformly distributed traffic. The CLP is better for MOQ than for OQ, how-

ever,  $N$  times more memory cells are used in the MOQ switch architecture. The MOQ architecture is also very promising since it can naturally support multicast traffic. It should be also noted, that the MOQ switch can be also modified to support different traffic priorities. Each of  $N$  output buffers of each output port can be further divided to support different packet priorities. Evaluation of such switch architecture is also the subject of further studies.

## References

- [1] K. Yoshigoe and K. J. Christensen, "An evolution to crossbar switches with virtual output queueing and buffered cross points", *IEEE Network*, vol. 17, no. 5, pp. 48–56, 2003.
- [2] G. Danilewicz, M. Głabowski, W. Kabaciński, and J. Kleban, "Packet switch architecture with multiple output queueing", in *6th NATO Reg. Conf. Milit. Commun. Inform. Syst. RCMCIS 2004*, Zegrze, Poland, 2004.
- [3] H. J. Chao, C. H. Lam, and E. Oki, *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches in IP Routers*. New York: Wiley, 2001.
- [4] P. Giaccone, D. Shah, and S. Prabhakar, "An implementable parallel scheduler for input-queued switches", *IEEE Micro*, vol. 22, no. 1, pp. 19–25, 2002.
- [5] D. Shah, P. Giaccone, and B. Prabhakar, "Efficient randomized algorithms for input-queued switch scheduling", *Proc. Hot-Intercon. IX*, vol. 22, no. 1, pp. 10–18, 2002.
- [6] P. Giaccone, B. Prabhakar, and D. Shah, "Randomized scheduling algorithms for high-aggregate bandwidth switches", *IEEE J. Select. Areas Commun.*, vol. 21, no. 4, pp. 546–559, 2003.
- [7] Y. Jiang and M. Hamdi, "A fully desynchronized round-robin matching scheduler for a VOQ packet switch architecture", in *IEEE HPSR'01*, Dallas, USA, 2001, pp. 407–411.
- [8] A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "A framework for differential frame-based matching algorithms in input-queued switches", in *IEEE INFOCOM'04*, Hong Kong, 2004.



**Grzegorz Danilewicz** was born in Poznań, Poland, in 1968. He received the M.Sc. and Ph.D. degrees in telecommunications from the Poznań University of Technology (PUT), Poland, in 1993 and 2001, respectively. Since 1993 he has been working in the Institute of Electronics, Poznań University of Technology, where he currently is an

Assistant Professor. His scientific interests cover photonic broadband switching systems with special regard to the realization of multicast connections in such systems. He is a member of the IEEE Communication Society. He has published one book and 35 papers.

e-mail: G.Danilewicz@et.put.poznan.pl  
 Institute of Electronics and Telecommunications  
 Poznań University of Technology  
 Piotrowo st 3A  
 60-965 Poznań, Poland



**Wojciech Kabaciński** received the M.Sc., Ph.D., and D.Sc. degrees in telecommunications from the Poznań University of Technology (PUT), Poland, in 1983, 1988 and 1999, respectively. Since 1983 he has been working in the Institute of Electronics and Telecommunications, Poznań University of Technology, where he currently

is an Associate Professor. His scientific interests cover broadband switching networks and photonic switching. He has published three books, over 100 papers and has 10 patents. Professor Kabaciński is a member of the IEEE Communication Society and the Association of Polish Electrical Engineers.

e-mail: [Wojciech.Kabacinski@et.put.poznan.pl](mailto:Wojciech.Kabacinski@et.put.poznan.pl)  
Institute of Electronics and Telecommunications  
Poznań University of Technology  
Piotrowo st 3A  
60-965 Poznań, Poland



**Janusz Kleban** was born in Pobiedziska, Poland. He received the M.Sc. and Ph.D. degrees in telecommunications from the Poznań University of Technology (PUT) in 1982 and 1990, respectively. From August 1982 to November 1983 he was with Computer Centre for Building Industry in Poznań, where he worked on data transmission

systems. He has been with Institute of Electronics and Telecommunications at PUT, where he currently is an Assistant Professor, since December 1983. He is involved in research and teaching in the areas of computer networks, switching networks, broadband networks and various aspects of networking. He is author and co-author of many publications and unpublished reports.

e-mail: [jkleban@et.put.poznan.pl](mailto:jkleban@et.put.poznan.pl)  
Institute of Electronics and Telecommunications  
Poznań University of Technology  
Piotrowo st 3A  
60-965 Poznań, Poland