Load Balancing for Holding-Time-Aware Dynamic Traffic Grooming

Juliana de Santi, André C. Drummond and Nelson L. S. da Fonseca
University of Campinas
Campinas, Brazil
Email: {santi,andred, nfonseca}@ic.unicamp.br

Admela Jukan
Technische Universität Carolo-Wilhelmina
Braunschweig, Germany
E-mail: jukan@ida.ing.tu-bs.de

***Abstract*— In this paper, a new algorithm for dynamic traffic grooming is introduced. It considers the holding-time of the connections and it aims at balancing the load among existing lightpaths to avoid the formation of bottlenecks and, consequently, high blocking probability values. Results indicate that it produces significantly lower blocking probabilities when compared to other holding-time-aware algorithm. Moreover, it promotes a fair distribution of blocking among source-destination pairs.**

I. Introduction

The huge amount of available bandwidth in the Wavelength Division Multiplexing (WDM) networks can be largely underutilized due to the disparity between the capacity of wavelengths and the bandwidth demand of IP traffic streams. Traffic grooming focus on narrowing this gap; for that, these techniques solve the problem of aggregating low-speed requests (subwavelengths) onto high speed optical channels to maximize resource utilization. Moreover, the deployment of WDM networks on metropolitan and local scales implies on the need for the consideration of dynamic traffic.

The availability of bandwidth has raised the expectation of users in relation to the attendance of their requests that are expected not only to be accepted, but also to have provided the Quality-of-Service(QoS) requirements of the applications they carry. The specification of QoS requirements in the Service Level Agreements (SLAs) makes available information on duration of the connections.

The holding-time of the connections is a fundamental piece of information for the development of strategies for increasing the efficiency of resource utilization. In [2], the impact of traffic predictability on network performance was demonstrated. In [3], the holding-time of connections was used to promote survivability of connections which are delay tolerant. In [4], it was proposed an approach to use the knowledge of the holding-times to groom dynamic traffic. Information on the duration of the connections was also employed on the provisioning of multicast requests [5].

The seminal algorithm proposed in [4], called Holding-Time-Aware (HTA), decreases the blocking probability when compared to traditional algorithms. However, this algorithm does not balance the traffic in the network, which can induce the creation of bottlenecks.

This paper introduces a dynamic traffic grooming algorithm for WDM mesh networks which is aware of the holding-time of the connections and that balances the network load with the aim of avoiding the creation of bottlenecks. Simulation results using the NSF, the USA and the Manhattan Street topologies indicate that the proposed algorithm significantly reduces the blocking of connection requests when compared to the HTA algorithm and yet promotes a fair distribution of resources among source destination pairs.

This paper is organized as follows. Section II describes the Holding-Time-Aware traffic grooming algorithm. Section III introduces a novel policy for dynamic traffic grooming employing traffic balancing. Section IV presents a numerical evaluation of the proposed algorithm. Finally, in Section V, conclusions are drawn.

II. Holding Time Aware Dynamic Traffic Grooming

Informations on the holding-time of connections can be used at the arrival of a connection request to determine the connection ending time [6]. Moreover, the lifetime of a lightpath can be determined by the largest ending time of a connection using this lightpath. Information on the ending time of a lightpath can, then, be employed to decide on the acceptance of connections, in order to maximize the utilization of lightpaths and to minimize the blocking probability of future demands [4], [7], [8].

To determine on which lightpath the connection will be groomed, the cost of the lightpaths are calculated considering the holding-times of the connections as well as the lifetimes of the lightpaths, as follow:

$$ht(p_i) = \begin{cases} p_i \times \epsilon & \text{if } H_i \geq h \\ p_i \times \epsilon + p_i \times \Delta_t & \text{if } H_i < h \\ p_i \times h & \text{if new path} \end{cases} \quad (1)$$

where: $ht(p_i)$ is the cost of i^{th} lightpath; h is the holding-time of the request; H_i is the lifetime of i^{th} lightpath; $\Delta_t = h - H_i$; $\epsilon = 10^{-5}$ is a constant value defined in [4]; p_i is the number of hops along the i^{th} lightpath.

Lightpaths already established tends to have lower costs than those to be created since the overlap between the lightpath lifetime and the incoming connection holding time reduces the cost of existing lightpaths [4]. The main idea of this approach

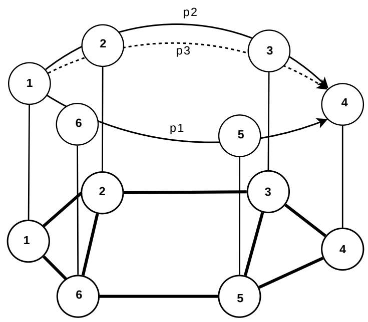

Fig. 1. Example topology: lightpath cost calculation (modified from [4])

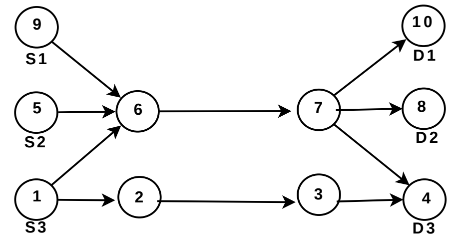

Fig. 2. Illustration of unbalanced network load

is to use the minimum number of new lightpaths as well as to minimize the enlargement of the lifetime of existing lightpaths.

Consider, as an example, the topology in Figure 1. An arriving request $d_1(1,4,2,30)$ requests 2Mbps of bandwidth during 30 seconds (holding-time) for the source-destination pair $(1-4)$. The paths p_1, p_2 and p_3 have, respectively, 4Mbps, 6Mbps and 1Gbps and they are candidates to support d_1. Only p_1 and p_2 are already established and their lifetime are, respectively, 10s and 20s. The request d_1 could be provisioned by grooming it on the lightpath p_1, which implies on extending its lifetime by 20s. The utilization cost of p_1 is given by the utilization cost of three links $1 \rightarrow 6$, $6 \rightarrow 5$ and $5 \rightarrow 4$ multiplied by the 20 additional seconds of d_1 ($3 \times 20 = 60$). Similarly, if p_2 is chosen, its lifetime needs to be extended by 10 seconds, thus, the utilization cost is the cost of the use of the links of p_2 ($1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 4$) multiplied by the additional time needed ($3 \times 10 = 30$). Moreover, p_3 can be created and the cost of its establishment using the links $1 \rightarrow 2$, $2 \rightarrow 3$ and $3 \rightarrow 4$ for 30s is ($3 \times 30 = 90$). Therefore, the lightpath which minimizes the cost of accepting the request d_1 is p_2.

This example illustrates that the overlap between the lightpath lifetime and the connection holding-time decreases the lightpath cost.

In line with that, lightpaths with close ending time will have higher cost since their lifetimes have small overlaps with the arriving request lifetimes.

III. Load Balancing of Holding Time Aware Dynamic Traffic

The holding-time-aware traffic grooming algorithm, presented in Section II, does not balance the traffic among existing lightpaths and its strategy always associate low cost to lightpaths with long lifetime. As a consequence, these lighthpaths tend to have their capacity exhausted.

To illustrate the consequences of unbalanced traffic distribution, let us consider the example illustrated in Figure 2. (S_1, D_1), (S_2, D_2) and (S_3, D_3) are source-destination pairs; the path $6 \rightarrow 7$ has one unit of bandwidth available and all the other paths have five units. At time t_0, a new request d_1(S_3,D_3,1,25) demands one unit of bandwidth during 25 seconds to the S_3-D_3 source-destination pair. The lightpaths p_1 (route 1, 6, 7, 4) and p_2 (route 1, 2, 3, 4) are already established, and their lifetimes are, respectively, 20s and 5s. Considering just the holding-time criterion to calculate the lightpath cost (Eq. 1), lightpath p_1 would be chosen to support the connection d_1 since it has the longest lifetime. This decision would exhaust the available capacity of the link $6 \rightarrow 7$ and, as a consequence, all incoming connections of the (S_1, D_1) and (S_2, D_2) source-destination pairs would be rejected. This happens since the link $6 \rightarrow 7$ is part of a critical path and its use should be considered as a scarce resource.

We propose an algorithm called Holding-Time-Aware-with-Traffic-Balancing (HTBalancing), which balances the incoming connections among existing lightpaths so that the blocking of future demands can be minimized as well as the fairness among source destination pairs can be improved.

To promote the load balancing of connections with know holding-time, the following cost function is used:

$$C(p_i) = (ht(p_i) \times \alpha) + \left(\frac{1}{bw(p_i)} \times \beta \right) \qquad (2)$$

were: $C(p_i)$ is the cost utilization of lightpath p_i; $ht(p_i)$ is cost given by holding-time cost function (Eq. 1); α determines the weight used in the holding-time cost function; bw is the available bandwidth in p_i; β determines the weight of bandwidth availability to $C(p_i)$

Considering the inverse of the available bandwidth in the cost function (second term of Equation 2) implies that lightpaths with small bandwidth availability will have high cost, so, its utilization is prevented in order to avoid the creation of bottleneck. On the contrary, a lightpath with large available bandwidth will have low cost, making it a potential good candidate to accommodate incoming requests.

The acquisition of information on bandwidth availability does not impose significant overhead when compared to the overhead of the HTA algorithm since the latter demands information on the lightpath lifetime.

The HTBalancind algorithm is presented in Figure 3. Whenever a new request arrives, the cost of utilizing the existing lightpaths is computed. To avoid the creation of unnecessary lightpaths induced by the availability of the whole bandwidth of the new lightpath, the bandwidth availability is accounted only for existing lightpaths. This results in using the equation

Algorithm HTBalancing

Require: Network graph $G = (V, E)$; demand $d_j(s, d, b, ht)$ requesting b bandwidth units during ht units of time between (s, d)

1:

Ensure: Feasible path between s and d to aggregate d_j

2:

3: **for all** (candidate lightpaths p_i) **do**

4:     **if** p_i is already **established then**

5:         $C(p_i) \leftarrow (ht(p_i) \times \alpha) + \left(\frac{1}{bw(p_i)} \times \beta\right)$ (Eq. 2)

6:     **else if** p_i is a **new** lightpath **then**

7:         $C(p_i) \leftarrow (ht(p_i) \times \alpha) + (1 \times \beta)$

8:     **end if**

9: **end for**

10: Apply Shortest-path to determine the lower cost lightpath

Fig. 3. Algorithm HTBalancing for lightpath cost computation

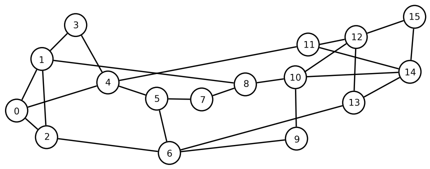

Fig. 4. NSF topology

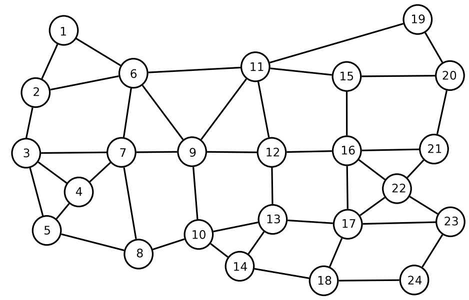

Fig. 5. USA topology

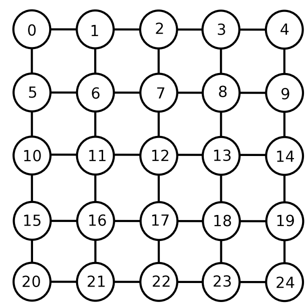

Fig. 6. Manhattan Street topology

in line 5 to the existing connections only and the equation in line 7 to those that can be potentially created.

The cost of each feasible optical lightpath is calculated and the labels of the edges representing the network are updated with the weight values $C(p_i)$. Then, traditional Shortest-path algorithm (line 10) is employed to determine the lightpath with lowest cost.

IV. Performance Evaluations

To assess the performance of the HTBalancing algorithm, simulations were conducted and results compared to those given by the HTA algorithm [4]. The RWA algorithm employed is a single-hop Fixed-Alternate Routing one with 5 alternative routes. The First-Fit wavelength assignment is also employed. The weights used to holding-time cost function and to bandwidth availability in Equation 2 were, respectively, $\alpha = 0.5$ and $\beta = 0.5$.

Simulations were performed considering the NSF topology, with 16 nodes and 25 bidirectional links (Fig. 4), the USA topology, with 24 nodes and 43 bidirectional links (Fig. 5) and the Manhattan Street (grid $5X5$) topology, with 25 nodes and 40 bidirectional links (Fig. 6). In these topologies, each fiber carries 16 wavelengths, each with bandwidth capacity of an OC-192 carrier (10 Gbps); each node is a multi-hop partial grooming node with 32 grooming port pairs (input,output) and no wavelength-conversion capability.

Connections arrive according to a Poisson process and their bandwidth demands are distributed according to the following probability distribution: $6/19$ for OC-3, OC-12 and OC-48; and $1/19$ for OC-192 carriers [9]. Connection requests are uniformly distributed among all pairs of nodes. The holding time follows a negative exponential distribution with a mean of one unit. The network load is given in Erlangs defined as the call arrival rate $\times$ call holding time $\times$ the calls bandwidth request normalized to the capacity of an OC-192 carrier.

The metric used to evaluate the algorithms is the bandwidth blocking rate (BBR), i.e., the percentage of the amount of blocked traffic over the total bandwidth requested during each simulation. Ten simulation runs were carried out for each point in the curves, each run involved 1 million requests. Confidence intervals with 95% confidence level were established.

BBR values for the NSF topology as a function of the load are presented in Figure 7. The BBR values produced by HTBalancing are considerably lower than those given by the HTA algorithm. As can be seen, the difference is about one order of magnitude for most of the loads. The BBR produced by HTBalancing under loads of 70 Erlangs is 93% lower than those produced by the HTA. Under higher loads, the differences between the BBR values decreases but that given by the HTA is still 46% higher under loads of 130 Erlangs. The lower connectivity of nodes in some paths in the NSF topology leads to the creation of bottlenecks, even under low loads. The central idea of the HTA algorithm is to minimize the amount of additional lifetime of existing lightpaths, tending to concentrate the traffic on few lighthpaths, which leads to

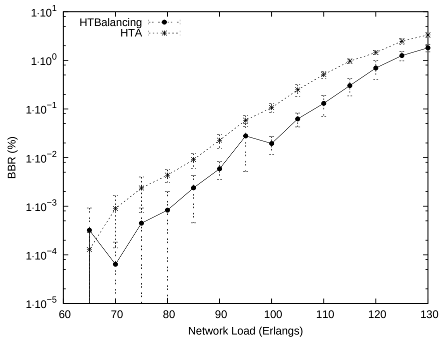

Fig. 7. BBR over the network load for NSF network

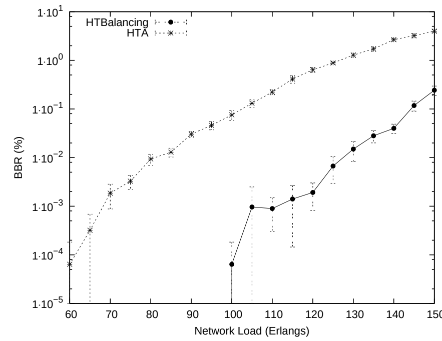

Fig. 9. BBR over the network load for USA network

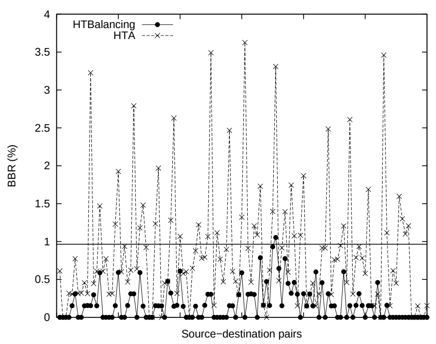

Fig. 8. BBR of each source-destination pair for a load of 115 Erlangs(NSF)

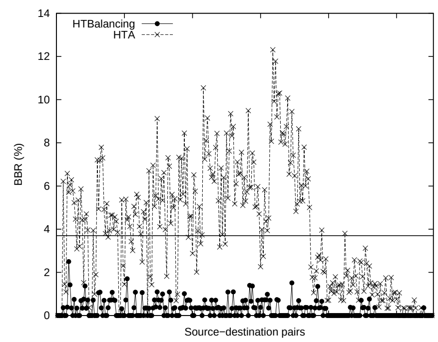

Fig. 10. BBR of each source-destination pair for a load of 150 Erlangs(USA)

bottlenecks, specially in topologies such as the NSF network. Conversely, the HTBalancing algorithm tries to distribute the load to avoid bottlenecks and, consequently, its BBR blocking rates are lower than those given by the HTA algorithm.

Figure 8 shows the per pair BBR distribution and the BBR mean value for a single simulation for a load of 115 Erlangs. For same pairs, the HTA algorithm generated blocking rates up to 3.7 times greater than the mean BBR value (0.96%) and up to 12 times greater than the HTBalancing BBR mean value (0.30%). This shows the importance of balancing the traffic to promote effective resource utilization.

Figure 9 shows the results for the USA network. In this scenario, the differences between the BBR given by HTBalancing and that given by the HTA algorithm are even larger than those for the NSF topology. Under loads lower than 100 Erlangs, the HTBalancing approach does not produce blocking, whereas the HTA algorithm blocks about 756 requests. Under loads of 100 Erlangs, the BBR produced by HTBalancing algorithm is 3 orders of magnitude lower than those produced by the HTA algorithm. The smallest difference between the BBR given by the two algorithms is under loads of 150 Erlangs, when the HTBalancing algorithm gives a BBR value 94% lower than that produced by the HTA algorithm. This difference is highly influenced by the USA topology characteristics, which has a high degree of node connectivity, providing various alternative paths which helps to avoid the creation of bottlenecks. Such topology favors even more the production of low BBR values by the HTBalancing due to its load balancing approach.

Figure 10 shows the per pair BBR distribution and the BBR mean value for a single simulation run for a load of 150 Erlangs for the USA topology. The algorithm HTA produces BBR values 3.1 times greater than its mean value (3.97%) and 51.29 times greater than the HTFBalancing mean value (0.24%). Balancing the load among several alternative lightpaths decreases even more the difference of BBR values among source destination pairs. Conversely, the HTA tends to concentrate the load in lightpaths with long lifetime increasing the blocking of requests for source-destination pairs that need those paths.

Figure 11 shows the BBR for the Manhattan network. The BBR values given by the HTBalancing are null for loads lower than 95 Erlangs, while the HTA rejects up to 156 requests. Under load of 95 Erlangs, the HTBalancing algorithm produces BBR values 1.5 orders of magnitude lower than those given by the HTA. Although the BBR difference decreases with the load increase, the HTA produces BBR values which are 91% higher than those given by the HTBalancing algorithm under loads of 150 Erlangs. The symmetry of the Manhattan Street topology enables the creation of multiple concurrent paths which favors the production of low blocking levels when the HTBalancing algorithm is employed.

Figure 12 presents the per pair BBR distribution and its

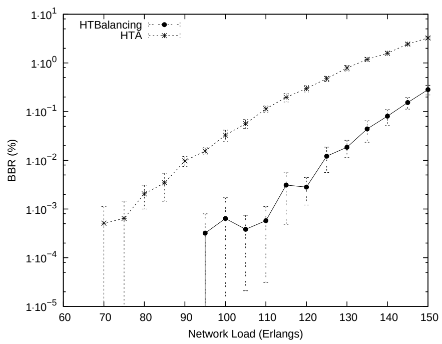

Fig. 11. BBR over the network load for Manhattan Street network

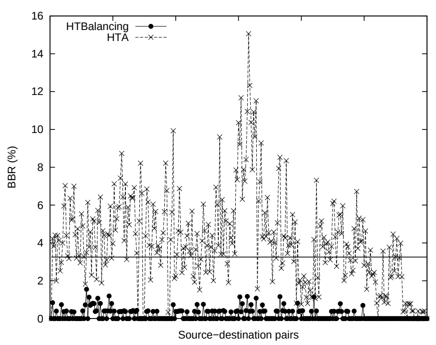

Fig. 12. BBR of each source-destination pair for a load of 150 Erlangs(Manhattan Street)

mean value for a single simulation for a load of 150 Erlangs. For some source destination pairs the blocking probability given by the HTA algorithm is up to 4.6 greater than its mean BBR value (3.25%) and up to 53.7 times greater than the mean BBR given by the HTBalancing algorithm (0.28%). The Manhattan network has even higher degree of connectivity than the other two topologies, so the HTBalancing algorithm is even more effective in avoiding network bottlenecks for this topology. It is important to note that the BBR values produced by the HTBalancing algorithm is distributed more uniformly among the source destination pairs than are those given by the HTA algorithm, evincing that the HTBalancing yields to higher degree of fairness.

V. Conclusion

The present paper introduced a novel dynamic traffic grooming algorithm (HTBalancing) that jointly employs knowledge of holding-times of the connections and the network bandwidth availability to balance the traffic, which avoids the creation of bottlenecks and, consequently, blocking. The BBR values produced by the HTBalancing algorithm are significantly lower than those given by the HTA algorithm. Such differences can be as large as three orders of magnitude. Furthermore, HTBalancing promotes fairness among source-destination pairs.

Acknowledgments

This work was partially sponsored by CNPq under support code 141246/2009-0 and by Deutsche Forschungsgemeinschaft (DFG) under support code JU2757/-1/1; and FAPESP under support code 2007/54867-4.

References

[1] S. S. Liu and L. F. Chen, "Deployment of carrier-grade bandwidth-on-demand services over optical transport networks: A verizon experience," in *National Fiber Optic Engineers Conference*. Optical Society of America, 2007, p. NThC3.

[2] E. Doumith and M. Gagnaire, "Impact of traffic predictability on wdm exc/oxc network performance," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 5, pp. 895–904, June 2007.

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

[4] M. Tornatore, A. Baruffaldi, H. Zhu, B. Mukherjee, and A. Pattavina, "Holding-time-aware dynamic traffic grooming," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 3, pp. 28–35, April 2008.

[5] A. Munir, S. Tanwir, and S. Zaidi, "Holding time aware dynamic bandwidth allocation algorithm for emerging bandwidth on demand multicast applications," in *HONET, 2009 6th Int. Symposium on*, 2009, pp. 16 –21.

[6] W. Fawaz, B. Daheb, O. Audouin, M. Du-Pond, and G. Pujolle, "Service level agreement and provisioning in optical networks," *Communications Magazine, IEEE*, vol. 42, no. 1, pp. 36–43, Jan 2004.

[7] M. Tornatore, A. Baruffaldi, H. Zhu, B. Mukherjee, and A. Pattavina, "Exploiting connection-holding time for an efficient dynamic traffic grooming," in *DRCN 2007. 6th Int. Workshop on*, Oct. 2007, pp. 1–8.

[8] ——, "Dynamic traffic grooming of subwavelength connections with known duration," in *OFC/NFOEC 2007. Conf. on*, March 2007, pp. 1–3.

[9] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in wdm mesh networks using a novel graph model," in *GLOBECOM '02. IEEE*, vol. 3, nov. 2002, pp. 2681 – 2685 vol.3.