Jan Długosz University in Częstochowa
Scientific Issues, Mathematics XVI, Częstochowa 2011

EVALUATION OF A METHOD FOR ENUMERATING THE MOST PROBABLE PACKET ARRANGEMENTS

Ireneusz Szcześniak

Institute of Computer and Information Sciences
Częstochowa University of Technology
ul. Dąbrowskiego 73, 42-200 Częstochowa, Poland
e-mail: iszczesniak@icis.pcz.pl

Abstract. The problem of calculating routing probabilities in packet synchronous networks, such as optically-switched packet networks, involves enumerating packet arrangements. Previously we published a method for enumerating the most probable packet arrangements, and in this article we present its novel graph interpretation and evaluate the method for several stop conditions and for the Poisson and geometric probability distributions of arriving packets.

1. Introduction

Optical packet switching (OPS) is a technology that could be deployed in future optical networks with packet switching [1]. In synchronous OPS packets are sent out a node at the beginning of a time slot, and packets last one time slot. Performance evaluation of OPS networks can be used for off-line network planning and on-line evaluation to proactively provision optical resources with OpenFlow [2].

One of the problems in the performance evaluation of synchronous OPS is the enumeration of the possibilities of packet arrivals at a node, which we call *packet arrangements* or just arrangements. Enumerating all arrangements may be impractical and unnecessary, and so in [3] we published an algorithm which efficiently finds any number of the most probable arrangements.

2. Problem statement

A flow is a sequence of packets grouped together according to a given criterion, such as the same source and destination nodes. A flow is described by a probability distribution of the number of packets that arrive at a node in a time slot. There are R flows for which the probability distribution functions are given by vector $F = (f_1, \dots, f_r, \dots, f_R)$. We assume that the distribution functions f_r are independent.

An arrangement takes place at a node in every time slot, and it is described by a discrete random variable $X = (x_1, \dots, x_r, \dots, x_R)$, where x_r is a discrete random variable of the number of packets that belong to flow r. The probability of arrangement X can be calculated as given by Eq. (1):

$$P(X, F) = \prod_{r=1}^{R} f_r(x_r). \tag{1}$$

Arrangement X can be alternatively described by a random variable $Y = (y_1, \dots, y_r, \dots, y_R)$, where the number of packets x_r is the y_rth most probable for flow r. For distribution $f_r(x_r)$, we can find the value of x_r that yields the highest value of $f_r(x_r)$, i.e. the mode, and denote it $\Gamma_r(1)$. The next most probable number is $\Gamma_r(2)$, and $\Gamma_r(y_r)$ for the further y_r-th most probable number of packets.

The number of possible arrangements is the product of the domain sizes of flow distributions. If a distribution for a flow is infinite, the number of arrangements is infinite too. Even if the number of arrangements if finite, enumerating all of them may be unnecessary.

Instead of enumerating arrangements in an arbitrary order, we find a sequence $X_e = (x_{1,e}, \dots, x_{r,e}, \dots, x_{R,e})$ of the most probable arrangements, where X_1 is the most probable one, while the next arrangements X_e have nonincreasing probabilities, i.e. $P(X_e, F) \geq P(X_{e+1}, F)$.

3. Algorithm

The algorithm finds efficiently any number of the most probable arrangements, and stops according to the stop condition provided by the user. In this section we present a novel graph interpretation of the algorithm.

There is given a weighted directed graph. A vertex represents an arrangement, and the label of the vertex represents the probability of the arrangement. The source vertex represents arrangement Y_1. An edge leaving vertex Y_e represents a possible way of obtaining a different arrangement by changing in arrangement Y_e the number of packets of a single flow to the next most

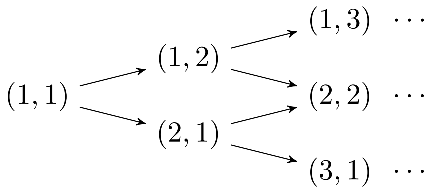


Figure 1: A graph of the possible arrangements for two flows.

probable value. The weight of an edge is negative and represents the decrease in probability of arrangement Y_e caused by the change.

Figure 1 depicts a graph for two flows, but the vertex labels and edge weights are omitted. The source vertex $Y_1 = (1, 1)$ is the most probable, and from it two new arrangements can be obtained: $(1, 2)$ and $(2, 1)$, but which of them is Y_2 depends on their probabilities.

In such a graph, the algorithm finds the longest paths from the source vertex to other vertexes. For this purpose we adapted the Dijkstra algorithm, even though the input data violates the conditions of the Dijkstra algorithm, that the edges are nonnegative and that vertex labels are nondecreasing.

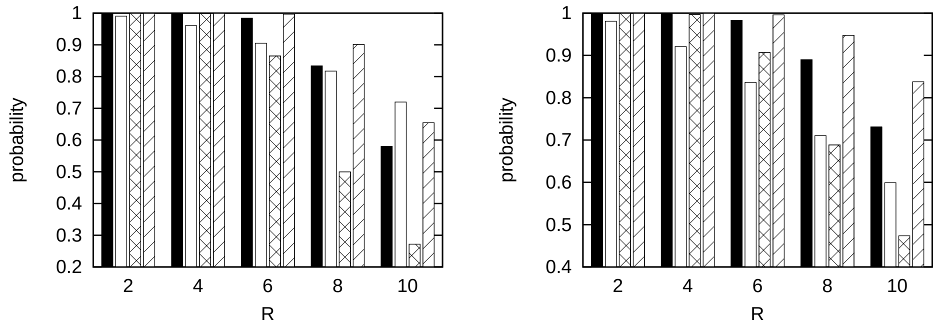
The adaptation involved a number of changes. First, the vertexes in the priority queue are sorted not with increasing labels, but with decreasing labels, as we search for vertexes with decreasing probabilities. Second, the relaxation of an edge is performed not when there is an edge that yields a lower label of a vertex, but when an edge yields a higher label of a vertex. Finally, the source vertex is labeled not with zero, but with the probability of the most probable arrangement.

4. Evaluation

The evaluation discussed in this section was implemented as part of the test suite of the OPUS software. The program file is test/test_arr_queue.cc, and is available for download at [4].

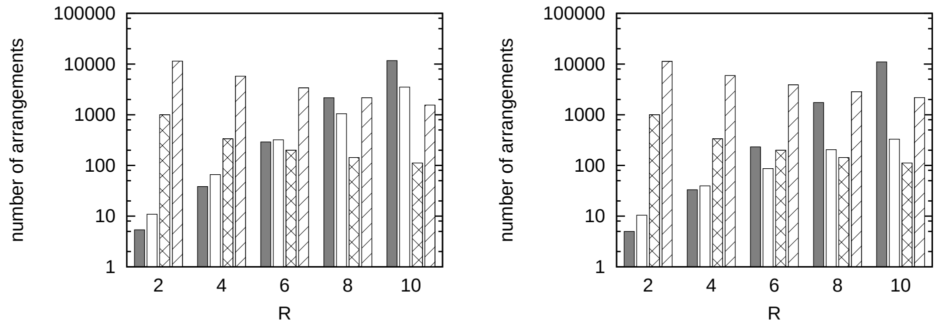
Figures 2, 3, 4 and 5 show results of finding arrangements for five stop conditions. In each of the figures there are two subfigures, one for the flows of the Poisson distribution, the other for the flows of the geometric distribution. The reported values are a function of the number of flows R.

A stop condition, when met, causes the algorithm to stop finding further arrangements. A number of stop conditions can be devised, but we define five stop conditions. They can be used separately or in a compound statement. The algorithm stops when: 1) a number of found arrangements reaches a given value, 2) the aggregate probability of found arrangements exceeds a given value, 3) the ratio of the probability of the found arrangement to



(a) The Poisson case (b) The geometric case

Figure 2: Aggregate probability of considered arrangements.

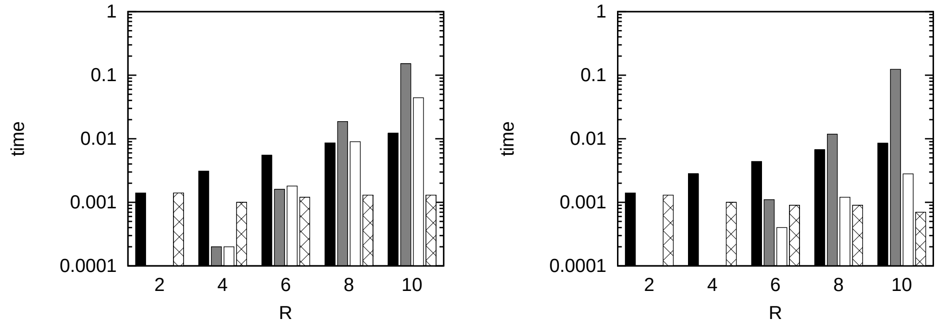


(a) The Poisson case (b) The geometric case

Figure 3: The number of considered arrangements.

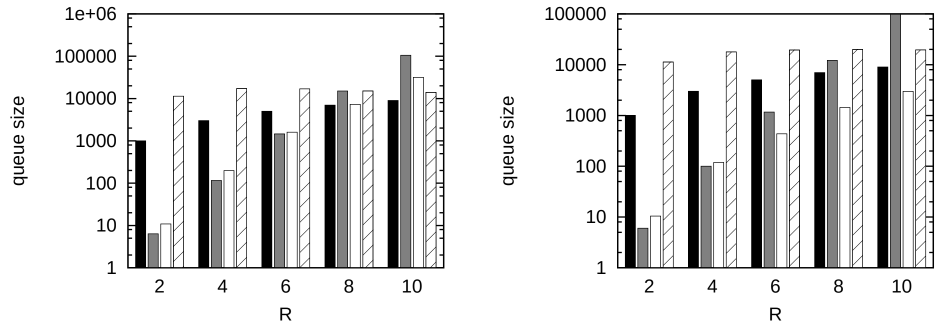
the probability of the most probable arrangement drops below a given value, 4) the size of the priority queue of arrangements exceeds the given value, 5) the execution time exceeds the given value.

Each bar in the figures reports a mean value of a sample of 100 test cases. A test case has the parameters of its flows chosen at random: for the Poisson flows the intensities $\lambda \in (0, 1)$, for the geometric flows the probabilities of success $p \in (0.5, 1)$. The standard errors of the means are small, and they are not reported in the figures. Results for each of the five conditions are shown in each figure with the same bars: the black bars ( ) are for the limit of 1000 on the number of arrangements, the gray bars ( ) are for the limit of 0.5 on the aggregate probability, the white bars ( ) are for the limit of 10^{-2} on the probability ratio, the checked bars ( ) are for the limit of 1000 on the arrangement queue size, and the slashed bars ( ) are for the limit of 10 ms on the execution time.



(a) The Poisson case (b) The geometric case

Figure 4: The time needed to find the arrangements.



(a) The Poisson case (b) The geometric case

Figure 5: The size of the queue.

Figure 2a shows the aggregate probability of the considered arrangements. For up to 6 flows, the number of considered arrangements were enough to result in the aggregate probabilities above 0.9. For 10 flows the probability was slightly above 0.7. Clearly, the stop condition for the number of considered arrangements yields aggregate probability that decreases faster than a linear function as the number of flows decreases. The other conditions gave results of similar characteristic, but the time condition gave the least decreasing aggregate probabilities as the number of flows increased. Figure 2b shows similar results for the geometric flows. Figures 3, 4, 5 show the results on a logarithmic scale. Most of the results are an exponential function of the number of flows, as the bars in the plots raise and fall like a linear function. However, there is one exception: the execution time with the limit on the queue length (Fig. 5, slashed bars) depends slightly on the number of flows, which suggests that the operation of inserting of an element into the queue dominates.

5. Conclusion

The evaluation showed that the proposed algorithm finds efficiently the most probable arrangements, because we adopted the efficient Dijkstra algorithm. However, as the number of flows increases, the number of arrangements required to obtain a given aggregate probability increases exponentially. For a small number of flows the algorithm performs satisfactorily, but for larger number of flows the algorithm is rendered unusable, because the number of arrangements to consider (i.e. the size of the problem) grows exponentially.

In the future work, one could research how the algorithm performed had the distributions been correlated.

References

[1] R. Nejabati, G. Zervas, D. Simeonidou, M.J. O'Mahony, D. Klonidis. The OPORON project: demonstration of a fully functional end-to-end asynchronous optical packet-switched network. *Journal of Lightwave Technology*, **25**(11), 3495–3510, 2007.

[2] S. Das, G. Parulkar, N. McKeown. Unifying packet and circuit switched networks. In: *2009 IEEE GLOBECOM Workshops*, pp. 1–6, December 2009.

[3] I. Szcześniak, T. Czachórski. Performance evaluation of a bufferless packet-switched node. In: *Proceedings of the Ninth International Conference on Transparent Optical Networks, ICTON 2007*, vol. 3, pp. 117–120, 2007.

[4] I. Szcześniak. The website of OPUS, the Optical Packet Switching Universal Solver. `http://www.irkos.org/opus`, April, 2011.