Approximate Analytical Performance Evaluation of Synchronous Bufferless Optical Packet-Switched Networks

Ireneusz Szcześniak, *Member, IEEE,* Biswanath Mukherjee, *Fellow, IEEE,* and Tadeusz Czachórski

***Abstract*—We present a method for analytical performance evaluation of synchronous optical packet-switched backbone networks. The proposed method is applicable to networks of any topology built of nodes of any degree with any number of wavelengths, and capable of allowing for deflections. The method provides not only general results such as fiber utilization, but also detailed results on where and at what mean rates packets travel. We analyze the steady-state performance of a network loaded with independent flows. The admission control admits in a fair way at most as many packets as there are output slots available, while the routing algorithm uses classes, distances, deflections, and wavelength conversion. We compare the results of our method to simulation results for a large set of randomly-generated test cases, and conclude that the proposed method yields reasonably accurate results.**

I. Introduction

The operational backbone networks are able to switch optically only wavelengths. A wavelength carries cells or packets, but they are switched by electronic hardware. Since the late 1980s, it has been proposed to move some functionality of the electronic hardware to the optical hardware in order to switch optically at the granularity below a wavelength. Since then, several sub-wavelength technologies have been devised [1], one of which is optical packet switching (OPS) [2], [3].

OPS has a chance to become a client of incumbent WDM networks [4]. The optical packet switches would communicate with each other through circuits provided by the WDM network, and would interface with client IP networks. In the OpenFlow architecture [5], it is proposed to adjust the capacities of IP links with dynamic circuit switching based on the measurements of packet losses and queue sizes in monitored network elements. OpenFlow admits to the network new flows, measures their impact, and then decides whether to establish new circuits to improve network performance. However, new circuits could be established proactively before new flows are admitted based on the predicted network performance. Such a prediction should take into account the current state of the network, and the service level agreements (SLA) of both the already-supported flows and the new flows.

Nowadays, simulation is mostly harnessed for such evaluation [6], but we present an *analytical* approach. While the proposed method has been developed for an OPS network, it should be applicable to other types of synchronous packet networks.

We advance the analytical performance evaluation of OPS in three ways. First, our method is applicable to any topology, while previous analytical methods for OPS were developed for restricted network topologies such as the Manhattan Street Network or the Shufflenet.

Second, we present a detailed probabilistic analysis of a network node, which was introduced without details in [7]. In the current article, we also present the algorithm for finding the most probable packet arrangements. Performance evaluation of a single node was usually carried out for nodes of small degrees and one wavelength per fiber, which limits analysis to simple networks [8], [9], [10]. It is not so in our work: we allow any node degree and any number of wavelengths.

Finally, we offer a method of modeling analytically the important constraints on the number of hops a packet can make and the distance a packet can travel, which to the best of our knowledge have not been previously addressed analytically. Most of previous works assumed that packets can stay in the network indefinitely, while the number of deflections was modeled in [10] and [11].

The chief constraint of OPS is the limited buffering, i.e., the lack of optical RAM, which is being tackled with rudimentary optical packet buffers built from optical delay lines. Bufferless OPS networks can be attractive, because they do not rely on optical packet buffers, which are expensive, degrade the quality of the optical signal, and can be bulky. Therefore, in this article, we assume bufferless nodes.

We evaluate admission control and routing. The task of admission control is to manage the packets asking for admission from the access network to the backbone network. The routing algorithm directs packets to their destinations, and also tries to optimize the use of scarce OPS resources. The routing algorithm is responsible for sending packets to appropriate output fibers on appropriate wavelengths. A fiber can accommodate in a time slot at most as many packets as it has wavelengths, and if more packets compete for that fiber, the routing algorithm needs to resolve the contention among the packets. Several methods have been proposed for contention resolution, which are typically DiffServ rules for establishing the routing order of packets [12].

Since we assume that there are no buffers, a packet that loses

Manuscript received April 18, 2011; revised August 20, 2011. This work was sponsored in part by the Polish-U.S. Fulbright Commission and the Polish grant N N516 479640.

I. Szcześniak is with the Institute of Computer and Information Sciences of the Częstochowa University of Technology, Częstochowa, Poland.

B. Mukherjee is with the University of California, Davis.

T. Czachórski is with the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, Gliwice, Poland.

a contention has to be either dropped or misrouted to a fiber with an available wavelength. The routing that can misroute packets is called *deflection routing*. The rationale behind deflection routing is to save packets from being dropped by using fibers between nodes as buffers. Deflection routing can improve network performance as reported in [6].

We ensure that our method is applicable also when deflection routing is used. Deflections make the analysis hard, because we need to take into account loops that packets make when they are deflected. Analysis is simpler when packets do not make loops, as is the case when no deflection routing or when loopless deflection routing is used [13].

Throughout the article, we use the Poisson distribution, which, simplistic as it is, suffices for simple and approximate evaluation. Recent research suggests that this distribution can model the backbone traffic well [14]. If need be, future analysis could take into account other models of traffic [15]. We hypothesize that, in the proposed method, other discrete distributions can be used to model the number of packets, not only the Poisson distribution. In order to model various discrete probability distributions, a probability distribution table could be used.

If we assume that the mean rate of packets per time slot is λ, then the probability that l packets arrive in a time slot is $f(l, \lambda)$ as given by Equ. (1). In the rest of the article, we assume that all mean rates are of the Poisson distribution and they relate to a time slot.

$$f(l, \lambda) = \frac{e^{-\lambda}\lambda^l}{l!} \tag{1}$$

The analysis and simulation are implemented as the OPUS (Optical Packet Switching Unified Solver) package, which is freely available for download at [16]. Its source code can be consulted for implementation details to which we refer in the article.

II. Problem Statement

There are N nodes in the network, each assigned a number from 1 to N, which are connected by fibers. Node i shown in Fig. 1 has output fibers to neighboring backbone nodes, and each fiber (i, j) to node j has $w_{i,j}$ wavelengths and is of length $d_{i,j}$ kilometers. The node's output capacity is $w_i = \sum_{j=1}^{N} w_{i,j}$. The numbers of wavelengths are given by the $N \times N$ wavelength matrix $\mathbf{W} = (w_{i,j})$, while the fiber lengths are given by the $N \times N$ distance matrix $\mathbf{D} = (d_{i,j})$.

The network is loaded with flows (i, n) from node i to node n. Flow (i, n) generates packets whose mean rate for every time slot is $\beta_{i,n}$. The $N \times N$ traffic matrix $\mathbf{B} = (\beta_{i,n})$ gives the mean rates for all flows.

The network works synchronously; and so its nodes receive and send packets at the beginning of time slots. The node delivers packets from the backbone network to the access network through the local drop fibers, and admits packets from the access network to the backbone network through the local add fibers.

Figure 2 presents how packets travel at a node and with what mean rates. Node i receives packets from its neighbor nodes at mean rate $\alpha_i = \sum_{n=1}^{N} \alpha_{i,n}$, where $\alpha_{i,n}$ packets are destined to node n (i.e., node n is their destination), of which $\alpha_{i,i}$ packets go to the local drop. Mean rates $\alpha_{i,n}$ are grouped into $N \times N$ matrix $\mathbf{A} = (\alpha_{i,n})$. There are $\alpha'_{i,n}$ packets *in transit* that go through node i to destination node n, where $\alpha'_{i,n} = \alpha_{i,n}$ for $i \neq n$, and $\alpha'_{i,i} = 0$ for any i. Therefore, the mean rate of all packets in transit at node i is $\alpha'_i = \sum_{n=1}^{N} \alpha'_{i,n}$. Mean rates $\alpha'_{i,n}$ are grouped into $N \times N$ matrix $\mathbf{A}' = (\alpha'_{i,n})$. Moreover, packets from the access network destined to node n arrive at node i at the mean rates $\beta_{i,n}$, totalling $\beta_i = \sum_{n=1}^{N} \beta_{i,n}$ packets that ask for admission. Since node i is not going to send packets to itself, then $\beta_{i,i} = 0$. Packets are admitted at the mean rate $\beta'_i = \sum_{n=1}^{N} \beta'_{i,n}$, and then together with the packets in transit α'_i, they are routed. Mean rates $\beta'_{i,n}$ are grouped into $N \times N$ matrix $\mathbf{B}' = (\beta'_{i,n})$. The routing algorithm accepts packets at mean rates $\mu_{i,n} = \alpha'_{i,n} + \beta'_{i,n}$ totalling $\mu_i = \sum_{n=1}^{N} \mu_{i,n} = \alpha'_i + \beta'_i$, and sends them out at the mean rate $\mu'_i = \sum_{n=1}^{N} \mu'_{i,n}$, where $\mu'_{i,n}$ is the mean rate of packets at node i that are destined to node n and that have already been sent out of the node. Mean rates $\mu_{i,n}$ are grouped into $N \times N$ matrix $\mathbf{U} = (\mu_{i,n})$. Matrix $\mathbf{U}$ is easily obtained: $\mathbf{U} = \mathbf{A}' + \mathbf{B}'$.

Our objective in admission control is to admit as many packets as possible, but not more than the node can send out. Therefore, in every time slot, we admit as many packets as there are output slots left, which equals to the difference between the number of available output slots and the number of packets in transit. The admitted packets are chosen from the packets requesting admission at random to facilitate fairness.

Apart from sending packets to their destinations, the routing algorithm should also optimize the network performance, so we use packet classes and distances. We first route packets of the better classes (first class is the best) before worse classes

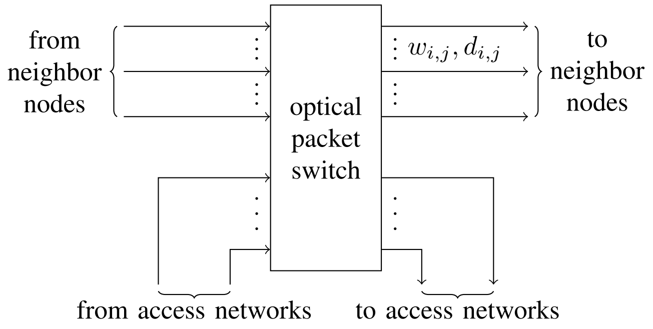


Fig. 1: Inputs and outputs of node i.

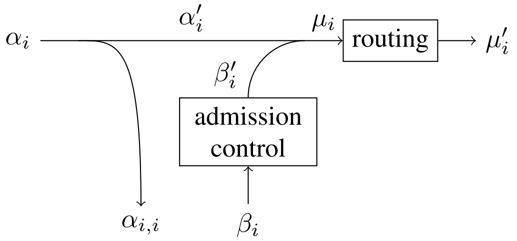


Fig. 2: Mean rates of packets at node i.

(second, third, and so on). Packets can be assigned different classes at each node according to any policy, just as DiffServ allows, and we use the following policy. At node i, each packet is assigned an integer class number from 1 up to the number of output fibers, where the class number equals to the number of preferred output fibers. We define the *preferred output fiber* of a packet as the output fiber that yields the distance to the packet's destination at most twice as high as the shortest distance to the packet's destination. If there are two or more packets of the same class, then the one with the shortest distance to its destination is routed first. The routing algorithm accommodates as many packets into output fibers as possible by converting wavelengths and resorts to deflection routing only when no wavelengths are left on preferred output fibers.

We introduce two integer parameters: η is the maximal number of hops a packet can make (i.e., transitions between two nodes), and δ is the maximal distance a packet can travel. A packet has a hop counter and a distance counter, which are inspected when a packet arrives at a node. A packet is removed if either of its counters exceeds the limit. The η parameter can model the time-to-live field of IPv4 packets or the hop limit field of IPv6 packets, and δ is the distance after which the packet's payload is still usable despite incurred optical distortions.

In the proposed method, we introduced two major approximations. First, the number of packets of a flow is approximated by the Poisson distribution. According to the Poisson distribution, the number of packets can be any, while in fact it is limited by the fiber capacity. Moreover, the assumption that a flow is of the Poisson distribution implies that it is independent of other flows, while in fact flows can have correlated distributions as their packets contend at nodes. Even a single flow can be of an autocorrelated distribution as its packets can revisit a node due to deflections. Second, in the analysis of a node, we approximate the probabilities of admission and routing based on the most probable arrangements while disregarding less probable arrangements.

The result of the analysis are the paths and mean rates of packets that belong to a given flow.

III. Analysis

The analysis is composed of several building blocks, which are described separately in the following subsections. We start with the polynomial notation in Section III-A. The algorithm for finding the most probable arrangements is presented in Section III-B, which is subsequently used in Section III-C for the node evaluation. In Section III-D, we present the analysis of a flow. Finally, in Section III-E, the whole network is evaluated.

A. Polynomial Notation

Packets can take different routes or even revisit nodes due to deflections. Packets of a single flow can get to a given node after having traveled only 100 km, while other packets of the same flow can get there after 250 km. Moreover, the mean rates of packets can differ depending on the distances the packets travel. To facilitate the description of packet rates and distances, we present a polynomial approach, similar to the one presented in [10], though there the polynomials keep track of the number of deflections, not the distance as in this article.

We are interested in the mean rates λ_d of packets that traveled exactly distance d kilometers, where d is an integer. If we need to distinguish distances smaller than a kilometer, the distances d could be expressed in meters. The mean rates are organized in a polynomial in variable x, as given by Equ. (2). We use such polynomials to describe packets that visit a node or traverse a fiber. The polynomial variable x and its terms x^d help in organizing mean rates λ_d into groups.

$$p(x) = \sum_{d=0}^{\delta} \lambda_d x^d \qquad (2)$$

For example, the polynomial $0.01304x^{460} + 1.25086x^{380}$ provides information that on average 0.01304 packets have traveled 460 km, and 1.25086 packets have traveled 380 km.

While polynomials describe the packets of a flow, what happens to the packets is modelled by two operations on polynomials: addition and multiplication. When packets of a flow arrive at a node along two or more fibers, we add the polynomials for the fibers in order to get the polynomial that describes the packets at a node.

We multiply polynomials by term φx^d to model the transition of packets along a fiber of length d, provided packets were sent to that fiber with probability φ. The probability φ of sending the packet is calculated in Section III-C.

In order to model the requirement that packets be removed from the network when they travel more than δ kilometers, we discard the polynomial terms with the power larger than δ. To see this constraint in action, the multiplication $0.2x^{100}(0.1x^{950} + 0.2x^{800})$ for δ = 1000, results in $0.04x^{900}$ as we dropped the term $0.02x^{1050}$.

Our polynomial notation can help track not only various packet paths, but also various delays caused by buffering packets at nodes. However, we only mention this use in passing here, as we do not study nodes with buffers.

The polynomials are not indispensable, and we could replace them by vectors of size $\delta + 1$ with mean packet rates as elements. Adding vectors would have the same interpretation as adding polynomials, while the multiplication by term φx^d would be implemented as shifting elements in a vector by d positions and multiplying them by φ. We simply needed a tool for differentiating between packets of different history, and found the polynomial notation handy and concise.

B. Algorithm for Finding Most Probable Packet Arrangements

A *packet arrangement* is a set of packets arranged in R groups that is considered for admission or routing. We describe a packet arrangement by vector $\mathbf{x} = (x_1, \ldots, x_r, \ldots, x_R)$, where x_r is the number of packets that belong to group r. We can group packets in any way, and, for example, later in the analysis of admission, we have two groups of packets: one group includes the packets in transit, while the other the packets asking for admission.

Packet arrangement $\mathbf{x}$ describes an outcome of a random process in which packets of R groups arrive at random. The number of packets in group r is a random variable of the Poisson distribution with the mean rate λ_r given by vector $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_r, \ldots, \lambda_R)$. The probability of packet arrangement $\mathbf{x}$ is given by Equ. (3).

$$P(\mathbf{x}) = \prod_{r=1}^{R} f(x_r, \lambda_r) \tag{3}$$

Since the number of packets in a group can be infinite, the number of arrangements is also infinite. However, there is no need to consider too many arrangements, because most of them are very improbable. For example, if $\mathbf{\Lambda} = (0.75, 0.5)$, then the probability of the arrangement $\mathbf{x} = (15, 10)$ is $P(\mathbf{x}) = f(15, 0.75)f(10, 0.5) \approx 8 \times 10^{-25}$, and so there is little need to consider this arrangement.

Given vector $\mathbf{\Lambda}$, the objective of the algorithm is to find the sequence of the most probable arrangements. The analysis of both admission and routing iterates over packet arrangements, and our algorithm ensures that the most important arrangements are considered and the unlikely ones discarded. We find the most probable arrangement denoted by $\mathbf{x_1} = (x_{1,1}, \ldots, x_{r,1}, \ldots, x_{R,1})$, and then continue finding the next most probable arrangements $\mathbf{x_e} = (x_{1,e}, \ldots, x_{r,e}, \ldots, x_{R,e})$, so that probability $P(\mathbf{x_e})$ does not increase as e increases. The last considered arrangement is of number $\Xi(\mathbf{\Lambda})$ given by Equ. (4), where the precision parameter ξ helps us influence the precision of the analysis: the smaller the value of ξ, the larger the number of considered packet arrangements.

$$\Xi(\mathbf{\Lambda}) = max\{e \in \mathbb{N} | \frac{P(\mathbf{x_e})}{P(\mathbf{x_1})} > \xi\} \tag{4}$$

In the course of selecting the most probable arrangements, we will be using the function[1] $\Gamma(y, \lambda)$ that returns the y-th most probable value of the random variable of the Poisson distribution with the mean rate λ. Therefore, $\Gamma(1, \lambda)$ is the mode, and the inequality Equ. (5) is satisfied.

$$f(\Gamma(y, \lambda), \lambda) \geq f(\Gamma(y + 1, \lambda), \lambda) \text{ for } y = 1, 2, \ldots \tag{5}$$

Aside from describing arrangement e with $\mathbf{x_e}$, we introduce vector $\mathbf{y_e} = (y_{1,e}, \ldots, y_{r,e}, \ldots, y_{R,e})$, where $y_{r,e}$ is the index of the consecutive most probable number of packets in group r. Having an arrangement described by vector $\mathbf{y_e}$, we can get vector $\mathbf{x_e} = (\Gamma(y_{1,e}, \lambda_1), \ldots, \Gamma(y_{r,e}, \lambda_r), \ldots, \Gamma(y_{R,e}, \lambda_R))$.

In the search for the sequence of the most probable arrangements, we employ the priority queue of vectors $\mathbf{y_e}$, which places at the front the vector describing the most probable arrangement. At the start of the search, we initialize the queue with vector $\mathbf{y_1} = (1, \ldots, 1)$, which is the most probable, because it describes an arrangement with modes only.

Finding the next most probable arrangement involves two steps. First, popping the vector from the front of the queue gives us the next most probable arrangement $\mathbf{y_e}$. Second, we insert into the queue exactly R vectors which differ from $\mathbf{y_e}$ only in one incremented element: $(y_{1,e} + 1, y_{2,e}, \ldots, y_{R,e})$, $(y_{1,e}, y_{2,e} + 1, \ldots, y_{R,e})$, $\ldots$, $(y_{1,e}, y_{2,e}, \ldots, y_{R,e} + 1)$.

e	$\mathbf{x_e}$	$\mathbf{y_e}$	Queue of vectors $\mathbf{y}$	$P(\mathbf{x_e})$
1	$(3, 10)$	$(1, 1)$	$(1, 1)$	0.02675
2	$(3, 11)$	$(1, 2)$	$(1, 2), (2, 1)$	0.02675
3	$(2, 10)$	$(2, 1)$	$(2, 1), (2, 2), (1, 3)$	0.02675
4	$(2, 11)$	$(2, 2)$	$(2, 2), (1, 3), (3, 1)$	0.02675
5	$(3, 12)$	$(1, 3)$	$(1, 3), (2, 3), (3, 1), (3, 2)$	0.02452
6	$(2, 12)$	$(2, 3)$	$(2, 3), (1, 4), (3, 1), (3, 2)$	0.02452
7	$(3, 9)$	$(1, 4)$	$(1, 4), (2, 4), (3, 1), (3, 2), (3, 3)$	0.02431

TABLE I: Seven most probable arrangements for $\mathbf{\Lambda} = (3, 11)$.

The first popped vector is always $\mathbf{y_1} = (1, 1, ..., 1)$, the second vector popped $\mathbf{y_2}$ (i.e., the second most probable arrangement) is one of $(2, 1, ..., 1)$, $(1, 2, ..., 1)$, $\ldots$, $(1, 1, ..., 2)$ depending on which one of these vectors describes a more probable arrangement. This process is repeated $\Xi(\mathbf{\Lambda})$ times to get $\Xi(\mathbf{\Lambda})$ vectors $\mathbf{y_e}$ describing the most probable arrangements.

Example[2]. We are given $\mathbf{\Lambda} = (3, 11)$ and $\xi = 10^{-2}$. The most probable arrangement is $\mathbf{x_1} = (3, 10)$ with probability $P(\mathbf{x_1}) = 0.02675$, and the last arrangement considered is the 157-th most probable one: $\mathbf{x_{157}} = (2, 22)$ with probability $P(\mathbf{x_{157}}) = 0.00027$, since $\Xi(\mathbf{\Lambda}) = 157$. The seven most probable arrangements are given in Table I, where each row gives information for one arrangement e described both with the $\mathbf{x_e}$ and $\mathbf{y_e}$ vectors, the content of the queue before popping arrangement e and replenishing the queue, and the probability $P(\mathbf{x_e})$ of arrangement e.

C. Node Evaluation

The objective of the node evaluation is to compute for a single node i the mean rates of admitting new packets and the probabilities of sending packets along output fibers. The input data are the mean rates of packets in transit, mean rates of packets requesting admission, and how the algorithms of admission and routing work. In Section III-E, the node evaluation is used repeatedly for all nodes.

1) Analysis of Admission Control: We seek the mean rates $\beta'_{i,n}$ of packets admitted at node i which are destined to node n. As the input data, we are given the mean arrival rates $\beta_{i,n}$ of packets that ask for admission, mean rate α'_i of packets in transit, and the output capacity w_i. We calculate the value of α'_i in Section III-E, where we study the whole network.

We use the algorithm presented in the previous section to get the sequence of the most probable arrangements. An *admission packet arrangement* number e for node i is $\mathbf{x_{adm,i,e}} = (a_{i,e}, b_{i,e})$, where $a_{i,e}$ is the number of packets in transit, and $b_{i,e}$ is the number of packets asking for admission. The mean rates for the two groups are given by $\mathbf{\Lambda_{adm,i}} = (\alpha'_i, \beta_i)$. For arrangement $\mathbf{x_{adm,i,e}}$, the remaining output capacity equals $g_{i,e} = \max(w_i - a_{i,e}, 0)$ and the number of admitted packets equals $b'_{i,e} = \min(g_{i,e}, b_{i,e})$.

We derive the mean admission probability ρ_i which applies to all packets regardless of their destination as we admit packets fairly. There is some number Z of packets that can

[1] Implemented in OPUS as function poisson::get_ith_max.

[2] Implemented in OPUS as test_arr_queue.cc.

arrive at the node and ask for admission. We randomly select one of these packets and denote it z. The probability ρ_i is conditional: it is the probability that packet z is admitted provided it is actually requesting admission. It is given by Equ. (6), where A_i is the event that packet z is admitted, and B_i is the event that packet z is asking for admission.

$$\rho_i = P(A_i|B_i) = \frac{P(A_i \cap B_i)}{P(B_i)} \tag{6}$$

$P(A_i \cap B_i)$ is the probability that packet z is requesting admission and is admitted as given by Equ. (7), where $B_{i,e}$ is the event that arrangement e takes place with packet z, and $D_{i,e}$ is the event that packet z is admitted in arrangement e.

$$P(A_i \cap B_i) = \sum_{e=1}^{\Xi(\mathbf{\Lambda_{adm,i}})} P(B_{i,e})P(D_{i,e}) \tag{7}$$

The probability of event $B_{i,e}$ is given by Equ. (8), where $E_{i,e}$ is the event that packet z is part of arrangement e.

$$P(B_{i,e}) = P(\mathbf{x_{adm,i,e}})P(E_{i,e}) \tag{8}$$

The probability of event $D_{i,e}$ is given by Equ. (9), where $C^{b_{i,e}}_{b'_{i,e}}$ is the number of combinations of size $b'_{i,e}$ from a set of size $b_{i,e}$.

$$P(D_{i,e}) = \frac{C^{b_{i,e}-1}_{b'_{i,e}-1}}{C^{b_{i,e}}_{b'_{i,e}}} = \frac{b'_{i,e}}{b_{i,e}} \tag{9}$$

The probability of event $E_{i,e}$ is the ratio of the number of combinations with packet z to the number of all combinations, as given by Equ. (10).

$$P(E_{i,e}) = \frac{C^{Z-1}_{b_{i,e}-1}}{C^{Z}_{b_{i,e}}} = \frac{b_{i,e}}{Z} \tag{10}$$

The probability of event B_i is given by Equ. (11).

$$P(B_i) = \sum_{e=1}^{\Xi(\Lambda_{adm,i})} P(B_{i,e}) \tag{11}$$

Finally, we put together Equ. (6)-Equ. (11) to get Equ. (12), the complete equation for ρ_i which does not depend on Z.

$$\rho_i = \frac{\sum_{e=1}^{\Xi(\mathbf{\Lambda_{adm,i}})} P(\mathbf{x_{adm,i,e}})b'_{i,e}}{\sum_{e=1}^{\Xi(\mathbf{\Lambda_{adm,i}})} P(\mathbf{x_{adm,i,e}})b_{i,e}} \tag{12}$$

Having ρ_i and $\beta_{i,n}$, we calculate the mean admission rates $\beta'_{i,n}$ with Equ. (13).

$$\beta'_{i,n} = \rho_i\beta_{i,n} \tag{13}$$

2) Analysis of Routing: In this subsection, we derive for node i the mean routing probabilities $\varphi_{i,j,n}$ of sending through output fibers (i, j) packets destined to node n, given that these packets are already at node i. The data for the analysis are the mean rates $\mu_{i,n}$ of packets asking for routing at node i that are destined to node n. Mean rates $\mu_{i,n}$ relate not only to packets in transit, but also to packets that have been admitted at node i.

The mean probabilities $\varphi_{i,j,n}$ are used in Section III-E for computing the mean rates at which packets leave nodes along fibers (i, j). Using these probabilities, we can also compute the mean rate $\mu'_i = \sum_{n=1}^{N} \mu'_{i,n}$ of packets sent out of node i, where $\mu'_{i,n} = \mu_{i,n} \sum_{j=1}^{N} \varphi_{i,j,n}$. The probability of dropping a packet destined to node n equals $1 - \sum_{j=1}^{N} \varphi_{i,j,n}$.

We employ the same approach as in the analysis of admission control: we enumerate the most probable arrangements of packets requesting routing; then, for each arrangement, we calculate routing probabilities and average them over the considered arrangements to get $\varphi_{i,j,n}$.

A *routing packet arrangement* number e for node i is $\mathbf{x_{rou,i,e}} = (x_{i,1,e}, \ldots, x_{i,n,e}, \ldots, x_{i,N,e})$, where $x_{i,n,e}$ is the number of packets asking routing at i destined to node n of arrangement e. The mean rates for packet groups are given by $\mathbf{\Lambda_{rou,i}} = (\mu_{i,1}, \ldots, \mu_{i,N})$.

In the network, there is some number Z_n of packets that travel to node n, and one of those packets we choose at random and denote z_n. The mean probability $\varphi_{i,j,n}$ is conditional as expressed by Equ. (14), where $H_{i,j,n}$ is the event that packet z_n is sent from node i to node j, and $L_{i,n}$ is the event that packet z_n is at node i.

$$\varphi_{i,j,n} = P(H_{i,j,n}|L_{i,n}) = \frac{P(H_{i,j,n} \cap L_{i,n})}{P(L_{i,n})} \tag{14}$$

The probability that packet z_n is present at node i and is sent to node j is the sum over considered arrangements as given by Equ. (15), where $\varphi_{i,e,j,n}$ is the probability that packet z_n is sent to node j provided it is at node i as part of arrangement e. We derive $\varphi_{i,e,j,n}$ later in this subsection.

$$P(H_{i,j,n} \cap L_{i,n}) = \sum_{e=1}^{\Xi(\mathbf{\Lambda_{rou,i}})} P(L_{i,n,e})\varphi_{i,e,j,n} \tag{15}$$

$L_{i,n,e}$ is the event that packet z_n is present at node i as part of arrangement e. The probability of this event is given by Equ. (16), where $R_{i,n,e}$ is the event that packet z_n is part of arrangement e.

$$P(L_{i,n,e}) = P(\mathbf{x_{rou,i,e}})P(R_{i,n,e}) \tag{16}$$

The probability of event $R_{i,n,e}$ is given by Equ. (17), where $C^{Z_n}_{x_{i,n,e}}$ is the number of combinations of size $x_{i,n,e}$ from a set of size Z_n. Note that, if arrangement e has no packets destined to node n, i.e., $x_{i,n,e}$ is zero, then the probability $P(R_{i,n,e})$ is zero, and so in Equ. (15) we are allowed to sum over all arrangements regardless of whether they have packet z_n.

$$P(R_{i,n,e}) = \frac{C^{Z_n-1}_{x_{i,n,e}-1}}{C^{Z_n}_{x_{i,n,e}}} = \frac{x_{i,n,e}}{Z_n} \tag{17}$$

The final expression for the numerator of Equ. (14) is given by Equ. (18), which is derived from Equ. (15), Equ. (16), and Equ. (17).

$$P(H_{i,j,n} \cap L_{i,n}) = \frac{1}{Z_n} \sum_{e=1}^{\Xi(\mathbf{\Lambda_{rou,i}})} P(\mathbf{x_{rou,i,e}}) x_{i,n,e} \varphi_{i,e,j,n} \tag{18}$$

Now, we continue to derive the denominator of Equ. (14). Event $L_{i,n}$ consists of mutually exclusive events $L_{i,n,e}$, and so the probability of $L_{i,n}$ is the sum of probabilities of events $L_{i,n,e}$ as given by Equ. (19).

$$P(L_{i,n}) = \sum_{e=1}^{\Xi(\mathbf{\Lambda_{rou,i}})} P(L_{i,n,e}) \tag{19}$$

It follows from Equ. (16), Equ. (17), and Equ. (19) that the probability $P(L_{i,n})$ is given by Equ. (20).

$$P(L_{i,n}) = \frac{1}{Z_n} \sum_{e=1}^{\Xi(\mathbf{\Lambda_{rou,i}})} P(\mathbf{x_{rou,i,e}}) x_{i,n,e} \tag{20}$$

Finally, following Equ. (14), Equ. (18), and Equ. (20), the complete equation for probability $\varphi_{i,j,n}$ is given by Equ. (21), which is independent on how the routing algorithm works.

$$\varphi_{i,j,n} = \frac{\sum_{e=1}^{\Xi(\mathbf{\Lambda_{rou,i}})} P(\mathbf{x_{rou,i,e}}) x_{i,n,e} \varphi_{i,e,j,n}}{\sum_{e=1}^{\Xi(\mathbf{\Lambda_{rou,i}})} P(\mathbf{x_{rou,i,e}}) x_{i,n,e}} \tag{21}$$

Now, we derive the probability $\varphi_{i,e,j,n}$ present in Equ. (21), which was introduced in Equ. (15). Packet z_n is one of the $x_{i,n,e}$ packets destined to node n of arrangement $\mathbf{x_{rou,i,e}}$. The routing algorithm sends to neighbor node j a number of those packets denoted by $x'_{i,e,j,n}$. Therefore, the probability that packet z_n is sent to neighbor node j provided it is at node i as part of arrangement e is a ratio of $x'_{i,e,j,n}$ to $x_{i,n,e}$ as given by Equ. (22).

$$\varphi_{i,e,j,n} = \frac{x'_{i,e,j,n}}{x_{i,n,e}} \tag{22}$$

Algorithm 1 computes numbers $x'_{i,e,j,n}$ and groups them into matrix $\mathbf{x'_{i,e}} = (x'_{i,e,j,n})$, where j and n are the matrix indices. This matrix if of dimensions $N \times N$ as there are at most N^2 numbers $x'_{i,e,j,n}$ for arrangement $\mathbf{x_{rou,i,e}}$.

The algorithm uses three functions: pop($\mathbf{x}$), output($p, \mathbf{W'}$), and dest(p), which implement the routing algorithm. The pop($\mathbf{x}$) function returns the first packet to route in arrangement $\mathbf{x}$ according to the rules described in Section II, and removes the packet from the arrangement. The output($p, \mathbf{W'}$) function returns node j where packet p should be sent, which is the node to which a preferred output fiber leads or the node where the packet is deflected. The function returns 0 if no output fiber is available, which results in dropping packet p. During its iterations, the algorithm keeps track of the number of wavelengths left on output fibers (i, j) with matrix $\mathbf{W'} = (w'_{i,j})$. The dest(p) function returns the destination node of packet p.

The analysis of admission and routing was developed for specific algorithms, but using a similar approach, we can develop analysis for other algorithms. We use the deflection routing, but its use is not required. Since the analysis applies to nodes with a large degree and a large number of wavelengths, it can be used in analyzing real-world nodes.

Algorithm 1 Given i, e, $\mathbf{W}$ and $\mathbf{x_{rou,i,e}}$, calculate $\mathbf{x'_{i,e}}$.

```
$\mathbf{x} \leftarrow \mathbf{x_{rou,i,e}}$
$\mathbf{W'} = (w'_{i,j}) \leftarrow \mathbf{W}$
$\mathbf{x'_{i,e}} = (x'_{i,e,j,n}) \leftarrow 0$
while $\mathbf{x}$ is not null do
  $p \leftarrow$ pop($\mathbf{x}$)
  $j \leftarrow$ output($p, \mathbf{W'}$)
  if $j \neq 0$ then
    $n \leftarrow$ dest($p$)
    $x'_{i,e,j,n} \leftarrow x'_{i,e,j,n} + 1$
    $w'_{i,j} \leftarrow w'_{i,j} - 1$
  end if
end while
return $\mathbf{x'_{i,e}}$
```

D. Analysis of a Flow

We introduce the *visit vector* $\mathbf{P_{a,b,v}}$ that is a row vector of size N given by Equ. (23), where polynomial $p_{a,b,v,n}(x)$ describes the visits that packets of flow (a, b) pay to node n during their visit number v.

$$\mathbf{P_{a,b,v}} = [p_{a,b,v,1}(x), \ldots, p_{a,b,v,n}(x), \ldots, p_{a,b,v,N}(x)] \tag{23}$$

We start with vector $\mathbf{P_{a,b,1}}$, which describes where packets reside when they are admitted to the network. Since the packets of flow (a, b) start at node a, then all elements of $\mathbf{P_{a,b,1}}$ equal 0 except element a, which is $p_{a,b,1,a}(x) = \beta'_{a,b}$, i.e., the mean rate of admitted packets.

We get the subsequent visit vectors $\mathbf{P_{a,b,v}}$ only for $v \leq \eta + 1$, because packets cannot hop more than η times, and so the visit vectors for $v > \eta + 1$ are null. We get the visit vectors $\mathbf{P_{a,b,v}}$ by multiplying the already-known $\mathbf{P_{a,b,v-1}}$ by the transition matrix $\mathbf{T_b}$ as given by Equ. (24).

$$\mathbf{P_{a,b,v}} = \mathbf{P_{a,b,v-1}} \mathbf{T_b} \text{ for } v > 1 \tag{24}$$

The $N \times N$ transition matrix[3] describes packet transitions along fibers (but it is not a Markov transition matrix). Flows that have the same destination node b, i.e., flows (a, b) for any source node a, have their own transition matrix $\mathbf{T_b} = (t_{i,j,b})$, because according to our routing algorithm packets of the same destination take the same routes. The elements $t_{i,j,b}$ are given by Equ. (25), where $\varphi_{i,j,b}$ is the probability (derived in Section III-C) that a packet destined to node b traverses fiber (i, j) of length $d_{i,j}$, provided the packet is at node i. Elements $t_{i,j,b}$ are zero (regardless of the destination node b) if there is no fiber between nodes i and j, which means that the number of nonzero elements equals to at most twice the number of fibers in the network. We have N transition matrices $\mathbf{T_b}$, one for each destination node b.

[3] Implemented in OPUS as type trans_matrix.

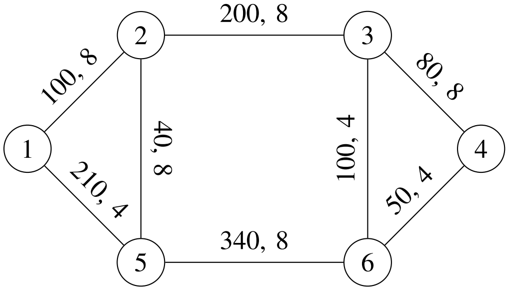


Fig. 3: Sample network, where the first number of the fiber label gives the fiber length, while the second the number of wavelengths.

Visit number	Node	Polynomial
1	5	$p_{5,1,1,5}(x) = 0.2$
2	1 2	$p_{5,1,2,1}(x) = 0.01x^{210}$ $p_{5,1,2,2}(x) = 0.19x^{40}$
3	1	$p_{5,1,3,1}(x) = 0.19x^{140}$

TABLE II: Description of packet visits for flow (5, 1).

$$t_{i,j,b} = \varphi_{i,j,b} \cdot x^{d_{i,j}} \tag{25}$$

While the visit vectors describe which nodes packets visit, they do not give information on the fibers used. Therefore, we introduce *hop matrices* $\mathbf{Q_{a,b,h}} = (q_{a,b,h,i,j}(x))$, where polynomials $q_{a,b,h,i,j}(x)$ describe packets of flow (a, b) that in their hop number h traverse fiber (i, j). Polynomial $q_{a,b,h,i,j}(x)$ is obtained by multiplying the polynomial $t_{i,j,b}$ that expresses the transition along fiber (i, j) by the polynomial that describes the packets residing at node i as given by Equ. (26).

$$q_{a,b,h,i,j}(x) = t_{i,j,b} \cdot p_{a,b,h,i}(x) \tag{26}$$

Example. A sample transition matrix $\mathbf{T_1}$ for packets traveling to node 1 in the network in Fig. 3 is given by Equ. (27).

$$\mathbf{T_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ x^{100} & 0 & 0 & 0 & 0 & 0 \\ 0 & x^{200} & 0 & 0 & 0 & 0 \\ 0 & 0 & x^{80} & 0 & 0 & 0 \\ 0.05x^{210} & 0.95x^{40} & 0 & 0 & 0 & 0 \\ 0 & 0 & x^{100} & 0 & 0 & 0 \end{bmatrix} \tag{27}$$

Packets of flow (5, 1) are admitted with the mean rate 0.2, and so $\mathbf{P_{5,1,1}} = [0, 0, 0, 0, 0.2, 0]$. Having $\mathbf{T_1}$, we can obtain the subsequent visit vectors $\mathbf{P_{5,1,2}} = \mathbf{P_{5,1,1}T_1} = [0.01x^{210}, 0.19x^{40}, 0, 0, 0, 0]$, $\mathbf{P_{5,1,3}} = \mathbf{P_{5,1,2}T_1} = [0.19x^{140}, 0, 0, 0, 0, 0]$. The very same information is presented in a more readable format in Table II.

Polynomials $q_{a,b,h,i,j}(x)$ are given in Table III. We can see that packets during the first hop traverse fiber (5, 1) at mean rate 0.01 and fiber (5, 2) at mean rate 0.19. Most of the packets travel along fiber (2, 1) with mean rate 0.19 making in total 140 km.

Hop number	Fiber	Polynomial
1	(5, 1) (5, 2)	$q_{5,1,1,5,1}(x) = 0.01x^{210}$ $q_{5,1,1,5,2}(x) = 0.19x^{40}$
2	(2, 1)	$q_{5,1,2,2,1}(x) = 0.19x^{140}$

TABLE III: Description of packet hops for flow (5, 1).

E. Network Analysis

In this subsection, we put together the previously-presented building blocks in order to analyze the network[4]. The network analysis is given the traffic matrix $\mathbf{B}$, the distance matrix $\mathbf{D}$, and the wavelength matrix $\mathbf{W}$. The solution is composed of visit vectors and hop matrices, based on which we can derive other metrics of interest such as fiber utilization. To simplify the description of the network analysis, we define $\mathbf{\Phi}$ as the set of all $\varphi_{i,j,n}$, $\mathbf{P}$ as the set of all $\mathbf{P_{a,b,v}}$, $\mathbf{T}$ as the set of all $\mathbf{T_b}$, and $\mathbf{Q}$ as the set of all $\mathbf{Q_{a,b,h}}$.

In order to calculate hop matrices $\mathbf{Q}$ and visit vectors $\mathbf{P}$, we need matrix $\mathbf{B'}$ with the mean rates of admitted packets, and the routing probabilities $\mathbf{\Phi}$, which in turn depend on matrix $\mathbf{A}$ with the mean rates of packets arriving from neighbors. The problem is that, for calculating matrix $\mathbf{A}$, we need hop matrices $\mathbf{Q}$.

We solve the problem of coupled $\mathbf{Q}$ and $\mathbf{A}$ by iterations. There are S iterations, and in every iteration s, we obtain visit vectors and hop vectors based on the results from the preceding iterations. We furnish the variables, vectors, and matrices with index s to denote that they relate to iteration s. For instance, $\mathbf{Q^{(s)}}$ are the hop matrices in iteration s.

We use hop matrices $\mathbf{Q^{(s)}}$ from iteration s to calculate matrix $\mathbf{A}$ that is used as input data in the succeeding iterations. The elements of matrix $\mathbf{A}$ are calculated with Equ. (28), where the values of polynomials are obtained for $x = 1$ to get mean rates regardless of the distance packets travelled. We denote this operation as $\mathbf{A}(\mathbf{Q^{(s)}})$. This notation, where the data of the type given before the parenthesis is calculated based on the data in the parenthesis, is used for similar operations.

$$\alpha_{i,n} = \sum_{a=1}^{N} \sum_{h=1}^{\eta} \sum_{j=1}^{N} q_{a,n,h,j,i}(x=1)^{(s)} \tag{28}$$

Matrices $\mathbf{B'^{(s)}}$ and $\mathbf{U^{(s)}}$ are the input data for iteration s, and they are an average of matrices calculated based on $K = \min(\kappa, s)$ preceding hop matrices $\mathbf{Q^{s-1}}, \ldots, \mathbf{Q^{s-K}}$. The parameter κ is the maximal number of preceding hop matrices used for calculation, if they exist. The averaging process helped us stabilize the calculation of mean rates of admitted packets, because the method presented in [17] of scaling the traffic matrix yielded unsatisfactory results.

In iteration s, we proceed as follows. First, matrices $\mathbf{B'^{(s)}}$ and $\mathbf{U^{(s)}}$ are calculated. Matrix $\mathbf{B'^{(s)}}$ is obtained as an average of matrices $\mathbf{B'}(\mathbf{A'}(\mathbf{A}(\mathbf{Q^{(s-k)}})), \mathbf{B}, \mathbf{W})$ calculated for varying $\mathbf{A'}(\mathbf{A}(\mathbf{Q^{(s-k)}}))$ for $k = 1, \ldots, K$. Matrix $\mathbf{U^{(s)}}$ is the sum of $\mathbf{B'^{(s)}}$ and the average of matrices $\mathbf{A'}(\mathbf{A}(\mathbf{Q^{(s-k)}}))$ calculated for $k = 1, \ldots, K$. Matrix $\mathbf{B'}$ is calculated as described in Section III-C1, while $\mathbf{A'}$ as in Section II.

[4] Implemented in OPUS as function ana_solution.

Algorithm 2 Given $\mathbf{B}$, $\mathbf{D}$, $\mathbf{W}$, calculate $\mathbf{P^{(S)}}$ and $\mathbf{Q^{(S)}}$.

```
$\mathbf{Q^{(0)}} \leftarrow 0$
for $s \leftarrow 1$ to $S$ do
    $K \leftarrow \min(\kappa, s)$
    $\mathbf{B'^{(s)}} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \mathbf{B'}(\mathbf{A'}(\mathbf{A}(\mathbf{Q^{(s-k)}})), \mathbf{B}, \mathbf{W})$
    $\mathbf{U^{(s)}} \leftarrow \mathbf{B'^{(s)}} + \frac{1}{K}\sum_{k=1}^{K} \mathbf{A'}(\mathbf{A}(\mathbf{Q^{(s-k)}}))$
    $\mathbf{\Phi^{(s)}} \leftarrow \mathbf{\Phi}(\mathbf{U^{(s)}}, \mathbf{D}, \mathbf{W})$
    $\mathbf{T^{(s)}} \leftarrow \mathbf{T}(\mathbf{\Phi^{(s)}}, \mathbf{D})$
    $\mathbf{P^{(s)}} \leftarrow \mathbf{P}(\mathbf{T^{(s)}}, \mathbf{B'^{(s)}})$
    $\mathbf{Q^{(s)}} \leftarrow \mathbf{Q}(\mathbf{\Phi^{(s)}}, \mathbf{B'^{(s)}})$
end for
return $\mathbf{P^{(S)}}$, $\mathbf{Q^{(S)}}$
```

Next, matrices $\mathbf{U^{(s)}}$, $\mathbf{D}$, and $\mathbf{W}$ are used as input for evaluating the routing probabilities $\mathbf{\Phi^{(s)}}$ as described in Section III-C2. Using the routing probabilities $\mathbf{\Phi^{(s)}}$ and distance matrix $\mathbf{D}$, we obtain the transition matrices $\mathbf{T^{(s)}}$.

Finally, the visit vectors $\mathbf{P^{(s)}}$ and the hop matrices $\mathbf{Q^{(s)}}$ are obtained using matrix $\mathbf{B'^{(s)}}$ and matrices $\mathbf{T^{(s)}}$ as explained in Section III-D. Algorithm 2 recaps the calculations.

Having hop matrices $\mathbf{Q^{(S)}}$, we can obtain fiber utilization $\psi_{i,j}^{(S)}$ of fiber (i, j), which is the ratio of the mean rate of packets traversing the fiber to the number of wavelengths of that fiber as given by Equ. (29).

$$\psi_{i,j}^{(S)} = \frac{\sum_{a=1}^{N} \sum_{b=1}^{N} \sum_{h=1}^{\eta} q_{a,b,h,i,j}(x=1)^{(S)}}{w_{i,j}} \tag{29}$$

There are three parameters that influence the accuracy of the method: ξ, S, and κ. The ξ parameter influences the number of arrangements considered in the analysis of admission control and routing. The smaller the parameter, the larger the number of considered arrangements. Moreover, the larger the number of nodes in the network, the smaller the value of ξ should be, so that more arrangements are considered in the analysis of routing, where the number of groups equals to the number of network nodes. In our studies, we noticed that the required value of ξ is approximately given by Equ. (30).

$$\xi = \frac{1}{10\log(N)} \tag{30}$$

The required number of iterations S can be as small as 10 for the networks with a few nodes, but it grows for larger networks. Number S is influenced also with the distance limit δ and the hop limit η, because packets can travel in the network longer, causing the solution to converge slower. In our studies, we noticed that number S is most influenced by the number of nodes and equals approximately to that number of nodes.

The κ parameter specifies how many hop matrices of the previous iterations are used in the current iteration, which stabilizes the convergence of the solution. We found that $\kappa = 10$ was enough for the networks that we studied, and for small networks $\kappa = 3$ was enough.

The time complexity of our method is as follows. The operation that dominates is the floating-point multiplication. There are S iterations, and in each iteration there are F flows analyzed. Packets of each flow make at most η hops, and so there are at most η visit vectors and at most η hop matrices calculated in an iteration. The calculation of a visit vector takes at most N^2 multiplications of a polynomial by term φx^d, which takes at most δ floating-point multiplications. Therefore, the time complexity of obtaining visit vectors is $O(SF\eta N^2\delta)$. The calculation of a hop matrix takes at most N^2 multiplications of a polynomial by a real number, which takes at most δ floating-point multiplications. Therefore, the time complexity of obtaining hop matrices is $O(SF\eta N^2\delta)$, and the total time complexity of the method is $O(2SF\eta N^2\delta) = O(SF\eta N^2\delta)$.

The memory complexity of the method is similar. There are at most κ sets of hop matrices $\mathbf{Q}$ stored in memory. For each of the F flows, there are at most η matrices in set $\mathbf{Q}$. A hop matrix has at most N^2 polynomials, each of which has at most δ floating-point numbers. Therefore, the memory complexity is $O(\kappa F\eta N^2\delta)$.

IV. Results

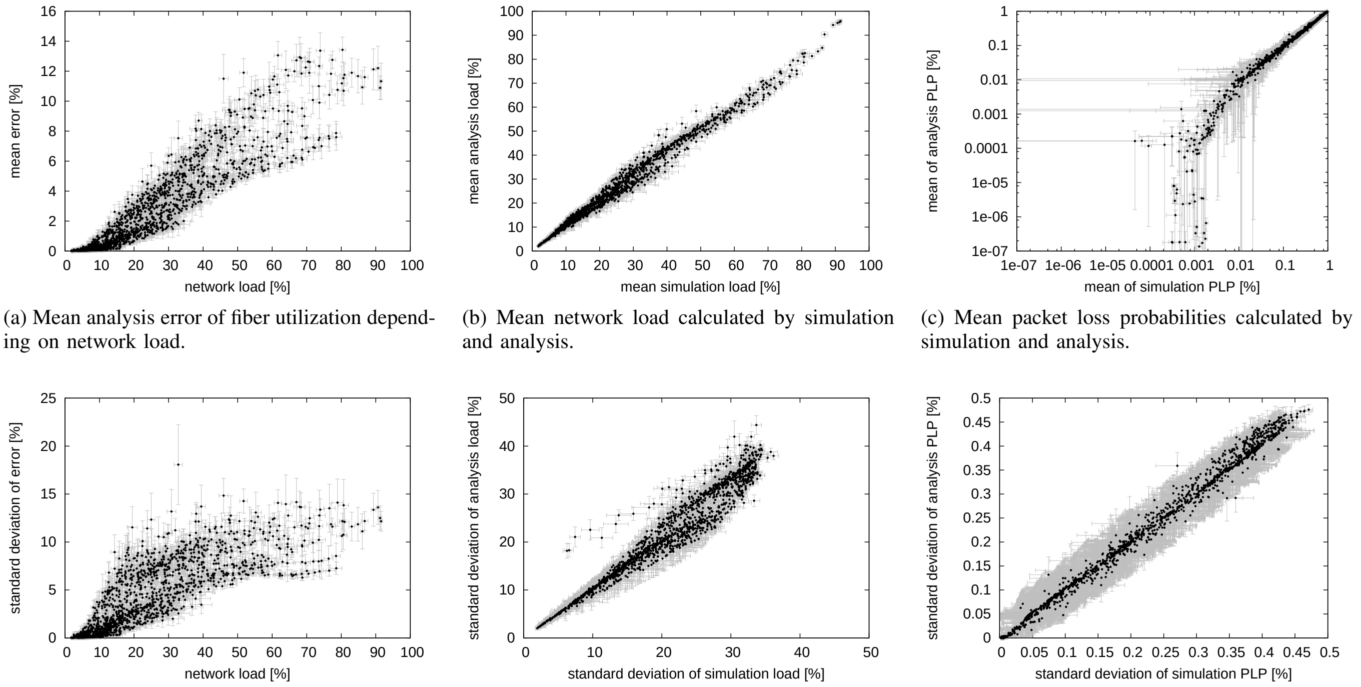
We studied 16000 test cases with random networks and random traffic matrices. A test case consists of a simulation run and an analytical run. Both runs are given the same network and traffic matrix, and return results of the same type. In the runs, we assumed the distance limit $\delta = 1000$ and the hop limit $\eta = 10$ because, for these values, OPS has been demonstrated to work well. In the analysis, we assumed the precision parameter $\xi = 10^{-2}$, while the average limit $\kappa = 10$, because they were enough to yield satisfactory results. We obtained more accurate results with lower values of ξ, and higher values of κ, but the analysis took more time. Using these test cases, we evaluated the accuracy of the results of the presented analytical method by comparing them to the results of simulation.

A random network is created for the given number of nodes and the given number of fibers. Two different nodes are randomly selected as end nodes of a fiber. Each fiber has two attributes: the first is the length expressed in kilometers which is uniformly randomly distributed between 50 and 200, while the other attribute is the number of available wavelengths which is uniformly randomly distributed between 1 and 40.

A random traffic matrix is generated for the given network, the given number of flows, and their given mean rate. There are the given number of flows created, each of the given mean rate, for which two different end nodes are randomly selected. Several flows with the same end nodes are reduced to a single flow with the mean rate being the sum of mean rates of the reduced flows.

The 16000 test cases are grouped into 1600 statistical samples, each with 10 test cases. Each test case in a sample has the same characteristic of the input data (i.e., the same number of nodes, fibers, and flows, and the same flow mean rate), but has a different randomly-generated network and traffic matrix. Based on a sample, we derive statistics for the population of test cases with the same characteristic of the input data as the sample.

Each of the 16000 test cases has different input data. The set with input data is created for the Cartesian product of



(a) Mean analysis error of fiber utilization depending on network load.

(b) Mean network load calculated by simulation and analysis.

(c) Mean packet loss probabilities calculated by simulation and analysis.

(d) Standard deviation of analysis error of fiber utilization depending on network load.

(e) Standard deviation of network load calculated by simulation and analysis.

(f) Standard deviation of packet loss probabilities calculated by simulation and analysis.

Fig. 4: Simulation and analysis results.

the sets of the version numbers $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ (a sample has all the test cases with these version numbers), the number of nodes $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, the number of fibers $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, the number of flows $\{10, 100\}$, and the set of flow mean rates $\{0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$.

For a test case, we calculate the mean analysis error of fiber utilization, which is the mean of analysis errors of fiber utilization calculated over all fibers in the test case network. The analysis error of fiber utilization is the absolute value of the difference between the fiber utilization obtained with analysis and the fiber utilization obtained with simulation.

Figure 4a shows the mean analysis error of fiber utilization depending on the network load, while Fig. 4d shows the standard deviation of that error also depending on the network load. We define the network load as the mean fiber utilization calculated over all network fibers. Rather than plotting against flow mean rates, we plot against the network load measured in simulation in order to observe the influence of the network load on the mean and the standard deviation of the analysis error of fiber utilization. In both figures, one data point shows the mean of values for a sample, while the error bars show the standard errors of the means.

The mean analysis error of fiber utilization increases as the network load increases: for small loads the error is a few percent, but for about 50% load the error reaches about 10%. The error is about 20% relative to the network load, and on average to the actual fiber utilization.

The analysis error of a specific fiber utilization varies depending on the network load: for small loads, the standard deviation is a few percent, for about 20% load, the deviation reaches about 10%. For larger loads, the deviation stays constant at about 12%, which is an advantage, because the analysis error keeps the same predictability as the network load increases. For instance, at the network load of 70%, based on the Chebyshev's inequality, the utilization values of at least half of all fibers are analytically evaluated with at most $12\sqrt{2}\% \approx 17\%$ of error.

Figures 4b, 4c, 4e, and 4f show the comparison of simulation and analysis results. Each data point reports means of a sample, where gray error bars represent the standard errors of the means. Since there are 1600 data points in each figure, the error bars form gray clouds that indicate the range of values for test cases. We should be able to reduce the standard errors by increasing the number of test cases in a sample. The coordinates of a data point are the means of the same measured values: the horizontal-axis coordinate is obtained with simulation, while the vertical-axis coordinate with analysis.

The simulation and analysis results agree well, because the data points lie near the diagonal of the plots. The exception is seen in Fig. 4e, where a number of data points lie conspicuously above the diagonal; these data points are for a number of samples with very heavy loads. As shown in Fig. 4c, the packet loss probabilities (PLP) calculated by our method agree well for the values at the order of about 10^{-3}, and for smaller values start to differ substantially. Smaller values of ξ than the assumed $\xi = 10^{-2}$ should result in better estimates of the PLP, because a larger number of less likely routing arrangements would be considered which can cause packet losses.

V. Conclusion

Since the presented method is reasonably accurate and applicable to topologies of real networks, it could be used as a starting point for performance evaluation of future OPS networks, which could eventually become operational. The method could be also used for evaluating the impact of new flows on the network, and used in proactively establishing circuits in architectures such as OpenFlow.

Future work could incorporate into the method the support for practically any discrete distributions by modeling them with probability distribution tables. Also, the support for the asynchronous networks in our method could be researched, especially the required continuous probabilistic analysis of a node.

References

[1] N. Bouabdallah, "Sub-wavelength solutions for next-generation optical networks," *IEEE Communications Magazine*, vol. 45, no. 8, pp. 36–43, August 2007.

[2] L. Xu, H. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 136–142, January 2001.

[3] S. J. B. Yoo, "Optical packet and burst switching technologies for the future photonic Internet," *Journal of Lightwave Technology*, vol. 24, no. 12, pp. 4468–4492, December 2006.

[4] R. Van Caenegem, D. Colle, M. Pickavet, P. Demeester, K. Christodoulopoulos, K. Vlachos, E. Varvarigos, L. Stampoulidis, D. Roccato, and R. Vilar, "The design of an all-optical packet switching network," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 52–61, November 2007.

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

[6] S. Yao, B. Mukherjee, S. Yoo, and S. Dixit, "A unified study of contention-resolution schemes in optical packet-switched networks," *Journal of Lightwave Technology*, vol. 21, no. 3, pp. 672–683, March 2003.

[7] I. Szcześniak and 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, July 2007, pp. 117–120.

[8] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, "A performance model of deflection routing in multibuffer networks with nonuniform traffic," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 509–520, October 1995.

[9] A. Acampora and S. Shah, "Multihop lightwave networks: a comparison of store-and-forward and hot-potato routing," *IEEE Transactions on Communications*, vol. 40, no. 6, pp. 1082–1090, June 1992.

[10] I. Szcześniak, T. Czachórski, and J.-M. Fourneau, "Packet loss analysis in optical packet-switched networks with limited deflection routing," *Photonic Network Communications*, vol. 16, no. 3, pp. 253–261, December 2008.

[11] I. Szcześniak, "Analysis of a finite number of deflections in fully and uniformly loaded regular networks," in *Proceedings of NETWORKING 2004*, ser. Lecture Notes in Computer Science, vol. 3042/2004, May 2004, pp. 675–686.

[12] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," RFC 2475, December 1998.

[13] J. Jue, "An algorithm for loopless deflection in photonic packet-switched networks," in *Proceedings of ICC 2002*, vol. 5, May 2002, pp. 2776–2780.

[14] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," in *Proceedings of INFOCOM 2004*, vol. 3, March 2004, pp. 1558–1569.

[15] H. Øverby, "Traffic models for slotted optical packet switched networks," *Photonic Network Communications*, vol. 13, pp. 183–194, April 2007.

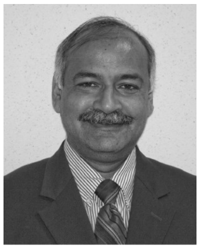
[16] I. Szcześniak. (2011, April) The website of OPUS, the Optical Packet Switching Universal Solver. [Online]. Available: http://www.irkos.org/opus

[17] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, "A versatile model for predicting the performance of deflection-routing networks," *Performance Evaluation*, vol. 16, no. 1-3, pp. 201–222, 1992.

Ireneusz Szcześniak (M'08) received the MS (honors) degree in computer science from the Silesian University of Technology in Gliwice, Poland in 2002, and the PhD degree also in computer science from the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences in Gliwice, Poland in 2009.

He worked for various laboratories and companies in the USA for a few years. Currently he is a university teacher and researcher at the Częstochowa University of Technology in Częstochowa, Poland. He is interested in optical networks, Linux networking, and parallel programming.

Dr. Szcześniak has been awarded several scholarships, most notably a Fulbright Scholarship. He is a member of the IEEE Communications Society, the Polish Fulbright Alumni Association, the Polish TeX Users Group, and the Polish Linux Users Group.



Biswanath Mukherjee (S'82–M'83–SM'05-F'06) received the B.Tech. (honors) degree from the Indian Institute of Technology, Kharagpur, India, in 1980, and the PhD degree from the University of Washington, Seattle in 1987.

He holds the rank of Distinguished Professor at University of California, Davis, where he has been since 1987, and served as Chairman of the Department of Computer Science during 1997 to 2000. He has supervised 47 PhDs to completion and currently has 20 advisees. He is co-winner of Optical Networking Symposium Best Paper Awards at IEEE Globecom 2007 and 2008. He is author of "Optical WDM Networks" (Springer, 2006), and Editor of Springer's Optical Networks Book Series. He served a 5-year term on Board of Directors of IPLocks, and served on Technical Advisory Board of several startup companies. His research interests include lightwave networks, network security, and wireless networks.

Prof. Mukherjee is a Fellow of IEEE. He was General Co-Chair of OFC-2011, TPC Co-Chair of OFC-2009, and TPC Chair of IEEE INFOCOM-96. He has served on eight journal editorial boards, most notably IEEE/ACM Transactions on Networking and IEEE Network.

Tadeusz Czachórski received the MS degree in 1972 and the PhD degree in 1979. He was awarded the habilitation in 1988, and the professor title in 1999.

Currently, he is a professor at the Silesian University of Technology in Gliwice, Poland and the director of the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences in Gliwice, Poland. He spent more than five years at several French universities and research institutes (IRISA, University of Versailles, Paris-Sud, Paris-Nord, INT) and still maintains scientific cooperation with them. He is a TPC member of international and national conferences such as Heterogeneous Networks NET-NETS, EuroNGI Conference, Polish Teletraffic Symposium, Computer Networks, and Internet in Information Society. He published over two hundred articles. He is interested in mathematical methods and software related to modelling and performance evaluation of wide area computer networks, especially the Internet.

Prof. Czachórski has been the scientific secretary of the Committee of Informatics of the Polish Academy of Sciences since 1990 and its vice-president since 2007.