Analysis of a Finite Number of Deflections in Fully and Uniformly Loaded Regular Networks

Ireneusz Szcześniak

Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Bałtycka 5, 44-100 Gliwice, Poland
Tel: +48 32 2317319, Fax: +48 32 2317026
iszczesniak@iitis.gliwice.pl

Abstract. This paper presents an analytical methodology to obtain steady state throughput of a uniformly and fully loaded regular network. The network operates using deflection routing under the condition that a data packet is allowed to experience a finite number of deflections. Unlike the published analytical methods, the proposed method allows the analysis of a final number of deflections a packet can experience by means of harnessing polynomials in a novel way. The analytical results of the network throughput agree with the simulation results with relative error of 1% on average. The analysis is presented in the context of the shufflenet with the node connectivity of two. The largest network considered in the article has 896 nodes.

Key words: deflection routing, performance evaluation, simulations.

1 Introduction

The all-optical technology [1] is being developed as the promise of future communication networks. Among various technical challenges, the most notable is the lack of optical memories [2], which eliminates store-and-forward routing. Therefore the all-optical switches cannot buffer packets but only delay, and then relay them using deflection routing. Thus deflection routing is currently essential to all-optical technology.

Deflection routing copes with the lack of optical memories by misdirecting a data packet (i.e. sending it using a wrong connection) due to the lack of the desired connection. Since this routing strategy is probabilistic, a packet can travel indefinitely long in the network (the livelock problem, [3]) and impede network performance. Moreover, unlimited deflection is undesirable, as the severely delayed packets are considered lost by the TCP protocol.

Barth et al. introduce in [4] the notion of mixed routing, a new method of ensuring the maximal delay of a packet, according to which a packet is forwarded with deflection routing up to the moment when it reaches some fixed number of

deflections. After that the packet is relayed in the network according to the rules of convergence routing on an Euler cycle. In order to evaluate the performance of their method, it is first necessary to evaluate the performance of deflection routing with a finite number of allowed deflections, which motivates this article.

We describe an analytical method to evaluate the performance of a regular network which operates using deflection routing under the condition of a finite number of deflections. After a packet reaches the limit of deflections, it is eliminated from the network and lost. The analysis applies to regular networks under a full and uniform traffic load.

The analysis is based on the approach of Acampora and Shah presented in [5], where they analyze uniformly loaded regular networks, but with no restrictions imposed on the number of deflections (packets can experience any number of deflections). We adopt their method to facilitate the analysis under the condition of a finite number of deflections. We encourage a reader unfamiliar with their method to read their article.

We extend their approach by the introduction of polynomials. Acampora and Shah express the probabilities of packet presence by real numbers, while we express these probabilities by polynomials of real coefficients. A polynomial serves as a tool to remember a number of deflections a packet undergoes.

2 Network Model

The network is homogeneous: every node of the network functions both as a routing node for packets in transit and as an access point where packets can enter or exit the network.

The network works synchronously, i.e. time is divided into time slots and is the same for every node. Packet transmissions take place at the beginning of time slots.

The load of the network is full and uniform. During every time slot at every node there are always enough packets waiting for admission, so that the network is constantly fully loaded. The probabilistic nature of packet arrivals before their admission is irrelevant to this analysis. A packet's source and destination are chosen uniformly from all the nodes. To ensure uniform traffic we allow a node to be both a packet's source and destination.

The analysis can be applied to a regular network or, using the graph theory nomenclature, to a node symmetric network. In a regular network every node is equivalent to every other node, which, together with the assumption of traffic uniformity, allows the analysis to be carried out for one node only.

2.1 Shufflenet

To ease the description of our analysis we shall present it for one regular network. For this purpose we choose the shufflenet with the node connectivity, also called node degree, of two: two inbound and two outbound links. The network has $N = n2^n$ nodes and $M = 2N$ unidirectional links, where n is the network size.

There are n columns of nodes, where each column contains 2^n nodes. A sample network of size $n = 2$ is shown in Fig. 1. Nodes in the first column and the last column are the same nodes.

This network can operate with mixed routing since it has an Euler cycle. A necessary and sufficient condition for a network to have an Euler cycle is that the node connectivity of its every node is even. Thus, the considered network has an Euler cycle because the node connectivity of its every node is two.

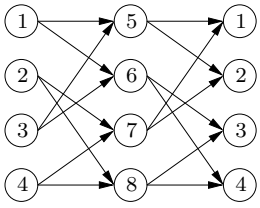

Fig. 1. A shufflenet of size $n = 2$.

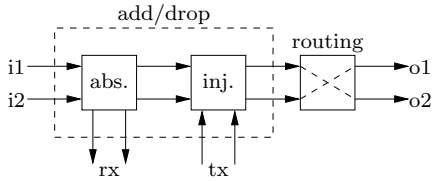

Fig. 2. Logical structure of a node.

2.2 Deflection Routing

Deflection routing does not buffer packets as store-and-forward routing does but instead relays them during the next time slot after the time slot in which they arrived.

A packet at a node may have a *preferred link* which is a link that yields a shortest path journey to its destination. A link is not considered preferred when every link at the node offers a journey of the same length.

For a packet every node can be only a "care" node or a "don't care" node. At a "care" node the packet has a preferred link, as opposed to a "don't care" node where neither of the links is preferred.

At a node a packet is either deflected (it undergoes a deflection) or correctly routed. A packet is deflected only at "care" nodes when it is refused a preferred link, and it is correctly routed otherwise.

For instance, consider the network in Fig. 1 and a packet which is destined for node 2. Node 5 is a "care" node where the bottom link is preferred. There the packet experiences a deflection when it is sent along the upper link. At node 6 the packet does not possess a preferred link, is insusceptible to a deflection, and therefore node 6 is a "don't care" node.

Let an integer S be defined as the maximal number of deflections a packet can suffer. A packet is allowed to have at most $(S-1)$ deflections, and is eliminated from the network when it undergoes the Sth deflection. Every packet keeps a counter of its number of experienced deflections.

A packet in the network always either is delivered at the destination or is eliminated from the network. A successful packet experiences during its journey

at most $(S-1)$ deflections and finally reaches the destination. Conversely, for an unsuccessful packet the number of deflections increases to S, and the packet is eliminated from the network.

2.3 Node Type

Each node is of the type shown in Fig. 2, which is a modified version of the type proposed in [6]. There are two major blocks: "add/drop" for receiving and injecting packets, and "routing" for directing packets to appropriate neighbor nodes. The "add/drop" block consists of two minor blocks, one absorbs packets, the other injects packets. The node connectivity is two (two inputs "i1, i2", and two outputs "o1, o2"), but the analytical method is not limited to this case only and should apply to any other node connectivity.

When a packet arrives at a node it is absorbed by the "abs." block on condition that this node is the packet's destination. In every time slot at most two packets can be absorbed, which are then relayed by the "rx" links. Packets are injected into the network by the "inj." block that can receive at most two packets from its four inbound links. First, the packets from the "abs." block are accepted. Next, new packets from the "tx" links are admitted if no packet or one packet arrived from the "abs." block. New packets are admitted even if they cause a contention.

The second major block is responsible for routing packets. Packets arrive along the block's inbound links, and are directed appropriately to the outbound links. If a contention takes place, then the winning packet uses the preferred outbound link, while the deflections counter of the losing packet is increased by one. The losing packet is eliminated from the network provided that its deflections counter reaches the value of S, or otherwise it is sent to the remaining outbound link.

A packet's elimination renders one output link (either "o1" or "o2") idle, which cannot be utilized to transmit any other packet. Therefore the number of links used by the eliminated packet during its journey is the packet's number of hops increased by one.

2.4 An Upper Bound on the Number of Used Links

In this subsection we obtain an upper bound on the number of links any packet can use, which is defined as an integer K. The upper bound is derived for any network, not only for the shufflenet. A successful packet, one that gets to its destination, makes at most K hops, whereas an unsuccessful packet, one that is eliminated from the network, uses at most K links.

To derive the value of K, consult Fig. 3, which presents the worst scenario of a packet journey. The packet enters the network at the source node A and is bound to the destination node B. The node A is at the greatest possible distance from the node B, i.e. at the distance equal to the network's diameter (the network's diameter is the length of the longest path of all shortest paths between any pair of nodes). The packet travels without a deflection from the node A to the node

C using $(diameter - 1)$ links. Right at the node C the packet experiences its first deflection, and is sent back to the node A using one link. Now the packet is again at the source node A where up to now it has used *diameter* links and experienced one deflection. This situation happens exactly $(S - 1)$ times, after which the packet is at the node A with $(S-1) \cdot diameter$ used links and $(S-1)$ experienced deflections.

Next, the packet is at the node C after $(diameter - 1)$ hops. From here a packet either gets to the node B in one hop, or is eliminated and uses one link. Therefore the worst scenario requires a packet to use K links:

$$K = S \cdot diameter. \tag{1}$$

For instance, for the network from Fig. 1 and $S = 1$ (no deflections allowed), the value of K equals 3.

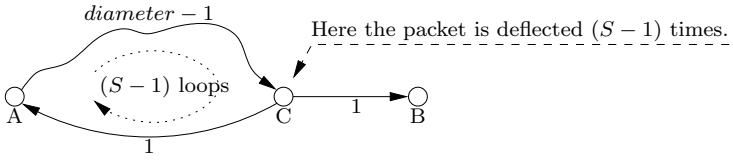

Fig. 3. The worst scenario of delivering a packet.

3 Analysis

The objective of the analysis is to obtain the steady state throughput λ of the system, i.e. the average number of packets that reach their destinations in a single time slot. Note that the number of packets admitted into the network is not equal to the number of packets arriving at their destinations because packets can be eliminated from the network.

The analysis in the following subsections is presented with the top-down approach: the throughput is derived using values that are described in detail in subsequent subsections.

3.1 Deriving λ

To obtain the network throughput first definitions of some probabilities are introduced. The probability that a packet is delivered to its destination using k links is denoted by:

$$\begin{aligned} P_D[k] = P[&\text{a packet is delivered} \\ &\text{using } k \text{ links}], k = 1, \ldots, K, \end{aligned} \tag{2}$$

and the probability that a packet reaches its destination is:

$$P_D = \sum_{k=1}^{K} P_D[k]. \tag{3}$$

The probability that a packet is eliminated and used k links is denoted by:

$$\begin{aligned} P_E[k] = P[&\text{a packet is eliminated and} \\ &\text{used } k \text{ links}], k = 1, \ldots, K, \end{aligned} \tag{4}$$

and the probability that a packet is eliminated is:

$$P_E = \sum_{k=1}^{K} P_E[k]. \tag{5}$$

Due to the limited number of deflections, every packet in the network eventually either reaches its destination or is eliminated from the network, and therefore:

$$P_D + P_E = 1. \tag{6}$$

Packets that have experienced fewer than S deflections and have not been eliminated from the network, usually reach their destinations using D_D links:

$$D_D = \frac{1}{P_D} \sum_{k=1}^{K} k P_D[k]. \tag{7}$$

Correspondingly, eliminated packets (every such packet has experienced exactly S deflections) use on average D_E links:

$$D_E = \frac{1}{P_E} \sum_{k=1}^{K} k P_E[k]. \tag{8}$$

Not all the packets present in the network reach their destinations, but only a fraction η of those packets do,

$$\eta = \frac{D_D P_D}{D_D P_D + D_E P_E}. \tag{9}$$

The number of packets present in the network is equal to the number M of the network links, since the network is fully loaded. However, the average number of packets that are present in the network and that reach their destinations is:

$$M' = \eta M. \tag{10}$$

Finally, from Little's law the network throughput is given by

$$\lambda = \frac{M'}{D_D}. \tag{11}$$

To compute λ we need not only the value of M, which is provided by the user, but more importantly the values of $P_D[k]$ and $P_E[k]$ for $k = 1, \ldots, K$. This is the crux of the analysis, discussed in detail in the following subsection.

3.2 Deriving $P_D[k]$ and $P_E[k]$

To obtain $P_D[k]$ and $P_E[k]$, the behavior of an average packet in the system is modeled by the behavior of the test packet. Since the network is regular and the traffic is uniform, the test packet is traced on its journey not to every node, but only to node 1.

We calculate the probabilities that the test packet resides at a particular node of the network during a particular time slot. These probabilities are stored in the vectors P_k for $k = 0, 1, 2, \ldots, K$, where K is the maximal number of time slots any packet can be present in the network. The probabilities of the test packet residency in the network during the kth time slot are expressed by the vector P_k:

$$P_k = \begin{pmatrix} p_{1,k}(x) \\ \vdots \\ p_{i,k}(x) \\ \vdots \\ p_{N,k}(x) \end{pmatrix}. \tag{12}$$

with polynomial elements $p_{i,k}(x)$ of the form:

$$p_{i,k}(x) = \sum_{j=0}^{S-1} p_{i,k}[j]x^j. \tag{13}$$

Each polynomial $p_{i,k}(x)$ provides information on the probability that the test packet resides at node i during the kth time slot. This probability is hereafter also referred to as *residency probability*. The polynomial is of the degree at most $(S-1)$, $deg\ p_{i,k}(x) \leq S-1$. The coefficient $p_{i,k}[j]$ provides the probability that not only the packet resides at node i during the kth time slot, but also that the test packet suffered j deflections on its journey up to this point. The polynomial $p_{i,k}(x)$ for $x = 1$ becomes: $p_{i,k}(1) = \sum_{j=0}^{S-1} p_{i,k}[j]$ and expresses the probability of the test packet presence at the ith node during the kth time slot regardless of the specific number of suffered deflections.

With the probability $1/N$ the test packet starts a journey at every node of the network, including the destination node (i.e. the 1st node) to preserve traffic uniformity. We realize that it is unrealistic for a packet to have the same node as the source and the destination. Nonetheless, under this assumption the presented analysis is more accurate. The vector P_0 is:

$$P_0 = \left(\frac{1}{N}, \ldots, \frac{1}{N}\right)^T. \tag{14}$$

The average value of the probability of a packet deflection at a "care" node is denoted by d and called *the probability of deflection*. This value is assumed to be constant and equal for every node. Conversely, $p = (1-d)$ is the probability that

the test packet at a "care" node is not misrouted, but assigned to its preferred outbound link.

Let us consider three examples of calculating residency probabilities. For each example the setting is identical: the considered network is shown in Fig. 1; during time slot 0 the test packet resides at every node with an equal probability; the test packet is destined to node 1. We are interested in the residency probabilities for nodes 3, 1 and 2 during time slot 1.

First, let us study the probability that the test packet resides at node 3. The packet can reach node 3 from nodes 6 and 8, which are "don't care" nodes with respect to node 1. From a "don't care" node the packet departs on any of the node's outbound links with an equal probability (the probability is 1/2 for the node connectivity of two), and so: $p_{3,1}(x) = \frac{1}{2} \cdot p_{5,0}(x) + \frac{1}{2} \cdot p_{7,0}(x) = \frac{1}{8}$.

Second, consider the probability that the test packet arrives at node 1. There the test packet can arrive from nodes 5 and 7, which are both "care" nodes. Hence the test packet at these two nodes has to be routed according to its preference: $p_{1,1}(x) = p \cdot p_{5,0}(x) + p \cdot p_{7,0}(x) = \frac{p}{4}$.

The last example is most interesting. Node 2 can be reached by the test packet from nodes 5 and 7 under the condition of losing a contention at each of the two nodes. Losing a contention occurs with the probability d and thus: $p_{2,1}(x) = xd \cdot p_{5,0}(x) + xd \cdot p_{7,0}(x) = \frac{d}{4}x$. Multiplication by x expresses a deflection. Therefore the probability $\frac{d}{4}x$ provides the information that the packet experienced one deflection.

A very important fact to stress is that multiplication of a residency probability $p_{i,k}(x)$ by xd may result in a polynomial of the degree equal to S. As stated earlier, a polynomial can be of the degree at most $(S-1)$. When such an event arises, the polynomial's highest term is discarded, which corresponds to elimination of the test packet from the network.

To calculate the residency probabilities for every node during time slot 1, the matrix T_0 is introduced. The polynomial element $t'_{i,j}(x)$ of the matrix expresses the probability of the test packet transition from node j to node i, provided that the test packet resides at node j. Having defined T_0, the computation of the vector P_1, i.e. residency probabilities for every node, is elementary:

$$P_1 = T_0 P_0. \tag{15}$$

The following is the transition matrix T_0 for the network shown in Fig. 1:

$$T_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & p & 0 & p & 0 \\ 0 & 0 & 0 & 0 & xd & 0 & xd & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ p & 0 & p & 0 & 0 & 0 & 0 & 0 \\ xd & 0 & xd & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & p & 0 & 0 & 0 & 0 \\ 0 & xd & 0 & xd & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{16}$$

The elements of the transition matrix are deduced as follows. If a link between nodes j and i does not exist, then the test packet cannot make a transition, and

for that reason $t'_{i,j}(x)$ equals 0. The test packet leaves a "don't care" node j by any of the two outbound links with an equal probability $t'_{i,j}(x) = 1/2$. Transition along a preferred link is represented by $t'_{i,j}(x) = p$, while $t'_{i,j}(x) = xd$ corresponds to a transition with a deflection. These rules are summarized below:

$$t'_{i,j}(x) = \begin{cases} 0 & \text{for no transition} \\ \frac{1}{2} & \text{for routing without preference} \\ p & \text{for preferred routing} \\ xd & \text{for routing with a deflection.} \end{cases} \tag{17}$$

The first time slot is special, because the test packet is allowed to leave the destination node, whereas during every other time slot the packet is absorbed by the destination. The form of the matrix T_0 reflects this exception by having two nonzero elements in the first column. For every other time slot the transition matrix is different and denoted by T:

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & p & 0 & p & 0 \\ 0 & 0 & 0 & 0 & xd & 0 & xd & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & p & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & xd & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & p & 0 & 0 & 0 & 0 \\ 0 & xd & 0 & xd & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{18}$$

The following matrix equation serves to obtain the vectors P_k:

$$P_k = TP_{k-1}, k = 2, \ldots, K. \tag{19}$$

Up to this point the polynomial vectors P_k of probabilities have been derived, from which the values of $P_D[k]$ and $P_E[k]$ are calculated. Let us start with providing the easier one:

$$P_D[k] = p_{1,k}(1), k = 1, \ldots, K. \tag{20}$$

The harder one is the probability of a packet elimination $P_E[k]$, which is equal to the product of the probability of deflection d and the probability that the test packet which endured $(S-1)$ deflections is subject to routing at a "care" node during the kth time slot:

$$P_E[k] = d \cdot \sum_{i=1}^{N} b_{i,k-1}[S-1], k = 1, \ldots, K, \tag{21}$$

where the vector B_k, with elements $b_{i,k}$, represents the residency probabilities of the test packet being subject to routing at "care" nodes during the previous time slot. The vector B_k is defined as:

$$B_k = (\alpha_1 e_{1,k}(x), \ldots, \alpha_i e_{i,k}(x), \ldots, \alpha_N e_{N,k}(x))^T, \tag{22}$$

where $\alpha_i = 1$ if node i is a "don't care" node (with regard to node 1), $\alpha_i = 0$ if node i is a "care" node (with regard to node 1), and elements $e_{i,k}(x)$ belong to the vector E_k:

$$E_k = \begin{cases} P_0 & \text{for } k = 0 \\ (0, p_{2,k}(x), \ldots, p_{N,k}(x))^T & \text{for } k = 1, 2, \ldots, L. \end{cases} \tag{23}$$

The vector E_k expresses the probabilities that the test packet is subject to routing at the kth time slot, while the vector B_k expresses the probabilities that the test packet at the kth time slot is subject to routing at "care" nodes.

Now, the matrices T_0 and T depend on the value of d. Following [5] the value of d for a full and uniform load is:

$$d = \frac{1}{4}(1 - P_{dc}), \tag{24}$$

where P_{dc} is defined in the following subsection.

3.3 Deriving P_{dc}

P_{dc} is the probability of encountering a "don't care" node (with regard to node 1) by the test packet any time it arrives at a node during its entire journey. It is defined as follows:

$$P_{dc} = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N} b_{i,k}(1)}{\sum_{k=1}^{K} \sum_{i=1}^{N} e_{i,k}(1)}. \tag{25}$$

In [5] the method for obtaining P_{dc} substantially differs from the method presented here. There the probability is calculated for the test packet when it reaches the destination, while our method takes into account not only delivered but also eliminated packets.

The probability P_{dc} depends on the vectors P_k. These vectors rely on the probability of deflection d, which in turn depends on P_{dc}. Thus a more precise value of P_{dc} is acquired by successive approximations.

The initial guess for the value of P_{dc} is the ratio of the number of "don't care" nodes to the number of all nodes, which is the first rough approximation. For the approximated value of P_{dc} the value of d is calculated from (24), new matrices T_0, T are generated, then vectors P_k are computed using (15), (19), and finally a new and refined value of P_{dc} is obtained from (25).

The process is repeated the necessary number of times until the desired precision of P_{dc} is reached. In our calculations the precision of the order of 10^{-4} was attained after a few iterations, as the successive values of P_{dc} converged smoothly and fast.

Let us evaluate the running time of the algorithm. One iteration, which produces the next approximation of P_{dc}, requires $O(KN^2)$ multiplications of two polynomials: K number of vectors P_k need to be obtained, and each of them costs $O(N^2)$ multiplications. A multiplication of two polynomials costs $O(S^2)$, and therefore the overall cost of an iteration is $O(KN^2S^2)$.

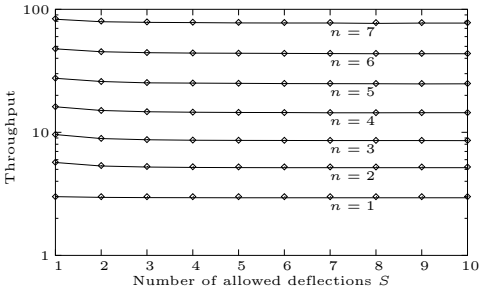

Fig. 4. Network throughput for various network sizes n.

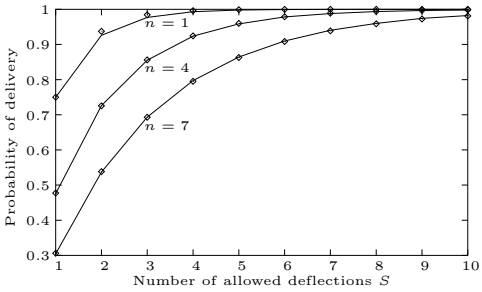

Fig. 5. Probability of a packet delivery for several networks.

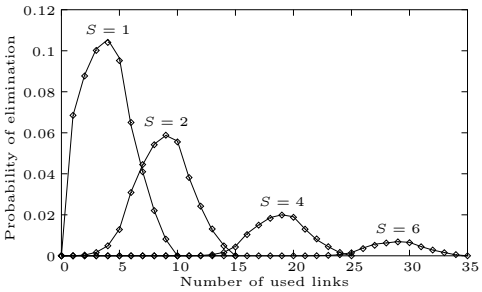

Fig. 6. Probability of a packet elimination for network size 5.

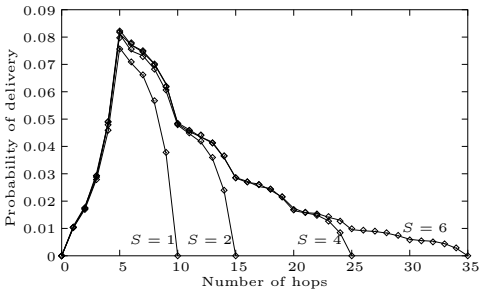

Fig. 7. Probability of a packet delivery for network size 5.

4 Results and Verification

This section presents our results of the analysis and their verification with simulations carried out within the framework of the *Omnet++* software [7]. The results are presented in Figures 4, 5, 6, 7 which display the analysis results as fine squares, whereas the discrete simulation results are presented as lines to distinguish them from the analysis results.

In comparison with simulations, the relative error of our analytical throughput was on average below 1%, and for the network of size $n = 1$ it equaled approximately 2%. The analysis of the network of size $n = 1$ was the most inaccurate, whereas the accuracy of the analysis was increasing as we were enlarging the size of the network.

Fig. 4 shows the network throughput λ as a function of the number of allowed deflections S for various network sizes n. The throughput is similar for different values of S. When the value of S is small, then packets do not live long in the network, and in their place new packets are admitted. As a result, even though many packets are eliminated (more than 50%), other packets are promptly ad-

mitted. When the value of S is large, then almost no packet is eliminated, but there are long living packets which block the network and lower the thoughput.

Fig. 5 presents the probability P_D with which a packet successfully arrives at the destination provided it is allowed to undergo S deflections. There are three series for $n = 1, 4, 7$. As expected, the probability increases in tandem with the number of allowed deflections.

The probability $P_E[k]$, that a packet in the network of size $n = 5$ is eliminated and that it used k number of links, is shown in Fig. 6 for $S = 1, 2, 4, 6$.

The last figure (Fig. 7) depicts the probability $P_D[k]$ with which a packet is delivered to its destination in k number of hops in the network of size $n = 5$. There are four series for $S = 1, 2, 4, 6$.

5 Conclusion

The article presented above discusses a novel methodology which incorporates polynomials of real coefficients to allow the analysis of a finite number of deflections. Simulations have confirmed that this method is accurate and serves its purpose.

Future work includes an analysis of mixed routing [4], to which our methodology should be applicable. Another future plan is to further increase the accuracy of our analysis.

Acknowledgment

The author wishes to thank his wife Arlena for assistance with English and the referees of this article for their constructive comments.

References

1. P. Green, "Progress in Optical Networking," *IEEE Commun. Mag.*, pp. 54-61, Jan. 2001.
2. L. Xu, H. G. Perros, G. Rouskas, "Techniques for Optical Packet Switching and Optical Burst Switching," *IEEE Commun. Mag.*, pp. 136-142, Jan. 2001.
3. J. T. Brassil, R. L. Cruz, "Bounds on Maximum Delay in Networks with Deflection Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 7, pp. 724-732, July 1995.
4. D. Barth, P. Berthomé, T. Czachórski, J. M. Fourneau, C. Laforest, S. Vial, "Mixed Deflection and Convergence Routing Algorithm: Design and Performance," in *Proc. Euro-Par 2002*, pp. 767-774, Aug. 2002.
5. A. S. Acampora, S. I. A. Shah, "Multihop Lightwave Networks: A Comparison of Store-and-Forward and Hot-Potato Routing," *IEEE Trans. Commun.*, vol. 40, pp. 1082-1090, June 1992.
6. A. Bononi, P. R. Prucnal, "Analytical Evaluation of Improved Access Techniques in Deflection Routing Networks," *IEEE/ACM Trans. Networking*, vol. 4, no. 5, pp. 726-730, Oct. 1996.
7. A. Varga, "The OMNeT++ Discrete Event Simulation System," in *Proc. European Simulation Multiconference (ESM'2001)*, Prague, Czech Republic, June 2001.