Towards an Analytical Tool to Support Planning of 400ZR+-Enabled IPoWDM Networks

Nicolas Jara,[1,*], Gerardo Rubino,[2], Hermann Pempelfort[1], Patricia Morales[1] and Alejandra Beghelli[3]

[1] *Dept. of Electronics Enginering, Universidad Tecnica Federico Santa Maria, Valparaiso, Chile*
[2] *INRIA Rennes Bretagne-Atlantique, Rennes, France*
[3] *Optical Networks Group, Electronic & Electrical Engineering Dept., University College London, London, UK*
[*] *nicolas.jara@usm.cl*

Abstract: A highly accurate analytical procedure to support emerging ZR+-enabled IPoWDM network planning tools is presented. The procedure is four orders of magnitude faster than simulation, allowing extensive and fast network analysis and design.

1. Introduction

Fast IP routers equipped with Terabit-capable Network Processing Units and pluggable module cages where low-cost, compact transceivers (known as 400ZR/ZR+ modules) can be plugged into are enabling, for the first time, an integrated IPoWDM architecture. Although still limited by the reach of ZR/ZR+ modules, this architecture exhibits two remarkable features for potentially higher capacity exploitation in the core. First, the dynamicity of the IP layer is transferred to the optical layer, allowing the allocation of spectrum resources only when and where required. Second, hop-by-hop wavelength convertible operation eliminates spectrum fragmentation due to continuity constraints. Both features are vital to achieving significant resource savings compared to current static operation [1]. These features make core IPoWDM networks an attractive near-term alternative for network operators when reach limitations are overcome.

One of the many challenges faced by this new integrated IPoWDM architecture is the lack of planning tools that include the parameters of the 400ZR/ZR+ pluggable modules [2]. In this paper, we present the first step to building a fast and accurate tool (available at GitLab [1]) to support the dimensioning stage of the planning process of an integrated IPoWDM architecture. Dimensioning requires determining how much capacity is required to operate the network for a maximum blocking ratio. This task is usually performed by iteratively simulating the network under different capacity scenarios for dynamic networks. However, this approach is very time-consuming. Integer Linear Programming models, also computationally complex, struggle with the non-linear nature of dynamic networks. Last, analytical models such as Erlang/Engset models are significantly faster. However, they usually rely on some unrealistic assumptions (e.g., link blocking independence, homogeneous traffic), thus inaccurate in most cases.

In this paper, we present a computationally simple and yet accurate tool to support the dimensioning process of an IPoWDM network by estimating the blocking ratio for a given capacity analytically. By applying a novel recurrence-based approach, the tool is four orders of magnitude faster than simulation and achieves the same results. This allows for fast and accurate dimensioning, a key stage of the network planning process.

2. Network Model

The integrated IPoWDM network is represented by a graph G=(N,L), where N is the set of nodes and L is the set of unidirectional links, with cardinalities N and L, respectively. The set of connections $C \subseteq N^2$, with cardinality C, is made of all source-destination node pairs on graph G.

Each node represents an IP router with Z 400ZR+ modules per link. Each 400ZR+ module can transmit at 400 Gbps using a spectral width of 75 GHz with a maximum reach of 960 km [3]. The IP router is assumed to perform electronic grooming of slow-rate client signals onto 400 Gbps optical channels.

Traffic requests are represented by the triplet (s,d,m), where s and d are the identifiers of the source and destination nodes, and m is an integer number representing how many 400Gbps signals are requested. Traffic temporal evolution between each node pair follows an ON-OFF model. During the ON period (of average length t_{ON_c}), the source transmits at a constant rate bw_c (a multiple m of 400 Gbps). During the OFF period (average length t_{OFF_c}), the source is inactive. The traffic load ρ_c offered by a given connection c is given by $\rho_c = \frac{t_{ON_c}}{t_{ON_c}+t_{OFF_c}}$.

[1] Code available at: https://gitlab.com/njarac/ipowdm-engine

3. The Maths Behind the Tool

Planning tools' performance depends on the speed and accuracy of their inner computing engines. Here, we present the maths enabling a fast and accurate engine for blocking evaluation of IPoWDM networks. By organising the computation steps in a novel recurrence-based manner (see Eq.(3) below), the engine takes fractions of a second to compute blocking.

Link Blocking Independence Assumption Relaxation. The computation of BC_c, the probability of connection c request being blocked can be simplified assuming link independence [4], as shown in Eq. (1):

$$BC_c = 1 - \prod_{\ell \in r_c} (1 - BL_\ell), \tag{1}$$

where BL_ℓ is the probability that a connection request is blocked on link ℓ, with $\ell \in L$. The link independence assumption decreases accuracy, but can be relaxed by using Kelly's Reduced Load Fixed Point method [5]. This method is an iterative procedure where the value of BC_c is successively refined until the relative difference between 2 consecutive values is lower than a given threshold. The refinement is performed by adjusting the arrival rate of each connection as $\lambda'_c = \lambda_c(1 - BC_c)$, modifying the link blocking, and consequently the connections blocking. For a fast calculation of BL_ℓ, for all $\ell \in L$, we have derived a recursive method, succinctly explained below.

Definitions. Let $T_\ell = \{c \mid \ell \in r_c\}$ be the set of connections that use a given link $\ell \in L$ in their respective paths, where r_c is the path used by connection c. To simplify the notation, connections using link $\ell \in L$ are (arbitrarily) renumbered with the numbers $1, 2, \ldots, T_\ell$. **Let** $X = \{X_c \mid c \in T_\ell\}$ be the set composed by X_c the state of the connection $c \in T_\ell$, where $X_c = 1$ if the connection is active (ON), and 0 otherwise (OFF). **Let** $P(X_c)$ be the probability that connection c is in a given state X_c. The state of a connection in link ℓ is independent of the state of the other connections, thus $P(X) = \prod_{c: c \in T_\ell} P(X_c)$. **Let** $\beta(X)$ **be** the total bandwidth occupied by the active connections (in ON state) belonging to the set X. That is, $\beta(X) = \sum_{c: X_c \in X} bw_c \cdot X_c$.

Fast Link Blocking Evaluation. Let $P_\ell(bw, T_\ell)$ the probability that link ℓ has exactly bw frequency slots used. Thus, bw is the sum of the bandwidth used by all active connections in T_ℓ and $P_\ell(bw, T_\ell)$ is the sum of the probability associated with all states X where the total bandwidth occupied equals bw, e.g., $\beta(X) = bw$. Going through all 2^{T_l} combinations of X increases exponentially with the size of the network and would make the method very slow. However, we can compute each $P(X)$ recursively adding one by one connection of the set X using conditional probabilities, this is $P(X) = P(X_c) \cdot P((X - \{X_c\})/X_c)$. Due to the independence of the states of the connections of the same link, the conditional probability $P((X - \{X_c\})/X_c)$ does not depend on the state X_c, so the evaluation of $P_\ell(bw, T_\ell)$ can be reduced to:

$$P_\ell(bw, T_\ell) = \sum_{X:\ \beta(X) = bw} P(X) = \sum_{X:\ \beta(X) = bw} P(X_c) \cdot P(X - \{X_c\}). \tag{2}$$

By mean value definition, $\rho_{c,\ell}$, the average traffic load of the connection $c \in T_\ell$ in the link ℓ is given by $\rho_{c,\ell} = P(X_c = 1) \cdot 1 + P(X_c = 0) \cdot 0$, for all $c \in T_\ell$. Therefore, $P(X_c = 1) = \rho_{c,\ell}$ and $P(X_c = 0) = 1 - \rho_{c,\ell}$. These definitions allow us to rewrite (2) in the following recurrence, key for fast calculation of blocking:

$$P_\ell(bw, C_k) = \begin{cases} 1, & \text{if } bw = k = 0 \\ \rho_{k,\ell} \cdot P_\ell(bw - bw_k, C_{k-1}) + (1 - \rho_{k,\ell}) \cdot P_\ell(bw, C_{k-1}), & \text{if } 0 \le bw \le \sum_{i=0}^{k} bw_i, k \ne 0,\ bw \ne 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where C_k is a subset of T_ℓ that contains k connections. Note that (3) allows evaluating $P_\ell(bw, T_\ell)$ recursively, and it does not depend on the identity of the k connections in C_k, but only that C_k contains k different connections. Consequently, the order in which Eq. (3) is evaluated in terms of the set C_{T_ℓ} is arbitrary. Note that we do not need to save all the probability, but only the last ones, which are the ones needed to evaluate.

Finally, BL_ℓ is equal to the rate of connection requests blocked in link ℓ divided by the total rate of connections arriving at link ℓ. The probability that connection c is blocked on ℓ link occurs when such connection requests are attempted to be established, and the available capacity of the link Z_ℓ is less than bw_c. In this case, the c connection is blocked because there is not enough spectrum to service the request. Thus:

$$BL_\ell = \frac{\sum_{c=1}^{T_\ell} \lambda_{c,\ell}(1 - \rho_{c,l}) \sum_{bw = Z_\ell - bw_c + 1}^{Z_\ell} P_\ell(bw, T_\ell - \{c\})}{\sum_{c=1}^{T_\ell} \lambda_{c,\ell}(1 - \rho_{c,l}) \sum_{bw=0}^{Z_\ell} P_\ell(bw, T_\ell - \{c\})}. \tag{4}$$

where $\lambda_{c,\ell}$ is the arrival rate from connection c to link ℓ computed as $\lambda_{c,\ell} = 1/t_{ON_c} + t_{OFF} : c$, and $P_\ell(bw, T_\ell - \{c\})$ is evaluated by means of the (3), for which the set C_k corresponds to the connections belonging to the set $T_\ell - \{c\}$.

4. Numerical Results

The accuracy and execution time of the blocking computing engine were evaluated and compared against an ad-hoc event-driven simulation built in Python. The simulation stops when enough connection requests were generated to obtain a relative statistical error of less than 5%, achieving a confidence interval of 95%. The threshold for the difference between two consecutive values of connection blocking was set to 10^{-6}. 21-node UKNet network topology with link lengths under the reach limit of ZR+ modules were studied (maximum link length equal to 460 km). The bitrate requested by each source-destination pair was assumed to be a multiple m=[1,3] of 400 Gbps. The mean ON period was set to 10 ms and the mean OFF period was varied to obtain different traffic loads ρ_c, in the range [0.05-0.5]. Each node was assumed to be equipped with $Z = 16$, 20 and 24 ZR+ modules per link.

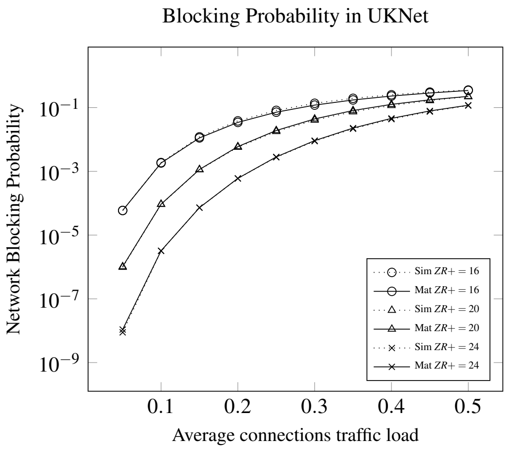

Fig. 1: Network blocking probability as a function of traffic load, obtained by simulation (`Sim`) and our engine (`Mat`).

ZR+ Modules	Method	Execution Time(s)
16	`Mat`	$9.30 \cdot 10^{-3}$
16	`Sim`	$4.63 \cdot 10^{2}$
20	`Mat`	$1.47 \cdot 10^{-2}$
20	`Sim`	$6.30 \cdot 10^{2}$
24	`Mat`	$1.72 \cdot 10^{-2}$
24	`Sim`	$7.83 \cdot 10^{2}$

Table 1: Average execution time of simulation (`Sim`) and our engine (`Mat`) for different network capacities.

Figure 1 shows the blocking probability obtained by the method presented here contrasted against simulation. The engine always obtained results very close to simulation, showing that the proposed procedure is highly accurate. Similar results were obtained for 3 other topologies, not included here due to lack of space. The proposed engine slightly overestimates blocking with respect to simulation, which allows the dimensioning process to guarantee the required quality of service.

The main difference between our engine and the simulation method is the average time required to evaluate the blocking probability for all traffic loads, as shown in Table 1. In all the experiments, we can observe that our proposal is four orders of magnitude faster than the simulation technique. For instance, in the case of IP routers equipped with 20 ZR+ modules per node, our engine took a hundredth of a second ($1.47 \cdot 10^{-2}$ seconds) while the simulation tool required around 10 minutes and 30 seconds ($6.30 \cdot 10^{2}$ seconds). In dimensioning tasks, where the blocking evaluation must be repeated hundreds of times to adjust the network capacity to the required blocking, such a difference can amount to several hours or even days of additional computational effort using simulation.

Future work will evaluate the benefits of the engine for dimensioning survivable IPoWDM network configurations and extending the tool into a proper network planning tool to ease the decision-making process of IPoWDM networks.

Acknowledgements Financial support from ANID FOVI210082, USM PIIC 018/2022, ANID Doctorado Nacional/2020-21200588 and ANID FONDECYT Iniciación 11201024 is gratefully acknowledged.

References

1. A. Zapata-Beghelli and P. Bayvel, "Dynamic vs. static wavelength routed optical networks, " J. Light. Technol. **26**, 3403-3415 (2008)
2. O. Gonzalez de Dios et al., "MANTRA Whitepaper: IPoWDM convergent SDN architecture," (Telecomm Infra Project, August 2022)
3. A. Gumaste et al., "Optimized IP-over-WDM core networks using ZR+ and flexible muxponders for 400Gb/s and beyond," J. Opt. Commun. Netw. **14**, 127-139 (2022)
4. N. Jara, R. Vallejos, and G. Rubino, "Blocking evaluation and wavelength dimensioning of dynamic WDM networks without wavelength conversion," J. Opt. Commun. Netw. **9**, 625–634 (2017)
5. F. P. Kelly, "Loss networks," The Annals Appl. Probab. **1**, pp. 319–378 (1991).