Traffic Grooming of Batches of Deadline-Driven Requests in Elastic Optical Networks

Pedro M. Moura *Rafael A. Scaraficci*
Nelson L. S. da Fonseca

Technical Report - IC-17-06 - Relatório Técnico
June - 2017 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Traffic Grooming of Batches of Deadline-Driven Requests in Elastic Optical Networkss

Pedro M. Moura[*] Rafael A. Scaraficci[†] Nelson L. S. da Fonseca[‡]

Resumo

This technical report introduces a novel traffic grooming algorithm for the connection establishment of deadline-driven requests in elastic optical networks, named Elastic Batch Grooming Algorithm. The algorithm grooms batches of requests to establish lightpaths with diverse bandwidth demands with deadline requirements. Results show that the algorithm significantly reduces the blocking ratio and the number of demanded transponder when compared to traditional non-batch algorithms.

1 Introduction

One of the main characteristics of the Internet architecture is to impose no constraint on the application layer, which allows the fast emergence of new applications with heterogeneous demands. A common QoS requirement of theses applications is the deadline to finish the data transmission. Applications which have such requirements are called deadline-driven.

Requests for connection establishment by deadline-driven applications can be postponed until the time at which data should be transmitted at the maximum available rate to meet the required deadline. The possibility of postponing a connection establishment can benefit both users and service providers. If bandwidth is unavailable, a connection establishment can be postponed avoiding the blocking of the request. Moreover, service providers can choose the rate of a connection according to the availability in the network links.

A batch is a set of requests which arrived in a period starting at the arrival time of the oldest request and ending at the earliest time one of the request should start transmission in order to meet its deadline. The flexibility of setting the starting time of a connection allows the creation of batches of requests that can be scheduled as a whole rather than as a sequence of individual requests. At the scheduling time of a batch, each connection will demand a specific transmission rate which can be groomed on a set of existing or potentially allocable lightpaths. The scheduling of batches allows greater combinations of connection in grooming decisions.

In Wavelength Division Multiplexing (WDM), the fixed capacity of a wavelength accommodates demands of different sizes. This leads to under utilization of the spectrum

---

[*] Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

[†] Telecom Research and Development Center, Brazil

[‡] Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

since demands rarely match the exact capacity of a wavelength. Sub-wavelength demands are usually groomed to decrease the waste of capacity. Such rigidness has recently led to the emergence of spectrum-sliced elastic optical path networking. In this technology, (Optical) Orthogonal Frequency Division Multiplexing (OFDM) is employed. OFDM is a multi-carrier transmission technology that slits high data rate channels into a number of orthogonal channels, called subcarriers, each with (sub-wavelength) low data rates. Elastic optical networks have gained great momentum and have attracted attention from industry and academia due to the technology maturity that enables their development and deployment.

Differently than WDM networks in which the optical spectrum is divided into frequency slots of fixed width 50GHz or 100GHz, allowing up to 40 and 80 wavelengths respectively, in elastic optical networks the spectrum is divided into slots with finer granularity, e.g. 12.5GHz or even 6.25GHz; and slots can be combined and assigned to a connection according to the requested bandwidth and modulation technique applied to convert the electrical signal into the optical signal.

The requirements of deadline-driven applications can be efficiently handled in elastic optical networks by an ASON/GMPLS or a SDN control plane to automate the establishment and tear down of lightpaths with variable spectrum widths. Although the high granularity of elastic optical networks allows efficient spectrum usage, it also has inherent problems. The proximity of connections allocated in spectrum using OFDM brings the necessity of using a band guard to separate the connections so that they do not interfere with each other. Another problem brought by the high number of subcarriers is the need of one transponder to transmit each set of contiguous subcarriers, increasing operational expenditure of the network [1].

Similar to the routing and wavelength assignment (RWA) problem in fixed-grid WDM networks, solutions for the routing and spectrum assignment (RSA) problem in elastic optical networks are needed to efficiently accommodate traffic demands. Besides the spectrum continuity constraint that imposes the allocation of the same spectrum in each fiber along the route of a lightpath, in an RSA formulation, slots (carrier) must be contiguously allocated in the spectrum (the spectrum contiguity constraint). Moreover, a lightpath can be defined by an RSA algorithm to carry a batch of connections to be groomed in this lightpath.

This paper introduces a novel traffic grooming algorithm that takes advantage of the deadline requirements of applications. Requests for connection establishment are grouped into batches and a lightpath allocated to the connections in the batch. In this way, there is no need to have guard bands separating the connections and this demands only a single transponder rather than several ones to serve a batch of connections.

This paper is organized as follow. The next section describes related work. Section III introduces the batch grooming algorithm and new cost functions. Section IV evaluates the performance of the batch grooming algorithm. Section V concludes the paper.

2 Related Work

Several papers addresses related issues in WDM networks. In [2], it was proposed an algorithm for dynamic traffic grooming aware of holding time which aggregates connections with known lifetime in order to optimize the timing of connection establishment. An extension of the algorithm that balances the load to achieve higher fairness was proposed in [3]. In [4], it was proposed a holding time aware algorithm that supports multipath routing for the provision of high capacity connections.

In [5], it introduced an interval graph that represents the times to schedule sets of requests mitigating conflicts and lowering the blocking ratio.

Cavdar at all. [6] defines delay tolerance as the maximum period duration that a request can wait to establish a connection. If there is no available capacity to serve the connection request, it can be postponed (redialed) as many times as allowed by its delay tolerance. Unlike the approach in [7], the transmission rate is fixed as well as the holding time. The employment of delay tolerance in scheduling criteria to differentiate service classes in WDM networks was investigated in [8]. The quality of service differentiation was studied based on information about connection setup time [9].

Time flexibility was employed in [10] to implement advanced reservation scheme for delivering quick responses to users. Re-provisioning of requests is employed to improve performance. An algorithm for advanced reservation for elastic applications was defined in [11].

A solution for deadline driven batch traffic grooming for WDM networks was proposed in [12]. It was proposed an algorithm called BatchGrooming that creates batches of requests with the same source destination pair based on their deadlines. The batches are allocated into wavelengths by an algorithm called Groom-Solver, that employs an integer linear programming to groom batches onto existing lightpaths or create new ones in case of need.

A traffic grooming algorithm for elastic optical networks was proposed in [1], which creates an auxiliary graph to represent the network components using three types of edges: lightpath edges, which represent the lightpaths already allocated. Transponder edges and spectrum edges are used to find new lightpaths in the network. The graph is used to try to groom arriving requests into existing lightpaths if there is an available path using lightpath edges between two nodes, or it can be used to find a paths to establish a new lightpath.

The algorithm in the present paper grooms connections which requests arrive dynamically and which have deadline requirements. The proposed algorithm explores the flexibility resulted from different deadline values to create and schedule batches, rather than to schedule individual requests. It is our best knowledge that this is the first batch grooming algorithm for elastic optical networks.

3 Batch Grooming Algorithm for Elastic Optical Networks

The proposed algorithm was designed to operate in elastic optical networks with dynamic arrival of requests for connection establishment. Each request specifies the amount of bytes to be transmitted and the deadline to finish the transmission. A connection is not necessarily provisioned upon arrival, and a deadline for scheduling it is computed based on the available capacity in the network and its QoS deadlines requirement. The required transmission rate transmission of a connection increases with the postponing of its scheduling time. Moreover, a connection is blocked if it is neither established at its deadline nor sufficient capacity is available to schedule it at its deadline.

The proposed algorithm takes advantage of the fact that some connection request can be delayed without violating the deadline constraint. This brings up the opportunity to create batches of requests for the same source and destination pair of nodes. Each batch has a scheduling deadline, which is determined by the request with more stringent deadline. At the batch scheduling deadline, an attempt to establish a lightpath is made and if it is established the batch connections is transmitted on it.

By gathering connections in a batch and establishing an elastic lightpath for transmitting all its connections instead of establishing an individual elastic lightpath for each connection, it is expected that the number active lightpaths will be reduced, saving guard bands between connections which increases the available capacity and consequently decreases the blocking ratio.

Before presenting the proposed algorithm, let us introduce some notion:

- s_i: source node of the connection request i;
- d_i: destination node of the connection request i;
- N_i: number of bytes required to be transmitted by the connection request i;
- D_i: deadline at which the transmission of request i must finish;
- $R_i = (s_i, d_i, N_i, D_i)$: deadline-driven connection request i;
- $r_i(t) = N_i/(D_i - t)$: transmission rate of the connection i with scheduling time t;
- $B_{s,d}(t) = (s, d, b)$: set of connection requests at time t with source s and destination d;
- $SB_{s,d}(t)$: scheduling deadline of the batch $B_{s,d}(t)$;
- $P_{s,d}$: set of postponed connection requests with source s and destination d;
- R: request of connection establishment;
- $RSA(R)$: the RSA algorithm used for lightpath establishment.

The Routing and Spectrum Assignment algorithm proposed in this section was designed to operate in networks with dynamic arrival of requests for the establishment of lightpaths. It is assumed that the RSA algorithm is implemented in ideal Path Computation Elements (PCE) and that information about the status of spectrum availability is stored in the PCEs databases. Each request defines the size of the message to be sent and the deadline for its transmission.

The proposed algorithm takes advantage of the possibility to avoid the immediate allocation of a requests, and postpone its establishment in order to create batches of connections for the same source destination pair. The connections in a batch are then allocated at once, or partially allocated if the spectrum availability does not allow the allocation of all of them. When a batch of connection is allocated, the connections are not separated by guard bands as if they were allocated as individual lightpaths for them. Moreover, they can be transmitted by only one transponder.

s_i: source node of the request R_i;

d_i: destination node of the request R_i;

b_i: number of bytes requested to be transmitted by the request R_i;

D_i: deadline of the request R_i;

$R_i = (s_i, d_i, b_i, D_i)$: the i^{th} request;

$r_i(t) = \frac{N_i}{D_i - t}$: bandwidth necessary to transmit the request R_i at time t

$B_{s,d}(t) = (s, d, b)$: batch of requests at time t, composed by a set of requests, the traffic demand is equals to the sum of all requested bandwidths r_i in the set;

$SB_{s,d}(t) = D(r_i(s, d, b)) \mid \forall j\ D(r_i(s, d, b)) < D(r_j(s, d, b))$: deadline of the batch $B_{s,d}$, given by the earliest deadline of the requests in the batch;

$P_{s,d}$: set of requests to have their allocation postponed;

$Q_{s,d}$: set of requests to be blocked;

P_n: path found by the shortest path algorithm where n is the first slot to be allocated;

Algorithm 1 ElasticBatchGrooming

```
$B_{s,d}(t) = B_{s,d}(t) \cup \{r_i\}$
At $SB_{s,d}(t)$
$P_{s,d} = \emptyset$
$Q_{s,d} = \emptyset$
$P_n = \emptyset$
while $B_{s,d}(t) \neq \emptyset$ or $P_n \neq \emptyset$ do
    $P_n = RSA(G(t), B_{s,d}(t))$
    if $P_n = \emptyset$ then
        Let $l_j \mid \forall R_k \in B_{s,d}(t)\ D_j > D_k$
        $B_{s,d}(t) = B_{s,d}(t) - l_j$
        if $D_j > SB_{s,d}(t)$ then
            $R_{s,d} = R_{s,d} \cup \{l_j\}$
        else
            $Q_{s,d} = Q_{s,d} \cup \{l_j\}$
        end if
    else
        establish $B_{s,d}(t)$ as $P_n$
    end if
end while
postpone $R_{s,d}$
block $Q_{s,d}$
```

Algorithm 1 details the deadline batch grooming algorithm. In Line 1 the new request is added to the batch of requests with same source and destination nodes ($B_{s,d}(t)$). The batch is then scheduled to be groomed at time $SB_{s,d}(t)$, which is the earliest deadline of a request in the batch (Line 2).

The loop in Line 6 repeats the search for routes and spectrum in the network in case no route with sufficient contiguous slots is available to supply the batch demand. In Line 7, an RSA algorithm is executed and in case a route with sufficient spectrum is found P_n is set to a value other than $\emptyset$. In case no path is found (Line 8), the algorithm selects the connection with the latest deadline $l_{s,d}$ (Line 9) to try to postpone its allocation since it has higher chances of being allocated in the future than the requests with earlier deadlines.

In case the deadline of the selected connection is further in time than $SB_{s,d}$, the connection is added to $R_{s,d}$ (Line 12) and its allocation will be postponed; in case the deadline is equal to $SB_{s,d}$, the request cannot be postponed, since its deadline will expire, and the request is blocked (Line 14).

In case a path is found the lightpath is established (Line 17). The sets of connections $R_{s,d}$ is postponed (Line 20) and the set of connection $Q_{s,d}$ is blocked (Line 21).

4 Numerical Evaluation

In order to evaluate the performance of the Elastic Batch Grooming algorithm (BG), simulation experiments were derived and the FlexGridSim [13] simulator was emplyed in this simulation. For each scenario, at least ten simulations were run; in each simulation 100,000 requests were generated. The load is increased in steps 25 erlangs for each simulation. Confidence intervals with 95% confidence level and maximum witdh of 4% were generated and they were ommitted from the graphs for beter visual interpretation. The USA (Figure 1) and the NSF (Figure 2) topologies were used. The NSF topology has the USA topology has 24 nodes and 43 links whereas the NSF tology has 16 nodes and 25 links. In the simulated elastic network, the spectrum was divided in 240 slots of 12,5GHz each. The mean arrival rate and the mean holding time are adjusted to simulated the desired load in erlangs.

Results given by the ElasticBatchGrooming algorithms were compared to those of algorithms which groom individual connections to access the effectiveness of grooming batches of requests. The Modified Shortest Path (MSP) algorithm employs a Dijkstra like algorithm which at each iteration computes the cost of path going through a neighbouring node if the nodes are connected by contiguous slots enough to satisfy the requested bandwidth. The Spectrum- Constraint Path Vector Searching (SCPVS) algorithm builds a tree to represent the potential paths. At every step, it adds a leave to the tree and computes the cost of this addition. Information about the paths is stored in an auxiliary data structure. Since it searches for all possible paths, it produces blocking probability lower than the ones given by the MSP [1].

The BG algorithm employs the SCPVS as its RSA algorithm. Therefore, the difference in results produced by these two algorithms can be interpreted as the effectiveness of adopting a batch scheduling approach. Moreover, the BG algorithm can adopt any RSA algorithm in Line 7. The generalization of the findings in this paper about the effectiveness of the batch approach can be investigated by the adoption of others RSA algorithms.

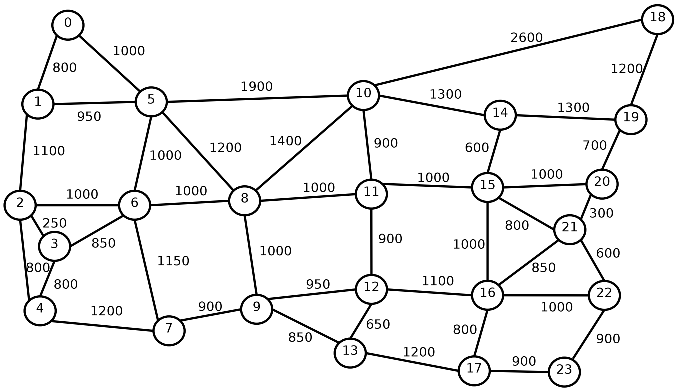

Figura 1: The USA topology

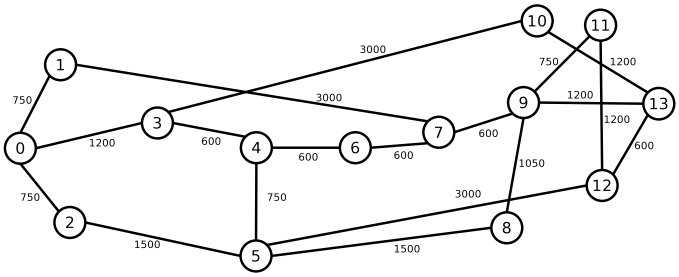

Figura 2: The NSF topology

Figure 3 shows the mean bandwidth blocking ratio as a function of the load for the topology USA. While the algorithms MSP and SCPVS start blocking requests under loads of 50 and 75 erlangs, respectively, BG starts blocking only under loads of 125 erlangs, having bandwidth blocking ratio under this load three orders of magnitude lower than those of MSP and SCPVS. Under loads of 200 erlangs the difference between the BBR produced by the BG algorithm and that given by the MSP algorithm is almost one order of magnitude and 60% when compared to that produced by SCPVS. Under high loads of 300 erlangs, the BBR produced by the MSP is twice the value produced by BG and the BBR given by SCPVS is 20% higher that the BBR given by BG. Figure 4 shows the mean bandwidth blocking ratio for the NSF topology. While blocking occurs under loads of 50 erlangs for the MSP and SCPVS algorithms, the BG algorithm produces blocking only under 75 erlangs and the difference in BBR to those of the other two algorithms is more than two orderes of magnitude. Moreover, produced by MSP and SCPVS is more than an order of magnitude higher than that given by the BG algorithm even under very high loads.

Such difference in blocking occurs due to the flexibility of postponing the allocation of requests that would be blocked by lack of resources at a certain time and that can be establish further in time when resources become available. Another benefit that leads to lower blocking probability is the allocation of batches as a whole that reduces the necessity of guard bands, leading to a more efficient spectrum usage.

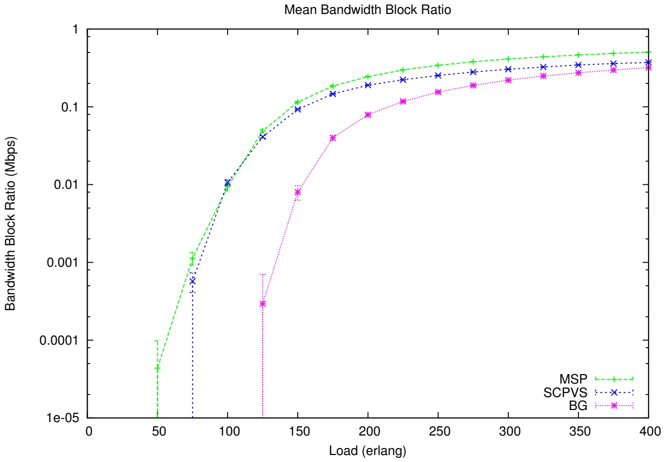

Figura 3: Bandwidth Blocking Ratio as a function of the load for the USA topology

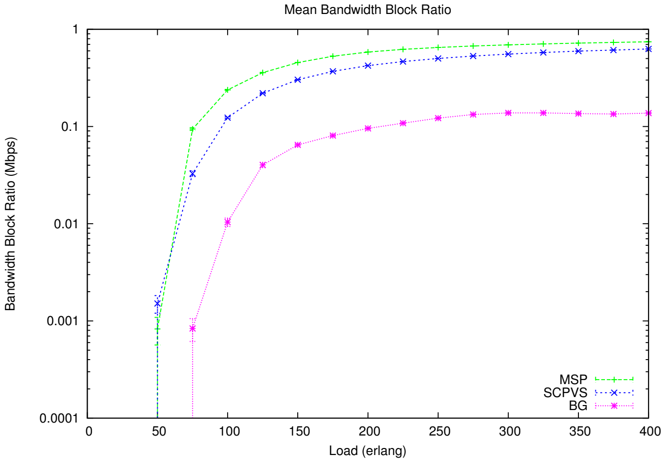

Figura 4: Bandwidth Blocking Ratio as a function of the load for the NSF topology

Figure 5 shows the mean number of transponders active in the network for the USA topology as a function of the load. Under low loads of 25 erlangs, the BG algorithm demands less 19% transponders than do the MSP and SCPVS algorithms. Such difference increases with the load, being the demand of the MSP and SCVPS algorithms very close. Under loads of 100 erlangs the two algorithms require more 200 transponder than does the BG algorithm and under loads of 250 erlangs this difference is of 600 transponders, twice the demand of the BG algorithm. Figure 6 shows the transponder usage for the NSF topology. The BG algorithm needs less 6% and 12% transponders than do the SCPVS and MSP algorithms, respectively. The NSF has lower node connectity than the USA topology and the transponder demand of the MSP and the SCPVS differ as a consequence of the lower number of paths established. Their demand becomes distinct around 100 loads and is up to 200 transponders under high loads of 400 erlangs. The demand of the BG algorithm is considerably lower for the NSF topology but the difference is not so high as that found for the USA topology. However, the different is still significant; under loads of 100 erlangs this difference is 200 transponders and under loads of 400 erlangs the difference is 400 and 600 transponders compared to the demands of the MSP and SCPVS algorithms, respectively. Such differences in transponder demands is due to the contiguous allocation of slots to support the requests in a batch, which demands only one transponder per batch rather than one transponder per request.

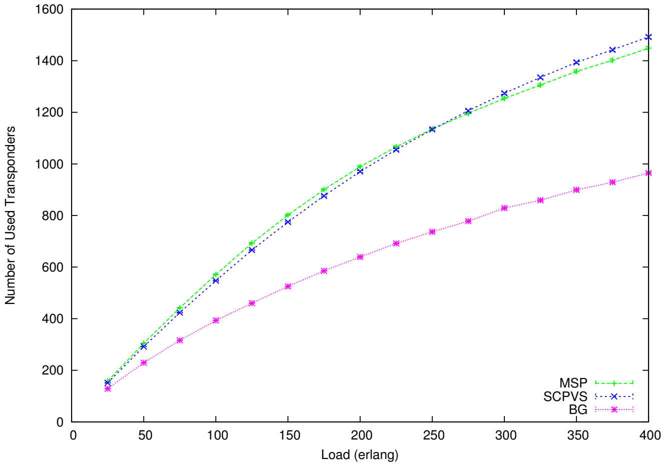

Figura 5: Mean number of active transponders in a network as a function of the load for the USA topology

Figures 7 and 8 show the average number of hops in the allocated lightpath for the USA and the NSF topologies. It can be concluded that the average number of hops per lightpath is not correlated to the bandwidth blocking ratio.

Figures 9 and 10 show the Jain Index of the blocking per source destination pair. The BG algorithm produces a Jain index value between that produced by the other two algorithms. BG produces an almost constant Jain Index regardless of the network load which does not happen with other two algorithms. In the USA topology, the SCPVS produce an unfair distribution of blocking among source destination pair. Although, the MSP starts blocking under loads of 50 erlangs, it reaches an almost fair distribution only under loads of 150 erlangs. For the NSF topology, the value of the Jain Index of Fairness increased due to the fact that blocking increased for all source destination pairs. The SCPVS still produces an unbalanced distribution of blocking and the MSP produces a balanced distribution only under a load of 50 erlangs higher than that under it started blocking requests. The difference between the Jain index produced by BG and SCPVS evinces clearly the advantage of balanced blocking when a batch approach is adopted.

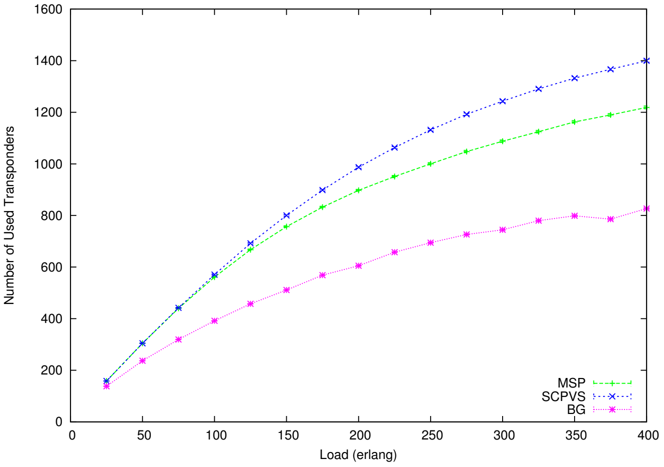

Figura 6: Mean number of active transponders in a network as a function of the load for the NSF topology

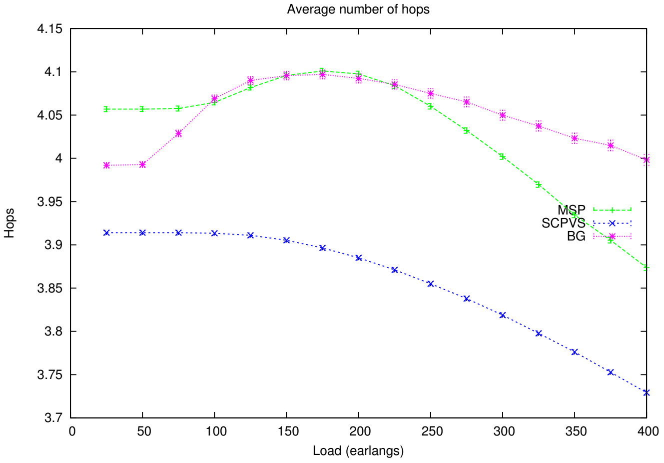

Figura 7: Average number of hops allocated per lightpath for the USA topology

5 Conclusion

This paper introduced a novel grooming algorithm for elastic optical networks which capitalize on the deadlines to finish the data transmission of requests. The algorithm, named Elastic Batch Grooming, aggregates requests for the same source destination pair and tries to allocate lightpaths for the batches. By doing so there is a significant decrease on the number of transponders demanded and an increase in the available capacity due to the need of a lower number guard bands. The proposed batch algorithm was compared to algorithms that allocate lightpaths to individual connections and results show a significant reduction in bandwidth blocking ratio, as well as in demand of transponders. The effectiveness of the batch grooming approach can be assessed by the difference between the results produced by

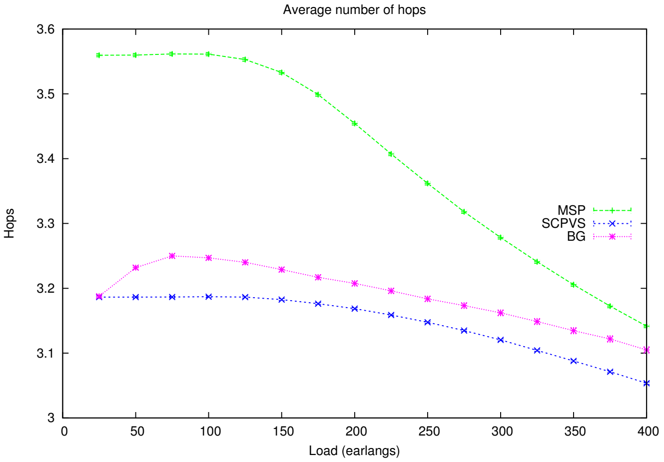

Figura 8: Average number of hops allocated per lightpath for the NSF topology

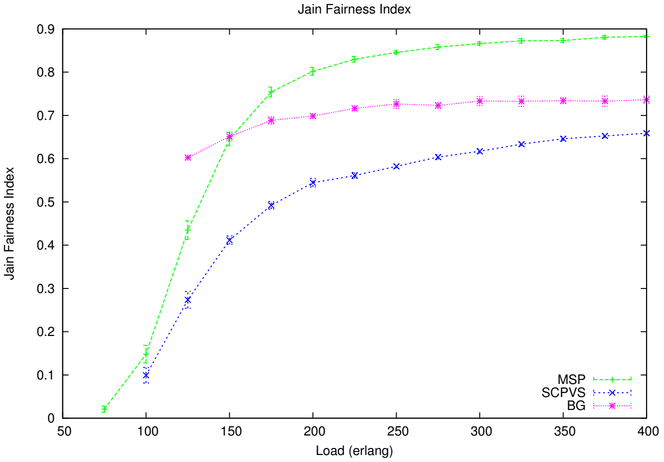

Figura 9: Jain Fairness Index as a function of the load for the USA topology

BG and SCPVS. The generalization of such advantages can be investigated by comparison the results of other RSA algorithms with those produced by BG employing other algorithms.

The proposed algorithm can be enhanced in different ways. For instance, when a connection cannot be served in a batch due to lack of network resources, instead of scheduling in a batch in the future, a lightpath to this connection can be allocated. The proposed algorithm demonstrated the clear advantage of adopting a batch approach for resource allocation in elastic networks which opens avenue for novel efficient batch grooming algorithms.

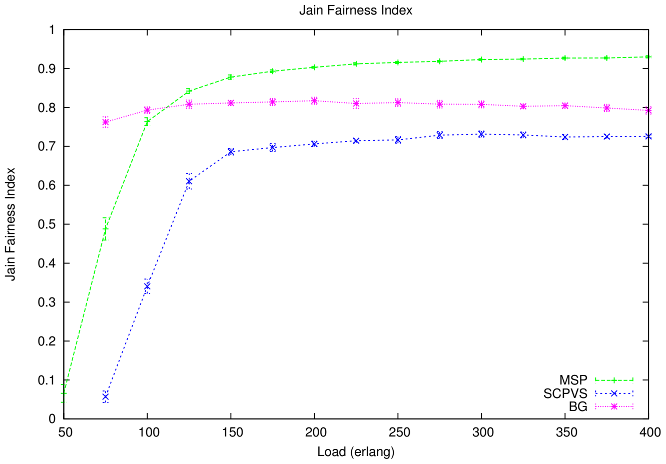

Figura 10: Jain Fairness Index as a function of the load for the NSF topology

6 Acknowledgements

This work was sponsored by State of Sao Paulo Research Foundation (FAPESP – grant 2013/01037-5), CNPq and INCT FOTONICOM.

Referências

[1] S. Zhang, C. U. Martel, and B. Mukherjee, "Dynamic traffic grooming in elastic optical networks.," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 1, pp. 4–12, 2013.

[2] M. Tornatore, A. Baruffaldi, H. Zhu, B. Mukherjee, and A. Pattavina, "Holding-time-aware dynamic traffic grooming," *IEEE J.Sel. A. Commun.*, vol. 26, pp. 28–35, Apr. 2008.

[3] J. de Santi, A. Drummond, N. L. S. da Fonseca, and A. Jukan, "Load balancing for holding-time-aware dynamic traffic grooming," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, Dec 2010.

[4] J. de Santi, A. Drummond, N. L. S. da Fonseca, X. Chen, and A. Jukan, "Leveraging multipath routing and traffic grooming for an efficient load balancing in optical networks," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 2989–2993, June 2012.

[5] B. Wang, T. Li, X. Luo, and Y. Fan, "Traffic grooming under a sliding scheduled traffic model in wdm optical networks," *IEEE BroadNets*, 2004.

[6] C. Cavdar, F. Buzluca, M. Tornatore, and B. Mukherjee, "Dynamic scheduling of survivable connections with delay tolerance in wdm networks," in *IEEE INFOCOM Workshops*, pp. 1–6, April 2009.

[7] D. Andrei, M. Tornatore, M. Batayneh, C. Martel, and B. Mukherjee, "Provisioning of deadline-driven requests with flexible transmission rates in wdm mesh networks," *Networking, IEEE/ACM Transactions on*, vol. 18, pp. 353–366, April 2010.

[8] A. Muhammad, C. Cavdar, L. Wosinska, and R. Forchheimer, "Effect of delay tolerance in wdm networks with differentiated services," in *Optical Fiber Communication Conference*, p. JWA056, Optical Society of America, 2011.

[9] W. Fawaz, I. Ouaiss, K. Chen, and H. Perros, "Deadline-based connection setup in wavelength-routed wdm networks," *Comput. Netw.*, vol. 54, pp. 1792–1804, Aug. 2010.

[10] L. Shen, X. Yang, A. Todimala, and B. Ramamurthy, "A two-phase approach for dynamic lightpath scheduling in wdm optical networks," in *ICC '07 - IEEE International Conference on Communications*, pp. 2412 –2417, june 2007.

[11] A. N. Patel and J. P. Jue, "Routing and scheduling for variable bandwidth advance reservation," *J. Opt. Commun. Netw.*, vol. 3, pp. 912–923, Dec 2011.

[12] J. de Santi, N. L. S. da Fonseca, and G. B. Figueiredo, "Algorithm for traffic grooming of batches of deadline-driven requests.," in *ICC*, pp. 2267–2271, IEEE, 2013.

[13] P. M. Moura and A. C. Drummond, "FlexGridSim: Flexible Grid Optical Network Simulator." http://www.lrc.ic.unicamp.br/FlexGridSim/.