

## Revenue maximization in optical router nodes

**Citation for published version (APA):**

Abidini, M. A., Boxma, O., Hurkens, C., Koonen, T., & Resing, J. (2020). Revenue maximization in optical router nodes. *Performance Evaluation*, 140-141, Article 102108. Advance online publication. <https://doi.org/10.1016/j.peva.2020.102108>

**Document license:**  
TAVERNE

**DOI:**  
[10.1016/j.peva.2020.102108](https://doi.org/10.1016/j.peva.2020.102108)

**Document status and date:**  
Published: 01/07/2020

**Document Version:**  
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

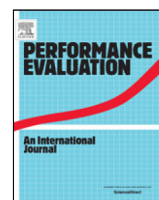
[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.



## Revenue maximization in optical router nodes

Murtuza Ali Abidini<sup>a,b</sup>, Onno Boxma<sup>a</sup>, Cor Hurkens<sup>a</sup>, Ton Koonen<sup>b</sup>,  
Jacques Resing<sup>a,\*</sup>

<sup>a</sup> Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

<sup>b</sup> Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands



### ARTICLE INFO

#### Article history:

Available online 7 May 2020

#### Keywords:

Optical routing  
Optical node  
Finite buffer  
Multiple wavelengths  
Revenue  
Optimization

### ABSTRACT

In this paper, a few models for optical router nodes are considered. The stations (ports) of such a node try to transmit packets. Successful transmission of a packet of type  $j$  at station  $i$  gives a profit  $\gamma_{ij}$ , but there is also a positive probability that such a packet is dropped, causing a penalty  $\theta_{ij}$ . Consider one fixed cycle (frame), in which each station is assigned some *visit time*. The goal is to choose the visit times in such a way that the revenue is maximized. In our first model there is only one wavelength, and we take the finiteness of buffers into account. The revenue maximization problem is shown to be separable concave, thus allowing application of a very efficient algorithm.

In our second model we allow multiple wavelengths. We aim to maximize the revenue by optimally assigning stations to wavelengths and, for each wavelength, by optimally choosing the visit times of the allocated stations within the cycle. This gives rise to a mixed integer linear programming problem (MILP) which is NP-hard. To solve this problem fast and efficiently we provide a three-step heuristic. It consists of (i) solving a separable concave optimization problem, then (ii) allocating the stations to wavelengths using a simple bin packing algorithm, and finally (iii) solving another set of separable concave optimization problems. We present numerical results to investigate the effectiveness of the heuristic and the advantages of having multiple wavelengths. Finally, some model variants are briefly discussed.

© 2020 Elsevier B.V. All rights reserved.

## 1. Introduction

In the last decades, optical fibers have emerged as the dominant transport medium in communication networks, because they offer major advantages over copper cables: huge bandwidth, extremely low drop probabilities and an extra dimension, viz., a choice of wavelengths (wavelength division multiplexing). Multiple wavelengths are to be used in order to enable the packet routing at various planes in the network (each at a specific wavelength). By including wavelength conversion, packets can be transferred between these planes, and thus congestion points can be circumvented. To handle packets at the Internet Protocol (IP) layer would imply lots of packet conversions from optical to electronic, after which the IP processing is done in the electrical domain, followed by a conversion back to optical. These Optical/Electrical/Optical (OEO) conversions introduce relatively significant time delays. All-optical routing in the nodes, as proposed and studied in

\* Corresponding author.

E-mail addresses: [m.a.abidini@tue.nl](mailto:m.a.abidini@tue.nl) (M.A. Abidini), [o.j.boxma@tue.nl](mailto:o.j.boxma@tue.nl) (O. Boxma), [c.a.j.hurkens@tue.nl](mailto:c.a.j.hurkens@tue.nl) (C. Hurkens), [a.m.j.koonen@tue.nl](mailto:a.m.j.koonen@tue.nl) (T. Koonen), [j.a.c.resing@tue.nl](mailto:j.a.c.resing@tue.nl) (J. Resing).

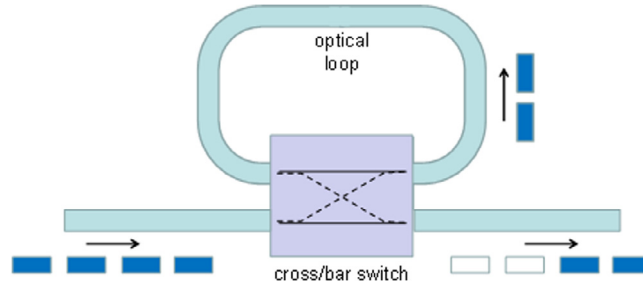


Fig. 1. Buffering packets in an optical FDL.

this paper, hence is valuable from the viewpoint of minimizing latency. In addition, avoiding this OEO conversion reduces the energy consumption in the network.

Future optical networks will probably need Optical Burst Switching (OBS) or Optical Packet Switching (OPS). The most used method to deal with contention in such optical networks is a combined use of wavelength converters and some type of optical buffering. However, the above-mentioned switching techniques offer substantial challenges, in particular w.r.t. buffering [1]; in fact, there have been proposals for optical routing and optical switching without the need for buffering (cf. [2–5]). Photons cannot be stored easily, and hence buffering of optical packets is more complicated than buffering in conventional communication systems. When photons need to be buffered, they are sent into a fiber delay line (FDL), which thus provides a small delay to the photons without displacing or losing them [6]. Packets can be inserted into and extracted from the FDLs by means of a cross/bar switch, cf. Fig. 1. If after the completion of such a loop the photon still cannot be transmitted, then it could again be sent into the FDL, or be considered as dropped. Such optical nodes are to be used in an all-optical packet-routing network, having multiple hops. See [7] for a discussion of stochastic modeling of optical buffers, and [8] for recent work on scheduling algorithms for situations with both optical buffers and wavelength converters.

The fact that photons/packets can be dropped naturally gives rise to an optimization problem. We allow several packet types because there can be several types of data at each port. We may subsequently assume that successful transmission of a packet of some type  $j$  at station (= port)  $i$  of the router node gives a profit  $\gamma_{ij}$ , but that there is also a positive probability that such a packet is dropped, causing a penalty  $\theta_{ij}$ . The packet drop probabilities depend on the amount of time a server (wavelength) is available for transmission of packets from station  $i$ . Hence one would like to determine how much time per cycle (= a predetermined frame time of fixed length) a station is allowed to transmit packets, using one of the available wavelengths.

In [9] we studied such an optimization problem, for the case of a single wavelength. We modeled a single-wavelength optical routing node as a queueing system with a single server (the wavelength) and  $N$  stations – the  $N$  ports of the routing node. We assumed that each successful transmission of a packet brings a certain profit. Our aim in [9] was to maximize the router performance by maximizing that profit. As a communication system typically works in frame time, we demanded that the time it takes the server to complete one cycle of the  $N$  stations is a given constant  $C$ . We then wanted to assign fixed amounts of time  $V_1, \dots, V_N$  to the visit periods (also called service windows) of the stations, such that  $\sum_{i=1}^N V_i = C - \sum_{i=1}^N S_i$ , where  $S_i$  is the time to switch to station  $i \in \{1, 2, \dots, N\}$ . We introduced the probability  $p_i(V_i)$  that a packet in a retrial loop of station  $i$  (representing an FDL) retries during visit period  $V_i$ , and the probability  $q_i(V_i)$  that a packet is dropped when it fails to retry during  $V_i$ . Under reasonable assumptions on those retry and drop probabilities, the revenue optimization problem in [9] was shown to be a separable concave optimization problem – a well-studied type of optimization problem that allows for an efficient and insightful algorithm (RANK; cf. [10]) that yields the optimal solution. Those assumptions are that  $p_i(\cdot)$  are increasing and concave, and  $q_i(\cdot)$  are decreasing and convex, and that the probability  $r_i(\cdot) := p_i(\cdot) + q_i(\cdot) - p_i(\cdot)q_i(\cdot)$ , that a packet in a retrial loop of station  $i$  leaves the system, is increasing.

The present paper considers a number of variants and extensions of the model of [9]. Firstly, in Section 2, we add the feature of finite buffers to the model of [9]. Optical buffering is one of the most severe bottlenecks in optical routing/switching, so considering finite buffers is quite relevant. Recirculating buffers, such as fiber loops, are inherently finite buffers; there cannot be more data buffered than the amount of data symbols which fits in the circumference of the fiber loop. We suggest an approximation for the drop probabilities of packets at the various stations. This approximation is not only very accurate, but also again gives rise to a separable concave revenue optimization problem, which can be solved by the RANK algorithm in a straightforward way.

Secondly, in Section 3, see also [11], we extend the model of [9] to the case of *multiple wavelengths*, thus doing justice to one of the key features offered by optical networking. Our goals in that section are (i) to formulate and solve the revenue optimization problem for an optical routing node with multiple wavelengths, and (ii) to investigate the advantage offered by having multiple wavelengths. It will turn out that the advantage, in terms of revenues, is very significant (in particular, going from one to two wavelengths). Solving the revenue optimization problem for multiple wavelengths is an NP-hard problem, and therefore we develop a heuristic; this heuristic is shown to work very well. Our numerical results give

insight into the sensitivity of various parameters and modeling assumptions. We restrict ourselves in Section 3 to the case of infinite buffers, because we prefer to focus on the aspect of multiple wavelengths without also having to add an approximation that allows us to handle finite buffers.

In Section 3.4 we briefly consider a variant of the multiple wavelength model, in which we now allow a station to be allocated to two adjacent wavelengths instead of assuming that each station must be allocated to exactly one wavelength.

Thirdly, in Section 4, we reflect upon one essential assumption made in [9,11] as well as in the first sections of the present paper, viz., that service times are negligible. Given that the aggregated line rate in a fiber network is typically amply exceeding the input data rates in the nodes, the assumption of negligible service times is quite realistic for line loads which are not so high that the line operates near congestion. Still, we believe it is interesting to extend the approach of [9,11] and the present paper to allow for nonnegligible service times. In Section 4 we suggest a simple approximation for the case of a single wavelength and infinite buffers (the setting of [9]) which allows one to solve a separable (but not necessarily concave) revenue maximization problem.

Section 5 contains conclusions and suggestions for further research.

## 2. Finite buffers

In this section we consider a single optical routing node with  $N$  stations which have finite buffers. We present a model description in Section 2.1, propose an approximation for the packet drop probabilities at the various stations in Section 2.2, consider the ensuing revenue maximization problem in Section 2.3, and present some numerical results in Section 2.4.

### 2.1. Model description

We model an optical routing node with  $N$  ports to route packets, and retrial loops to store packets. The representation we propose in this section is a single server polling model, i.e., a queueing model with a single server which cyclically visits all  $N$  queues. Customers, i.e., packets, arrive at queues  $1, \dots, N$  according to independent Poisson processes with rates  $\lambda_1, \dots, \lambda_N$ . The server visits queue  $i$  for a fixed time  $V_i$ ,  $i = 1, \dots, N$ , regardless of the numbers of customers present at the queues. After a visit to queue  $i$ , it switches to queue  $i + 1 \bmod N$ , which requires a switchover time  $S_{i+1 \bmod N}$ . A cycle along all  $N$  queues hence takes

$$C := S_1 + V_1 + \dots + S_N + V_N.$$

Such a fixed cycle time corresponds to the fixed frame time in which these communication systems often operate.

The dynamics at queue  $i$ ,  $i = 1, \dots, N$ , are as follows. If a packet arrives at queue  $i$  during a visit period  $V_i$ , then it is served instantaneously, with zero service time (see Section 4 for a relaxation of the latter assumption). Otherwise, the packet is placed in a buffer which can hold  $B_i < \infty$  packets. If the buffer is full upon arrival of a packet, that packet is dropped (lost). Finally, when the server returns to queue  $i$ , all packets in the buffer independently retry (which here amounts to instantaneously receiving service) with probability  $p_i(V_i)$ , which we for now allow to depend on  $V_i$ , and stay in the buffer till the next visit to queue  $i$  with probability  $1 - p_i(V_i)$ .

We assume that each successful transmission (service) at queue  $i$  yields a profit  $\gamma_i$ , whereas each dropped packet at queue  $i$  results in a penalty/cost  $\theta_i$ . Of course one could capture profits and costs in one parameter, but we prefer to make the profits and penalties separately visible.

Let  $q_i(V_i)$  denote the probability that an arbitrary packet arriving at queue  $i$  is dropped,  $i = 1, \dots, N$ . Here,  $q_i(V_i)$  is not an exogenous quantity, but it is determined later on and depends on the choice of the visit period. The net mean revenue per time unit at queue  $i$  then is given by

$$R_i(V_i) = \gamma_i \lambda_i (1 - q_i(V_i)) - \theta_i \lambda_i q_i(V_i), \quad i = 1, \dots, N. \quad (2.1)$$

Our goal is to maximize the net total mean revenue per time unit,

$$R(V_1, \dots, V_N) := \sum_{i=1}^N R_i(V_i), \quad (2.2)$$

by suitably choosing the lengths  $V_1, \dots, V_N$  of the visit periods, under the constraints that all  $V_i \geq 0$  and  $\sum_{i=1}^N V_i = C - \sum_{i=1}^N S_i$ . In the next subsection we turn to the determination of those drop probabilities.

### 2.2. An approximation for the packet drop probabilities

For a given visit period length  $V_i$ , there is no interaction between queue  $i$  and the other queues, and hence the drop probability  $q_i(V_i)$  only depends on the parameters involving queue  $i$ . Therefore we can determine  $q_i(V_i)$  by just analyzing the queue behavior at queue  $i$ . The only feature that connects the queues is the choice of the  $V_i$ , with its constraint  $\sum_{i=1}^N V_i = C - \sum_{i=1}^N S_i$ .

In the remainder of this subsection we omit the subscript  $i$ , focussing on some arbitrary queue with arrival rate  $\lambda$ , visit period  $V$ , retry probability  $p$  at the start of each visit, and drop probability  $q(V)$ . Obviously,

$$q(V) = \frac{C - V}{C} \pi, \quad (2.3)$$

where

$$\pi = P(\text{packet is dropped} | \text{arrival in non - serving period}). \quad (2.4)$$

$\pi$  clearly is the fraction of arrivals in a non-serving period that finds the buffer full. Denoting the number of packets in the buffer at the start of a non-serving period by  $X$  and the number of arrivals during that non-serving period by  $A$ , we have, with  $(y)_+ = \max(0, y)$ :

$$\pi = \frac{E[(X + A - B)_+]}{E[A]}. \quad (2.5)$$

$A$  is Poisson distributed with mean  $(C - V)\lambda$ . It is also not very difficult to determine  $E[(X + A - B)_+]$ , by observing the following. Let  $X_n$  denote the number of packets at the start of the  $n$ th non-serving period,  $A_n$  the number of arrivals in that period and  $Y_n$  the number of packets at the end of that period,  $n = 1, 2, \dots$ . Then  $\{X_n, n = 1, 2, \dots\}$  is an irreducible, aperiodic, positive recurrent Markov chain, which is specified by the recursion (with  $\text{Bin}(n, p)$  denoting a binomially distributed random variable with parameters  $n$  and  $p$ ):

$$X_n = \text{Bin}(Y_{n-1}, 1 - p), \quad Y_n = \min(X_n + A_n, B). \quad (2.6)$$

It is easy to determine the steady-state distribution  $P(X = j)$  for this Markov chain, and thus to determine  $\pi$  and  $q(V)$ . However, for the purpose of performing revenue maximization, we would like to have a relatively simple explicit formula for  $q(V)$ . Below we propose such a formula. In the next subsection we also show that it has the pleasing property of being convex in  $V$  at least for the case  $p(V) \equiv p$ ; that will allow us to use the RANK algorithm in maximizing mean total revenue as its expression becomes a separable, concave function of  $V_1, \dots, V_N$ .

Starting-point of our approximation is to remove the assumption that  $B$  is finite. When  $B$  is infinite, the steady-state number of customers at the start of a visit period, now denoted by  $\hat{X}$  to indicate that  $B$  is no longer assumed to be finite, satisfies the recursion, cf. (2.6),

$$\hat{X} \stackrel{d}{=} \text{Bin}(\hat{X} + A, 1 - p(V)). \quad (2.7)$$

Remember that  $A$  is Poisson distributed with mean  $(C - V)\lambda$ . Taking generating functions, or using Poisson properties regarding summation and thinning, it is easily seen that  $\hat{X}$ , too, is Poisson distributed, with  $E[\hat{X}] = (C - V)\lambda \frac{1 - p(V)}{p(V)}$ . It should be noted that, when approximating  $\pi$ , cf. (2.5), we make  $\pi$  too large by replacing  $X$  by  $\hat{X}$ . We propose to compensate for this by writing

$$\pi \approx C_0 \frac{E[(\hat{X} + A - B)_+]}{E[A]}. \quad (2.8)$$

Here  $C_0$  is a multiplicative constant, which we want to choose such that  $\pi$  is exact for  $B = 0$  (notice that  $\pi$  is also exact for  $B = \infty$ , since then  $\pi = 0$ ). Obviously, we then need to take  $C_0 = \frac{E[A]}{E[\hat{X} + A]}$ , resulting in

$$\pi \approx \frac{E[(\hat{X} + A - B)_+]}{E[\hat{X} + A]}. \quad (2.9)$$

From (2.3) and (2.9) we obtain the approximation

$$q(V) \approx \frac{C - V}{C} \frac{E[(\hat{X} + A - B)_+]}{E[\hat{X} + A]}. \quad (2.10)$$

We now provide an expression for  $E[(\hat{X} + A - B)_+]$ . Introducing  $Z := \hat{X} + A$ , with  $\hat{X}$  and  $A$  being independent, it immediately follows that  $Z$  is Poisson distributed with mean  $(C - V)\frac{\lambda}{p(V)}$ . Now

$$\begin{aligned} E[(Z - B)_+] &= \sum_{j=B+1}^{\infty} (j - B)P(Z = j) \\ &= -BP(Z > B) + \sum_{j=B+1}^{\infty} j e^{-(C-V)\lambda/p(V)} \frac{((C - V)\lambda/p(V))^j}{j!} \\ &= -BP(Z > B) + (C - V)\frac{\lambda}{p(V)} P(Z \geq B). \end{aligned} \quad (2.11)$$

Combining (2.10) and (2.11), and using  $E[\hat{X} + A] = (C - V)\frac{\lambda}{p(V)}$ , gives

$$q(V) \approx \frac{C - V}{C}P(Z \geq B) - \frac{Bp(V)}{\lambda C}P(Z > B). \quad (2.12)$$

Sijtsma in his bachelor thesis [12] has tested this approximation for a wide range of  $B$  values, concluding that the approximation is accurate over the whole range, the largest errors occurring roughly when  $B$  equals  $E[Z]$ .

### 2.3. Revenue maximization

As indicated at the end of Section 2.1, our goal in this section is to choose  $V_1, \dots, V_N$  such that the net mean revenue is maximized while satisfying some constraints on the  $V_i$ ,  $i = 1, \dots, N$ . Hence, cf. (2.1) and (2.2), we are faced with the following optimization problem:

$$\text{Max} \sum_{i=1}^N [\gamma_i \lambda_i (1 - q_i(V_i)) - \theta_i \lambda_i q_i(V_i)], \quad (2.13)$$

subject to

$$V_1, \dots, V_N \geq 0, \quad \sum_{i=1}^N V_i = C - \sum_{i=1}^N S_i. \quad (2.14)$$

Since  $V_i$  only appears in the  $R_i(V_i)$  part of the revenue function, this is a separable optimization problem. We now show that in the case  $p_i(V_i) \equiv p_i$  it is a separable *concave* optimization problem; for this we need to show that  $R_i(V_i)$  is a concave function of  $V_i$ , and hence that  $q_i(V_i)$  is convex. Once we have established this, we have shown that our revenue maximization problem falls in a class of optimization problems which are solved in a straightforward way by the RANK algorithm, cf. [10]. Again suppressing the subscript  $i$ , and using (2.12), we can write:

$$\frac{dq(V)}{dV} = -\frac{1}{C}P(Z \geq B) + \frac{C - V}{C} \frac{d}{dV}P(Z \geq B) - \frac{Bp}{\lambda C} \frac{d}{dV}P(Z > B). \quad (2.15)$$

The last two terms cancel, as can be seen in the following way: For  $B = 0, 1, \dots$ ,

$$\begin{aligned} \frac{d}{dV}P(Z > B) &= \sum_{j=B+1}^{\infty} \frac{d}{dV} e^{-(C-V)\lambda/p} \frac{((C-V)\lambda/p)^j}{j!} \\ &= -\frac{\lambda}{p} e^{-(C-V)\lambda/p} \frac{((C-V)\lambda/p)^B}{B!} = -\frac{\lambda}{p} P(Z = B). \end{aligned} \quad (2.16)$$

Similarly for  $\frac{d}{dV}P(Z \geq B)$ ; and finally use that  $P(Z = B) = \frac{(C-V)\lambda}{Bp} P(Z = B - 1)$  for  $B = 1, 2, \dots$

Hence we conclude that the derivative of  $q(V)$  w.r.t.  $V$  equals  $-\frac{1}{C}P(Z \geq B)$ , which is negative. Furthermore, the derivative is increasing in  $V$  as long as  $V$  is increasing towards  $C$ , as follows from (2.16). This shows that  $q(V)$  is a convex function of  $V$ , and hence that  $R(V)$  is concave.

### 2.4. Numerical results

In this subsection we use the RANK algorithm to present some numerical results for the case of three queues/stations; see Remark 1 below for a global description of RANK. Our baseline choice for the parameters is:  $C = 10$ ,  $S_i = 1/3$ ,  $\lambda_i = 1$ ,  $B_i = 10$ ,  $p_i = 1/2$ ,  $\gamma_i = 1$  and  $\theta_i = 1$  for  $i = 1, 2, 3$ . In the three tables of this subsection we vary one parameter (successively:  $\lambda_i$ ,  $p_i$  and  $B_i$ ), while keeping all other parameters symmetric and as just specified.

The numerical results for the visit lengths  $V_1, V_2, V_3$  suggest that the fraction of time spent at a queue  $i$  decreases with  $p_i$  (see Table 2) and with  $B_i$  (see Table 3), and increases with  $\lambda_i$  (see Table 1). These results could have been expected. Indeed, increasing  $p_i$  would result in  $i$ 's buffer emptying more during a visit period  $V_i$ . The buffer then has more space to fill during a non-serving period, so on average it takes longer to fill the buffer. Hence the number of dropped packets decreases, so a shorter visit period  $V_i$  suffices. A similar argument holds when  $B_i$  is increased: the buffer again has more space to fill, and a shorter visit period  $V_i$  suffices. The reverse is true when increasing  $\lambda_i$ . Now more packets arrive at the buffer, and the time to fill it up decreases. Hence more packets are dropped, so a longer visit period  $V_i$  is required to reduce the number of dropped packets. Also notice that in the numerical results of Table 1, the total revenue increases when some of the  $\lambda_i$ 's decrease. This is due to the fact that in that case less packets are dropped and as a consequence less penalty costs have to be paid.

**Remark 1.** We give a simple step-by-step guideline to follow the RANK procedure outlined in Section 2.2 of [10]. In (2.13) we are maximizing  $\sum_{i=1}^N R_i(V_i)$ , where the  $R_i(V_i)$  are concave increasing functions, and hence  $R'_i(V_i)$  are decreasing. Let  $C_S := C - \sum_{i=1}^N S_i$  represent the total available time to be divided among the stations.

**Table 1**  
Results for varying  $\lambda_i$ .

$\lambda_1$	$\lambda_2$	$\lambda_3$	$V_1$	$V_2$	$V_3$	$\sum_{i=1}^3 R_i(V_i)$
1	1	1	3.000	3.000	3.000	1.732
1	1	0.9	3.564	3.564	1.872	1.769
1	0.9	0.8	4.315	3.214	1.471	1.825
1	0.8	0.7	4.949	2.923	1.127	1.873
1	0.8	0.6	5.347	3.628	0.025	1.919
1	0.8	0.5	5.356	3.644	0.000	1.951

**Table 2**  
Results for varying  $p_i$ .

$p_1$	$p_2$	$p_3$	$V_1$	$V_2$	$V_3$	$\sum_{i=1}^3 R_i(V_i)$
0.5	0.5	0.5	3.000	3.000	3.000	1.732
0.5	0.5	0.625	3.538	3.538	1.923	1.928
0.5	0.625	0.75	4.400	3.000	1.600	2.255
0.5	0.625	0.875	4.750	3.438	0.813	2.385
0.375	0.625	0.875	5.800	3.000	0.200	2.255
0.25	0.625	0.875	6.857	2.143	0.000	2.098
0.1	1.0	1.0	9.000	0.000	0.000	2.475

**Table 3**  
Results for varying  $B_i$ .

$B_1$	$B_2$	$B_3$	$V_1$	$V_2$	$V_3$	$\sum_{i=1}^3 R_i(V_i)$
10	10	10	3.000	3.000	3.000	1.732
10	10	12	3.386	3.386	2.228	1.900
10	12	14	4.096	2.997	1.907	2.203
10	14	16	4.732	2.652	1.616	2.453
8	14	16	5.496	2.283	1.222	2.337
4	14	18	7.339	1.661	0.000	2.213
4	24	25	8.865	0.135	0.000	2.912

1. Calculate all  $R'_i(0)$  and sort them in decreasing order, say  $R'_1(0) \geq R'_2(0) \geq \dots \geq R'_N(0)$ .
  2. Allocate  $C_S$  to the station with largest slope  $R'_i(0)$ ; so here to station 1.
  3. Compute  $R'_1(C_S)$ .
    - If  $R'_1(C_S) \geq R'_2(0)$  then the procedure stops; the optimal strategy is  $V_1 = C_S$  and all other  $V_i = 0$ .
    - If  $R'_1(C_S) < R'_2(0)$  then solve for  $U_1$  and  $U_2$  such that  $R'_1(U_1) = R'_2(U_2)$  and  $U_1 + U_2 = C_S$ , i.e., the total time is divided between the two stations such that if there is any small additional time available it can be given to either station 1 or station 2, giving the same revenue.
    - Now either  $R'_1(U_1) \geq R'_3(0)$  or  $R'_1(U_1) < R'_3(0)$ . In the first case, the procedure stops. The optimal strategy is  $V_1 = U_1$ ,  $V_2 = U_2$  and all other  $V_i = 0$ . In the second case, we solve for  $W_1, W_2, W_3$  such that  $R'_1(W_1) = R'_2(W_2) = R'_3(W_3)$  and  $W_1 + W_2 + W_3 = C_S$ , i.e., the total time is divided between the three stations such that if there is any small additional time available it can be given to either of the three stations, giving the same revenue.
- And so on.

**Remark 2.** The choice  $p(V) \equiv p$  may not always be that realistic. In a future study, one could either consider other choices for  $p(V)$  that still give rise to a separable concave optimization problem, or sacrifice the concavity, thus having to solve a somewhat more complicated separable optimization problem.

**Remark 3.** It should be observed that allowing larger buffers amounts to having longer FDL. This implies a cost increase, which could also be taken into account in the revenue maximization.

### 3. Multiple wavelengths

In this section we consider an optical routing node with  $N$  stations, under the assumption that multiple wavelengths are available. We present a model description in Section 3.1, consider the revenue maximization problem in Section 3.2, present some numerical results in Section 3.3, and briefly discuss a variant in which stations may be allocated to two adjacent wavelengths in Section 3.4.

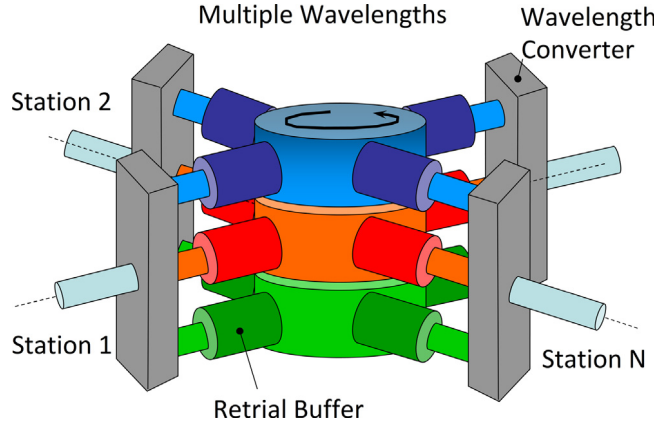


Fig. 2. Optical node with multiple wavelengths.

### 3.1. Model description

Consider a  $K$ -wavelength optical routing node with  $N$  stations (ports) to route packets and with fiber delay lines (retrial buffers) to store packets, cf. Fig. 2. We represent it by a queueing model with  $K$  servers which visit  $N$  queues. We shall assume that there is a fixed assignment of stations to servers, in which each station is assigned to only one server (how to do that assignment is part of our optimization problem).

**The packets:** Packets of type  $j$ ,  $j = 1, \dots, M$ , arrive at station  $i$ ,  $i = 1, \dots, N$ , according to independent Poisson processes with rate  $\lambda_{ij}$ , for all  $i, j$ . If at the time of packet arrival the station is being served (i.e., the station is being visited by a server = wavelength) then the packet is instantaneously transmitted; else it enters a retrial loop (FDL). We assume the retrial time to be random, because delay lines of various lengths may be used. If, at the time of retrial, the station is not in service then the packet again goes into a retrial loop and this process continues.

**The servers:** The servers go through cycles of fixed length  $C$  (the frame time). In each cycle a server visits each of its assigned stations once, for a fixed period of time  $V_i$  for station  $i$ . A visit to  $i$  is preceded by a deterministic switchover (setup) time  $S_i$  of the server. During  $V_i$ , there may be two types of arrivals: (i) newly arriving packets, and (ii) packets which were in a retrial loop; we assume the latter retry during  $V_i$  with some probability  $p_i(V_i)$ . In view of the huge available bandwidth, we assume the server serves all these packets (new arrivals + retrials) instantaneously, i.e., whenever a station is being served, any packet which arrives at it or retries, is transmitted immediately. Hence for practical purposes the service times are negligible (see Section 4 for a relaxation of the latter assumption). In this section, unlike the previous section, we assume that buffers are infinite. However, to take into account that in reality packets may get lost, we assume the following. At the end of each visit of station  $i$  each packet which still resides in a retrial loop of  $i$  is dropped with probability  $q_i(V_i)$ . Hence the probability that a packet in a retrial loop of station  $i$  leaves the system, either served during a visit at station  $i$  or dropped after a visit of station  $i$ , is  $r_i(V_i) := p_i(V_i) + q_i(V_i) - p_i(V_i)q_i(V_i)$ .

**Revenue:** Every served packet generates a profit and every dropped packet incurs a loss to the system. Our goal is to assign stations to servers, and subsequently visit times within a frame time  $C$  to stations, such that the revenue of the system is maximized. Assume that:

- a packet of type  $j$  served at station  $i$  gives a profit  $\gamma_{ij}$  (depending both on the type of packet and the type of source).
- a packet of type  $j$  dropped at station  $i$  causes a penalty  $\theta_{ij}$ . Indeed, the server has an obligation to meet the contract it has with each source. If the server fails to meet this contract it incurs a penalty: loss of packets/reputation/further contracts. One could also view  $\Theta_i := \sum_j \lambda_{ij}\theta_{ij}$  as contract costs of the service provider per time unit, and  $\Gamma_i := \sum_j \lambda_{ij}(\gamma_{ij} + \theta_{ij})$  as the maximum revenue that can subsequently be earned back by successfully serving packets.

For  $K = 1$  wavelength (cf. also [9] where that case was studied), the mean earnings per cycle are

$$\sum_j \lambda_{ij}\gamma_{ij} \left[ (C - V_i) \frac{p_i(V_i)}{r_i(V_i)} + V_i \right],$$

and the mean costs per cycle are

$$\sum_j \lambda_{ij}\theta_{ij} \left[ (C - V_i) \left( 1 - \frac{p_i(V_i)}{r_i(V_i)} \right) \right],$$

yielding the following net revenue for station  $i$  per cycle:

$$R_i(V_i) = M_i(V_i) - C\Theta_i,$$



where for all  $i = 1, \dots, N$ ,

$$M_i(V_i) := \Gamma_i \left[ (C - V_i) \frac{p_i(V_i)}{r_i(V_i)} + V_i \right]. \quad (3.1)$$

Note that  $\Gamma_i$  is the maximum available revenue that can be gained from station  $i$  per time unit. Since the server only serves a station during its visit period, all the arrivals during this period are served and hence we have the term  $\Gamma_i * V_i$ . But the packets which arrive during a non-visit period of a station are eventually served with probability  $\frac{p_i(V_i)}{r_i(V_i)}$ , and hence the revenue from this period is given as  $\Gamma_i * (C - V_i) * \frac{p_i(V_i)}{r_i(V_i)}$ . Further,  $\Theta_i$  is the cost incurred per time unit by the service provider to run the service. Choices of  $\Gamma_i$  and  $\Theta_i$  can be varied depending upon the traffic intensity, priorities, and available resources. These help the service provider to run, expand and sell its services. More details regarding these terms depend on the type of networks and nodes used, which is outside the scope of this paper. Finally, in [9] it was explained that since  $\Theta_i$  is the fixed cost incurred irrespective of how the resource is distributed, the maximization of  $\sum_i M_i(V_i)$  subject to conditions on  $V_i$  is enough to maximize the revenue  $\sum_i R_i(V_i)$  of the system subject to conditions on  $V_i$ .

Even if there are no explicit profits and costs attached to packet transmissions, the concept of using a revenue function for performance analysis of an optical switching node may provide us with various useful insights. Firstly, the revenue function acts as a substitute for the normalized throughput of the system. Hence it provides system owners a methodology for allocating optimal bandwidths to the various subscribers, and thus for optimizing their service (w.r.t. throughput). Secondly, the concept of the reward function helps the system to prioritize subscribers; those with higher priorities (higher time sensitivities) receive a higher reward and thus are assigned more bandwidth. Finally, the penalty function for the dropped packets forces the system to provide service even to the lowest priority packets, thereby maintaining the fairness of the system.

In the next subsection we present an algorithm to allocate the stations to different wavelengths such that each wavelength has a set of stations to serve; subsequently the visit periods are chosen such that the revenue for each wavelength is maximized.

### 3.2. Resource allocation

In this subsection we propose a procedure for solving the revenue maximization problem that was globally described in Section 3.1. For each wavelength  $k$ , we have  $C = \sum_{i \in \mathbf{P}_k} (S_i + V_i)$  where  $\mathbf{P}_k$  represents the set of all stations served by wavelength  $k$ . Note that if there is only one station being served by a wavelength, then there is no switchover involved. In that case,  $V_i = C$  where  $i$  is the only element of  $\mathbf{P}_k$ . Further we denote the set of stations which are each served by one complete wavelength as  $\mathbf{P}$  and the set of stations which are not served by any wavelength as  $\mathbf{Q}$ .

We now define the optimization problem REVENUE which produces maximum revenue via an optimal allocation of stations to wavelengths and visit periods to stations. With  $M_i(V_i)$  given in Eq. (3.1), and defining  $x_{ik} = 1$  if station  $i$  is served by wavelength  $k$ , but station  $i$  is not the only one being served by it, and 0 otherwise,  $y_{ik} = 1$  if station  $i$  is the only station being served by wavelength  $k$ , and 0 otherwise, we have:

#### REVENUE

$$\max \sum_{i=1}^N M_i(V_i)$$

subject to

$$\sum_{i=1}^N [(S_i + V_i)x_{ik} + V_i y_{ik}] = C, \quad \forall k = 1, 2, \dots, K, \quad (3.2)$$

$$\sum_{k=1}^K [x_{ik} + y_{ik}] \leq 1, \quad \forall i = 1, 2, \dots, N, \quad (3.3)$$

$$\sum_{i=1}^N x_{ik} + N \sum_{i=1}^N y_{ik} \leq N, \quad \forall k, \quad (3.4)$$

$$x_{ik}, y_{ik} \in \{0, 1\} \text{ and } 0 \leq V_i \leq C, \quad \forall i, k. \quad (3.5)$$

Conditions (3.2) and (3.5) are system properties, and they state that the allocation per wavelength should be equal to its capacity  $C$  and the visit period cannot be negative or more than  $C$ . Condition (3.3) states that each station  $i$  can only be served by at most one wavelength. Finally, condition (3.4) states that if for a wavelength  $k$  some  $y_{ik} = 1$  then no other station can be served on it.

This problem is a non-linear mixed integer programming problem. Under certain realistic assumptions regarding the system parameters (see also [9]), we can reduce the objective function of this maximization problem to separable concave terms; however, the occurrence of the integers  $x_{ik}$ ,  $y_{ik}$  prevents us from using the RANK algorithm [10] that was used

in [9]. The so-called PARTITION problem, which is NP-complete [13], is a special case of REVENUE, as will be elucidated below; hence REVENUE is an NP-hard problem.

The PARTITION problem deals with a finite set of non-negative numbers  $\{q_1, q_2, \dots, q_N\}$  and the question is whether or not one can split the set into two subsets such that the sum of the numbers in each of the subsets equals  $\frac{1}{2} \sum_{j=1}^N q_j$ . This problem is related to our problem REVENUE in the special case that  $K = 2$ ,  $C = \frac{1}{2} \sum_{j=1}^N q_j$  and  $S_i = 0$  for all  $i$ . If we choose  $\frac{p_i(V_i)}{r_i(V_i)} = 1 - e^{-V_i}$ , we have that  $M_i(V_i) = \Gamma_i C + \Gamma_i(V_i - C)e^{-V_i}$  and hence  $M'_i(V_i) = \Gamma_i e^{-V_i}(1 + C - V_i)$ . Now if we choose  $\Gamma_i = e^{q_i}/(1 + C - q_i)$  we see that  $M'_i(q_i) = 1$ , for all  $i$ , and hence choosing  $V_i = q_i$  gives the optimal solution for the optimization problem  $\max \sum_{i=1}^N M_i(V_i)$  subject to  $\sum_{i=1}^N V_i \leq 2C$ . So, if we solve our problem REVENUE and find a solution such that  $\sum_{i=1}^N M_i(V_i) = \sum_{i=1}^N M_i(q_i)$ , then apparently the answer to the PARTITION problem is positive (and the solution of REVENUE gives the split in the PARTITION problem). On the other hand, if we find a solution such that  $\sum_{i=1}^N M_i(V_i) < \sum_{i=1}^N M_i(q_i)$ , then apparently the answer to the PARTITION problem is negative.

Below we propose a heuristic to solve REVENUE. We argue that this heuristic should produce results which are close to optimal, and we provide numerical results in Section 3.3 to support that claim.

The idea behind our approach is the following. In *Step 1* we act as if there is only one wavelength, but a frame time of length  $KC$  instead of  $C$ . We use the RANK algorithm to get an optimal choice of the visit periods  $\tilde{V}_i$  for such a situation. That should already give a quite good first estimate of the visit periods. In *Step 2* we use those  $\tilde{V}_i$  values to assign stations to wavelengths. This is done such that each of the  $K$  wavelengths gets roughly the same  $\sum(S_i + \tilde{V}_i)$  – which hence should be close to  $C$ . Finally, in *Step 3*, with those  $K$  allocations we use RANK again, but now for  $K$  separate single-wavelength problems. Below we provide the details of these three steps.

**Step 1** We first define the following optimization problem.

**ONE**

$$\begin{aligned} & \max \sum_i M_i(\tilde{V}_i) \\ & \text{subject to } \sum_i \tilde{V}_i = KC - \sum_i S_i, \\ & \text{and } 0 \leq \tilde{V}_i \leq C - S_i, \quad \forall i. \end{aligned}$$

The solution of this optimization problem gives us the values of  $\tilde{V}_i$  required by each station to give the maximum revenue, subject to the condition that the maximum amount of resource available is  $KC$ . The upper bound on  $\tilde{V}_i$  is included because a station cannot be served by more than one wavelength. Note that  $M_i(\tilde{V}_i)$  is the same as given in Eq. (3.1).

We solve the (separable, concave) optimization problem ONE using RANK, and we thus obtain values of  $\tilde{V}_i$ . Every station  $i$  which has  $S_i + \tilde{V}_i = C$ , is allocated to a single wavelength. These stations belong to the set  $\mathbf{P}$  and as described at the start of this section, all stations belonging to this set have their visit periods equal to  $C$ . Further, all the stations with  $\tilde{V}_i = 0$  belong to the set  $\mathbf{Q}$ . These stations will not be allocated to any wavelength, and as mentioned earlier they will have zero visit period. By renumbering, we may assume that the stations in  $\mathbf{Q}$  are the highest numbered stations, immediately preceded by the stations in  $\mathbf{P}$ . Also assume that the latter  $N(\mathbf{P})$  stations (where  $N(\mathbf{P})$  denotes the number of elements in  $\mathbf{P}$ ) are assigned to the  $N(\mathbf{P})$  highest numbered wavelengths.

We now turn to our procedure for assigning stations to wavelengths (Step 2) and subsequently determining the exact visit periods (Step 3).

**Step 2** Take the values of  $S_i + \tilde{V}_i$  for the first  $N - N(\mathbf{P} + \mathbf{Q})$  stations (i.e., those not in  $\mathbf{P}$  or  $\mathbf{Q}$ ). Sort these values in descending order, say  $S_1 + \tilde{V}_1 \geq S_2 + \tilde{V}_2 \geq \dots \geq S_{N-N(\mathbf{P}+\mathbf{Q})} + \tilde{V}_{N-N(\mathbf{P}+\mathbf{Q})}$ . Then allocate those stations to the first  $K - N(\mathbf{P})$  wavelengths following the so-called *Longest Processing Time first* (LPT) rule. This amounts to first assigning stations  $1, \dots, K - N(\mathbf{P})$  to wavelengths  $1, \dots, K - N(\mathbf{P})$ ; and subsequently assigning each of the remaining stations, one by one in descending order of their values, to that wavelength for which the sum of the already assigned values is the smallest. This procedure is continued until all stations have been assigned. For example, in the 4-station, 2-wavelength case of Table 5, if the values for  $\tilde{V}_i$  would be as in the fourth row of the table, we would have  $\mathbf{P}$  empty (no  $\tilde{V}_i = C$ ) and  $\mathbf{Q}$  empty (no  $\tilde{V}_i = 0$ ). We then would order the  $S_i + \tilde{V}_i$  from large to small. According to LPT, station 2 (with the highest  $S_i + \tilde{V}_i$ ) was first assigned to wavelength 1 and station 4 to wavelength 2; subsequently station 3 was assigned to wavelength 2 (which had a smaller total assigned value than wavelength 1), and finally station 1 was assigned to wavelength 1.

**Remark 4.** The idea to use LPT comes from multiprocessor scheduling. Consider a set of  $N$  tasks which have to be served on  $K$  parallel servers. The service of a task on a server, once started, cannot be interrupted. In multiprocessor scheduling the goal often is to minimize the *makespan*, i.e., the time until all tasks are completed. This is an NP-hard problem. The makespan minimization problem can be reformulated in the terminology of bin-packing, where it amounts to finding the smallest common capacity of the bins, sufficient to pack all  $N$  pieces. Many heuristics have been developed for solving the bin-packing or makespan minimization problem; see, e.g., [14]. LPT is a simple and accurate heuristic procedure. It is intuitively clear that assigning tasks in decreasing order of size should work well when  $K$  and  $N$  are not too small: because the smallest tasks are assigned last, it is likely that all makespans are close to each other. See [15] for a probabilistic analysis of various bin-packing heuristics, and [16] for a probabilistic analysis of LPT list scheduling.

**Table 4**  
3 station system.

Allocation	Visit period	Revenue
[1 1 2]	<b>[0.48 1.12 2.00]</b>	<b>10.11</b>
[1 2 1]	[0.28 2.00 1.32]	9.81
[2 1 1]	[2.00 0.61 0.99]	8.65

**Table 5**  
4 station system.

Allocation	Visit period	Revenue
<b>[0 1 1 2]</b>	<b>[0.00 0.61 0.99 2.00]</b>	<b>14.65</b>
[1 2 2 1]	[0.14 0.61 0.99 1.46]	14.25
[1 2 1 2]	[0.28 0.48 1.32 1.12]	14.03
[1 1 2 2]	[0.48 1.12 0.67 0.93]	13.34
[1 1 1 2]	[0.00 0.61 0.99 2.00]	14.65
[1 1 2 1]	[0.00 0.48 2.00 1.12]	14.22
[1 2 1 1]	[0.00 2.00 0.67 0.93]	13.23
[2 1 1 1]	[2.00 0.00 0.67 0.93]	11.23

**Step 3** Now that we have assigned all stations to a wavelength, we still need to determine the visit periods for those stations that use wavelengths  $1, \dots, K - N(\mathbf{P})$ , because the extended visit periods  $S_i + \tilde{V}_i$  of the stations that are assigned to a particular wavelength do not exactly sum up to  $C$ . For this we solve optimization problem TWO, for  $k = 1, \dots, K - N(\mathbf{P})$ :

#### TWO

$$\begin{aligned} & \max \sum_{i \in \mathbf{P}_k} M_i(V_i) \\ & \text{subject to } \sum_{i \in \mathbf{P}_k} V_i = C - \sum_{i \in \mathbf{P}_k} S_i, \\ & \text{and } V_i \geq 0, \quad \forall i \in \mathbf{P}_k. \end{aligned}$$

The solution of this optimization problem gives us the values of  $V_i$  required by each station allocated to wavelength  $k$ , subject to the maximum amount of resource available at that wavelength. We thus obtain new extended visit periods  $S_i + V_i$  for stations  $1, \dots, N - N(\mathbf{P} + \mathbf{Q})$ .

**Remark 5.** If, in Step 2, a station  $i^*$  is the only one being assigned to a wavelength, then we do not run TWO for it but take  $V_{i^*} = C$ .

This concludes the description of the heuristic procedure. In the next section we shall investigate its accuracy. Its computational complexity is low. The optimization problems ONE and TWO are concave separable with linear constraints and can be solved in polynomial time; and we use ONE once, TWO at most  $K$  times. We also use LPT once. Further, we need to sort the extended visit periods in Step 2 once.

### 3.3. Numerical examples

In this subsection we present a few numerical examples to illustrate various properties of our system. For all the examples in this section we assume that the probability of retrial and drop probability for a station  $i$  are given by  $p_i(V_i) = 1 - e^{-v_i V_i}$  (corresponding to exponentially distributed retrial times) and  $q_i(V_i) = e^{-\mu_i V_i}$ . Further, the revenue of a station  $i$  is equal to  $M_i(V_i)$  as given in Eq. (3.1) (and not  $R_i(V_i)$ , see above (3.1); hence we have nonnegative revenues). It should be noticed that these  $p_i(\cdot)$  and  $q_i(\cdot)$  are, respectively, increasing concave and decreasing convex functions, while  $r_i(\cdot)$  are increasing. Problem ONE featuring in Step 1 now is a separable concave optimization problem.

**Example 1.** We first consider a toy example with  $K = 2$  wavelengths and either  $N = 3$  or  $N = 4$  stations, for which all possible assignments allocating all stations to a wavelength are listed. For each station  $i$ , the parameters  $v_i$  and  $\mu_i$  are equal to 0.5. The switchover times  $S_i = 0.2$  for each station  $i$  and frame time  $C = 2$ . Finally,  $\Gamma_i = i$ , for each station  $i$ . The allocation of stations to different wavelengths is shown, along with the corresponding visit period (obtained by using TWO) and the revenue obtained by the system. Note that an allocation 0 implies that the station was not allocated to any wavelength.

In Tables 4 and 5 the values given by our procedure described in the previous section are printed boldface. We observe that in both cases our procedure gives the best allocation. In Table 4, the allocation [1 1 2] indicates that stations 1 and 2 are assigned to wavelength 1 and station 3 to wavelength 2. The [0.48 1.12 2.00] in this table implies that a frame for wavelength 1 consists of a visit period 0.48 for station 1, followed by an 0.2 switchover time, an 1.12 visit period for station 2 and an 0.2 switchover time, while a frame for wavelength 2 is fully occupied by a 2.00 visit period of station 3.

**Table 6**  
Varying  $\Gamma_i$ .

	Maximum	Average	Minimum	Percent
(i)	475.72	468.89	454.24	1.46
(ii)	475.50	441.36	300.33	0.24
Algorithm	474.51			

$\Gamma_i = 0.5 * i$ ,  $v_i = 0.5$ ,  $\mu_i = 0.5$  and  $S = 0.2$ .

**Table 7**  
Varying  $v_i$ .

	Maximum	Average	Minimum	Percent
(i)	387.29	384.58	381.94	9.89
(ii)	387.14	358.36	224.93	0.87
Algorithm	385.65			

$\Gamma_i = 4$ ,  $v_i = 0.05 * i$ ,  $\mu_i = 0.5$  and  $S = 0.2$ .

**Table 8**  
Varying  $\mu_i$ .

	Maximum	Average	Minimum	Percent
(i)	413.19	413.15	412.98	0.00
(ii)	413.19	377.54	231.52	0.00
Algorithm	413.19			

$\Gamma_i = 4$ ,  $v_i = 0.5$ ,  $\mu_i = 0.05 * i$  and  $S = 0.2$ .

**Table 9**  
Varying  $S_i$ .

	Maximum	Average	Minimum	Percent
(i)	398.81	398.06	395.60	0.00
(ii)	398.79	351.53	181.94	0.00
Algorithm	398.81			

$\Gamma_i = 4$ ,  $v_i = 0.5$ ,  $\mu_i = 0.5$  and  $S = 0.05 * i$ .

**Table 10**  
Completely random.

	Maximum	Average	Minimum	Percent
(i)	360.85	355.23	338.07	4.56
(ii)	360.83	338.14	231.45	0.62
Algorithm	359.93			

$\Gamma_i \sim U(0, 8)$ ,  $v_i \sim U(0, 1)$ ,  $\mu_i \sim U(0, 1)$ , and  $S_i \sim U(0, 0.4)$ .

**Example 2.** In this example we compare the results obtained using our procedure with the results obtained by randomly allocating wavelengths to different stations and then optimizing the visit periods at each wavelength. We show numerical results for five different cases for a system with  $N = 16$  stations,  $K = 4$  wavelengths and frame time  $C = 8$ . In each of the first four cases, we vary one parameter while keeping all the other constant and in the last case we use random system parameters; the  $\Gamma_i$  are uniformly distributed on  $(0, 8)$ ; the  $v_i$  and  $\mu_i$  on  $(0, 1)$ , and the  $S_i$  on  $(0, 0.4)$ .

We take 10000 independent allocations of wavelengths in two different ways, (i) and (ii). In (i) we allocate stations in such a way that each wavelength gets at most 4 stations, whereas in (ii) there is no restriction on the number of stations allocated to a wavelength. In both cases we subsequently use TWO. For both (i) and (ii) we show the maximum, the average and the minimum obtained revenue among the 10 000 cases and the percentage of allocations which generated a revenue above the value generated using our algorithm.

Tables 6–10 suggest that a random assignment of stations to wavelengths, but still using TWO to subsequently choose  $V_i$ , is much worse than the assignment of our algorithm. However, the symmetric assignment, in which each of the four wavelengths serves (at most) four out of the 16 stations, and for which the visit times are calculated using TWO, yields results that are typically quite close to the values obtained using our algorithm (and in a few cases even better).

**Example 3.** In this example we study which effect increasing the number  $K$  of wavelengths has on the revenue of the system. We take the allocation obtained using the procedure of Section 3.2. For each  $K$  we take  $N = 16$  stations,  $S_i = \mu_i = v_i = 0.05 * i$ ,  $\Gamma_i = 0.5 * i$  and  $C = 8$  (see Table 11).

**Table 11**  
Varying the number of wavelengths.

$K$	Revenue	# of stations served
1	170.54	3
2	322.62	8
3	400.97	11
4	452.88	13
5	480.40	14
6	499.60	14
7	517.23	15
8	525.21	15
16	544.00	16

**Table 12**  
 $\Gamma_i = 0.5 * i$ ,  $v_i = 0.5$  and  $\mu_i = 0.5$ .

Station	Allocation	Visit	Revenue
1	0	0.00	0.00
2	0	0.00	0.00
3	3	0.93	6.54
4	4	1.22	10.68
5	4	1.45	14.89
6	3	1.67	19.27
7	2	2.16	24.96
8	1	2.25	28.90
9	1	2.34	32.89
10	2	2.46	37.00
11	3	2.20	39.45
12	4	2.23	43.23
13	4	2.30	47.24
14	3	2.40	51.49
15	2	2.78	57.03
16	1	2.81	60.94
Total		29.20	474.51

We observe that increasing the number of wavelengths increases the revenue obtained and also the number of stations served. However, the marginal increment decreases with an addition of each wavelength. In this example the change from  $K = 1$  to  $K = 2$  almost doubles the revenue and more than doubles the number of stations served, whereas the change from  $K = 7$  to  $K = 8$  increases the revenue by less than two percent (and the number of stations served does not change). In the case of  $K = 16$ , the revenue equals  $C * \sum_{i=1}^{16} \Gamma_i = 544$ . The system operator can choose an optimal number of wavelengths so as to maximize its utility. This observation may be of interest in networks where traffic is highly variable and the cost of running extra resources is high.

**Example 4.** In this example we consider a system with  $N = 16$  stations,  $K = 4$  wavelengths, frame time  $C = 8$  and switchover period from each station  $S_i = 0.2$ , for all  $i = 1, \dots, N$ . We show three different cases, each of which has one of  $\Gamma_i$ ,  $v_i$ , and  $\mu_i$  different for all stations, the other two parameters being equal for all stations. In these numerical experiments we study how the procedure described in Section 3.2 allocates resources depending on each factor, and develop insight into the influence of these factors on the system performance. In Table 12, we mention the wavelength to which each station is assigned, the visit period each station receives and the revenue each station gives, for the three cases.

From Table 12 we see that in general  $\Gamma_i > \Gamma_j$  does not imply  $V_i > V_j$ , but when  $i$  and  $j$  are allocated to the same wavelength this implication appears to be true. Also, if the value of  $\Gamma_i$  is very low, then – even though our procedure allocates that station to a wavelength – it may not receive any service (equivalent to not being allocated).

In Table 13 we see that in general, within a wavelength, stations with lower  $v_i$  receive higher  $V_i$ . This happens because the system tries to allocate longer visit periods to stations with low retrial rates so as to maximize the number of packets it can serve. However, if  $v_i$  is very low (see station 1), then the system, subject to limited resources, might not allocate any resource to that station.

From Table 14 one can generally observe that the stations with higher drop probability, i.e., lower  $\mu_i$ , receive longer visit periods to have fewer losses. Also, like in the previous case the difference in revenue generated from each station is not big.

Three final observations: 1. The spread in visit periods is small in Table 14 compared to those in Tables 12 and 13. This suggests that the factor  $\mu_i$  is less important than the factors  $v_i$  and  $\Gamma_i$  in the solution of this problem. 2. Our procedure often results in a more or less even spread of revenues among stations if  $\Gamma_i$  are equal. This suggests that the procedure makes the system reasonably fair, i.e., tries to provide the best service to each station. 3. Even though the revenues obtained

**Table 13** $\Gamma_i = 4$ ,  $v_i = 0.05 * i$  and  $\mu_i = 0.5$ .

Station	Allocation	Visit	Revenue
1	0	0.00	0.00
2	1	3.35	26.05
3	2	2.33	22.33
4	3	2.18	23.09
5	4	2.07	23.83
6	4	1.97	24.37
7	3	1.88	24.80
8	2	1.83	25.30
9	1	2.16	28.02
10	4	1.69	25.85
11	3	1.64	26.09
12	2	1.60	26.42
13	1	1.89	28.59
14	3	1.50	26.76
15	4	1.47	26.96
16	2	1.44	27.19
Total		29.00	385.65

**Table 14** $\Gamma_i = 4$ ,  $v_i = 0.5$  and  $\mu_i = 0.05 * i$ .

Stations	Allocation	Visit	Revenue
1	3	1.85	22.76
2	4	1.86	23.36
3	2	1.87	23.94
4	1	1.87	24.48
5	3	1.86	24.90
6	4	1.85	25.29
7	2	1.84	25.66
8	1	1.83	26.01
9	1	1.82	26.32
10	3	1.80	26.56
11	2	1.78	26.81
12	4	1.76	27.03
13	4	1.73	27.23
14	2	1.71	27.43
15	3	1.69	27.62
16	1	1.68	27.79
Total		28.80	413.19

from stations with different retrial rates and drop probabilities are similar, the resources required by these stations are different. For a lower retrial rate and/or higher drop probability, a longer visit period is required to give similar revenue. This is a techno-economic trade-off to consider while designing the router.

### 3.4. Multiple wavelengths – a variant

So far in this section we assumed that each station can be allocated to at most one wavelength. We now briefly discuss a variant in which stations can be allocated to two adjacent wavelengths. This is technically possible and offers additional flexibility, but at the expense of requiring some additional switchover times. We mention two options for studying this trade-off. A very simple approach would be the following. After optimal visit periods  $\tilde{V}_i$  are determined for the case of a frame time of length  $KC$  in Step 1 of the heuristic procedure, just divide these visit periods over the  $K$  different wavelengths by cutting the frame of length  $KC$  into  $K$  pieces each of length  $C$ . In that way, some stations are allocated to two adjacent wavelengths because they are in one of the cuts. In this approach, we do not have to go through Steps 2 and 3 of the procedure. However, a disadvantage of allocating a station, say station  $i$ , to two different wavelengths is that on both wavelengths also a switchover time has to be scheduled (and as a consequence some visit periods have to be shortened and some revenue is lost). Hence one preferably only allocates station  $i$  to two wavelengths if the corresponding switchover time  $S_i$  is small. This brings us to the following, somewhat more refined, heuristic.

For  $\ell = 0, 1, \dots, K - 1$  select the set of stations  $J_\ell$  with the  $\ell$  smallest switchover times  $S_i$ . The stations in  $J_\ell$  are the stations that will be allocated to two neighboring wavelengths. For  $\ell = 0$ , the set  $J_\ell$  is empty and we assign stations to wavelengths and visit periods to stations according to the three-step approach sketched in Section 3.2. For  $\ell > 0$ , we adapt the three-step approach in the following way. In Step 1 we apply ONE with modified constraint  $\sum_i \tilde{V}_i = KC - \sum_i S_i - \sum_{i \in J_\ell} S_i$ . In Step 2, we then consider a  $K$ -machine scheduling problem with jobs of length  $2S_i + \tilde{V}_i$ ,

for  $i \in J_\ell$ , and jobs of length  $S_i + \tilde{V}_i$ , for  $i \notin J_\ell$ . Modify the first set of jobs by sorting the jobs from large to small and cutting each job in two halves, of lengths  $S_i + \frac{1}{2}\tilde{V}_i$ . Assign these  $2 * \ell$  half jobs over the  $K$  machines, by putting the  $j$ th half job on machine  $j$ , for  $j < K$ , and by putting half jobs  $j = K, \dots, 2\ell$  on machine  $2K - j$ , respectively. Remark that in this way two corresponding half jobs are scheduled on neighboring machines (as wanted) and furthermore for large  $\ell$ , when many machines will get two half jobs, these machines either get one large and one small half job or two middle-sized half jobs. After that we assign the remaining jobs of length  $S_i + V_i$ , for  $i \notin J_\ell$ , to the different machines according to the LPT rule. In Step 3 we adapt the sizes of the jobs (i.e. the visit periods of the stations) in order to achieve that each machine obtains a total amount of work equal to  $C$ . This can be done by either shifting part of the work of half jobs on a machine to the corresponding half jobs on neighboring machines or alternatively by solving TWO again for each of the machines separately. In this way we get different heuristic solutions for different choices of  $\ell$ , and at the end we choose the  $\ell$  and the corresponding solution for which the revenue is maximal.

#### 4. Nonnegligible service times

So far, we have assumed that service times are negligibly small, arguing (cf. the end of Section 1) that this is quite realistic in most settings under consideration. In the present section we briefly consider the case that we cannot assume that the service times are zero, sketching a possible approach that basically allows one to follow the analysis in [9]. Apart from the nonzero service times we also follow their setting, i.e., the buffers are infinite, and there is only one wavelength available. Below we focus on one arbitrary queue  $i$ , again suppressing the subscript  $i$ . Assume that service times have a mean  $E[T] > 0$ . Also assume that service times are typically considerably shorter than  $V$ .

Let  $\zeta$  denote the probability that a new arrival during a  $V$  period can immediately receive service. We propose to approximate  $\zeta$  by the fraction of time that there is no service in  $V$ . Assume that, when a new arrival cannot immediately receive service, it is a candidate for a retrial. Assume that such arrivals, as well as arrivals during a non-visit period, have a probability  $p(V)/r(V)$  of still being served. Then the counterpart of  $M_i(V_i)$  as studied in (3.1) is given by

$$M(V) = \gamma\lambda[\zeta V + (1 - \zeta)V\frac{p(V)}{r(V)} + (C - V)\frac{p(V)}{r(V)}]. \quad (4.1)$$

Hence the mean total service time during one  $V$  period equals

$$\lambda E(T)[\zeta V + (1 - \zeta)V\frac{p(V)}{r(V)} + (C - V)\frac{p(V)}{r(V)}], \quad (4.2)$$

implying that  $\zeta$  satisfies the following equation:

$$1 - \zeta = \lambda E(T)[\zeta + (1 - \zeta)\frac{p(V)}{r(V)} + (\frac{C}{V} - 1)\frac{p(V)}{r(V)}], \quad (4.3)$$

so, with  $\rho := \lambda E(T)$ :

$$\zeta = \frac{1 - \rho \frac{C}{V} \frac{p(V)}{r(V)}}{1 + \rho - \rho \frac{p(V)}{r(V)}}. \quad (4.4)$$

Finally, we need to maximize  $\sum_{i=1}^N M_i(V_i)$  under the usual constraints  $V_1, \dots, V_N \geq 0$  and  $\sum_{i=1}^N V_i = C - \sum_{i=1}^N S_i$ , with (now no longer suppressing subscripts)

$$M_i(V_i) = \gamma_i \lambda_i C \frac{p_i(V_i)}{r_i(V_i)} + \gamma_i \lambda_i V_i \left(1 - \frac{p_i(V_i)}{r_i(V_i)}\right) \frac{1 - \rho_i \frac{C}{V_i} \frac{p_i(V_i)}{r_i(V_i)}}{1 + \rho_i - \rho_i \frac{p_i(V_i)}{r_i(V_i)}}. \quad (4.5)$$

$M_i(V_i)$  is not necessarily concave. Hence one cannot use RANK, and the numerical evaluation of the maximization problem is more involved.

#### 5. Conclusions and suggestions for further research

To understand the behavior and study the performance of future optical networks, we have considered a few revenue optimization problems for single- and multiple-wavelength optical routing nodes. The two main models under consideration were (i) a single-wavelength model in which we focussed on the issue of finite buffers, and arrived at a separable concave optimization problem; and (ii) a model in which we explored the advantages of having multiple wavelengths, and arrived at a mixed integer non-linear programming problem. The latter problem is extremely time-consuming to solve even for a small number of wavelengths. Since one would like to solve this revenue optimization problem quite frequently, we have developed an efficient and near-optimal heuristic procedure for (i) assigning stations to wavelengths and subsequently (ii) assigning visit times to stations within a fixed frame time.

Several topics for further research suggest themselves. Firstly, one might make adaptations to the proposed heuristic procedure for the multiple-wavelength model. For example, the extended visit periods  $S_i + \tilde{V}_i$  from Step 2 in Section 3.2,

of the stations that are assigned to a particular wavelength, do not exactly sum up to  $C$ ; we therefore used TWO in Step 3 to make final choices for the visit periods  $V_i$ . Instead, one could simply scale all  $V_i$ , that belong to one and the same wavelength, by the same factor  $\alpha$  such that  $\sum(S_i + \alpha V_i) = C$ . Also the approach sketched in Section 3.4 could be explored further. Secondly, one might work out the approach sketched in Section 4 to handle the case that service times are not negligibly small. Finally, it would be worthwhile to study the trade-off between investing in a higher number of fiber delay lines – which should result in a lower drop probability – and using more wavelengths.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

The research in this paper is supported by NWO Gravitation project NETWORKS, Grant Number 024.002.003. The authors are indebted to Arjen Sijtsma for his contributions to Section 2.

### References

- [1] M. Maier, *Optical Switching Networks*, Cambridge University Press, Cambridge, 2008.
- [2] D. Barth, P. Bethomé, D. Chiaroni, J.M. Fourneau, C. Laforest, S. Vial, Mixing convergence and deflection strategies for packet routing in all-optical networks, *J. Opt. Commun. Netw.* 3 (2009) 222–234.
- [3] L. Sadeghioon, A. Gravey, B. Uscumlic, P. Gravey, M. Morvan, Full featured and lightweight control for optical packet metro networks, *J. Opt. Commun. Netw.* 7 (2015) A235–A248.
- [4] A. Triki, I. Popescu, A. Gravey, X. Cao, T. Tsuritani, P. Gravey, TWIN as a future-proof optical transport technology for next generation metro networks, in: *International Conference on High Performance Switching and Routing (HPSR)*, 2016, pp. 87–92.
- [5] I. Widjaja, I. Saniee, R. Giles, D. Mitra, Light core and intelligent edge for a flexible, thin-layered, and cost-effective optical transport network, *IEEE Commun. Mag.* 41 (2003) S30–S36.
- [6] F. Masetti, P. Gavignet-Morin, D. Chiaroni, G. Da Loura, Fiber delay lines optical buffer for ATM photonic switching applications, in: *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 1993, pp. 935–942.
- [7] W. Rogiest, *Stochastic Modeling of Optical Buffers* (Ph.D. Thesis), Ghent University, Ghent, 2008.
- [8] K. Van Hautegeem, *Scheduling Algorithms for Optical Packet and Burst Switched Networks* (Ph.D. Thesis), Ghent University, Ghent, 2018.
- [9] M.A. Abidini, O.J. Boxma, A.M.J. Koonen, J.A.C. Resing, Revenue maximization in an optical router node – allocation of service windows, in: *International Conference on Optical Network Design and Modeling (ONDM)*, Cartagena, Spain, 2016.
- [10] T. Ibaraki, N. Katoh, *Resource Allocation Problems*, MIT Press, Cambridge, 1988.
- [11] M.A. Abidini, O.J. Boxma, C.A.J. Hurkens, A.M.J. Koonen, J.A.C. Resing, Revenue maximization in an optical router node using multiple wavelengths, in: *Proc. VALUETOOLS 2019*, Palma de Mallorca, Spain, 2019.
- [12] A. Sijtsma, *Revenue Optimization in an Optical Router-Node* (Bachelor thesis), Eindhoven University of Technology, 2017.
- [13] M.R. Garey, D.S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, 1990.
- [14] E.G. Coffman, M.R. Garey, D.S. Johnson, Approximation algorithms for bin-packing – a survey, in: D.S. Hochbaum (Ed.), *Approximation Algorithms for NP-Hard Problems*, 1997, pp. 46–93.
- [15] H.L. Ong, M.J. Magazine, T.S. Wee, Probabilistic analysis of bin packing heuristics, *Oper. Res.* 32 (1984) 983–998.
- [16] O.J. Boxma, A probabilistic analysis of the LPT scheduling rule, in: E. Gelenbe (Ed.), *Performance'84*, 1985, pp. 475–490.

**Jacques Resing** is assistant professor in the Stochastic Operations Research group of the Department of Mathematics and Computer Science at Eindhoven University of Technology. He received his Ph.D. degree in Mathematics from Delft University of Technology in 1990. His research interests include queueing theory and the performance analysis of communication and production systems.