Online Distributed Traffic Grooming on Path Networks

Jean-Claude Bermond, David Coudert, Joseph Peters

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Online Distributed Traffic Grooming on Path Networks

Jean-Claude Bermond — David Coudert — Joseph Peters

N° 6833

February 2009

Thème COM

rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6833--FR+ENG

Online Distributed Traffic Grooming on Path Networks

Jean-Claude Bermond[*†] , David Coudert[*†] , Joseph Peters[‡§]

Thème COM — Systèmes communicants
Projets Mascotte

Rapport de recherche n° 6833 — February 2009 — 14 pages

Abstract: The *grooming factor* C of a WDM optical network is the number of connections that can share the bandwidth of each wavelength and the process of grouping the requests that will share each wavelength is called *traffic grooming*. The goal of traffic grooming is either to reduce the transmission cost by reducing the number of wavelengths or to reduce the hardware cost by reducing the number of Add-Drop Multiplexors (ADM).

In this paper, we investigate traffic grooming for directed path networks with online connection requests and distributed routing algorithms. When connection requests are online, the *virtual topology* that results from the assignment of ADMs to wavelengths cannot be changed with each request. The design of efficient virtual topologies that minimize either the number of ADMs needed to satisfy any set of connection requests or the blocking of connection requests depends strongly on the routing algorithm. We show how to design the best possible virtual topologies, independently of the routing algorithm, when each node is equipped with the same number of ADMs, and we analyze the performance of distributed greedy routing algorithms.

Key-words: Traffic grooming, distributed algorithm, online traffic

This work has been partly funded by the European project IST FET AEOLUS and associated team RESEAUXCOM.

[*] MASCOTTE, INRIA, I3S, CNRS, Univ. Nice Sophia, Sophia Antipolis, France.
[†] firstname.lastname@sophia.inria.fr
[‡] School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada, V5A 1S6
[§] peters@sfu.ca

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Groupage de traffic en ligne et en distribué sur le chemin

Résumé : Le facteur de groupage C d'un réseau optique WDM est le nombre de connexions pouvant partager la bande passante d'une longueur d'onde. Le fait de pouvoir grouper les requêtes qui vont partager une longueur d'onde est appelé groupage de trafic. L'objectif du groupage de trafic est d'une part de reduire le coût de transmission en réduisant le nombre de longueurs d'ondes utilisées et d'autre part le coût des équipements en réduisant le nombre de multiplexeurs à insertions/extractions (ADM).

Dans ce rapport, nous étudions le groupage de trafic sur le chemin orienté avec du trafic en ligne et un algorithme de routage distribué. Lorsque les connexions sont en ligne, la topologie virtuelle construite en affectant les ADMs aux longueurs d'ondes n'est pas modifiée. La conception d'une topologie virtuelle performante qui minimize le nombre d'ADMs nécessaire pour soit satisfaire toute instance de trafic ou pour réduire la probabilité de blocage dépend fortement de l'algorithme de routage. Nous montrons comment concevoir la meilleure topologie virtuelle possible, indépendemment de l'algorithme de routage, lorsque chaque nœud est équippé avec le même nombre d'ADMs. Puis, nous analysons les performances d'un algorithme de routage distribué.

Mots-clés : Groupage de trafic, algorithme distribué, trafic en ligne

1 Introduction

In connection-oriented networks, such as *wavelength division multiplexing* (WDM) optical networks, *traffic grooming* is used to improve the usage of the bandwidth and components by combining *connections* (low speed traffic streams) onto high speed wavelengths. The goal is to minimize the network-wide electronic switching cost. Typically, nodes of the network are equipped with *add drop multiplexers* (ADMs) to insert and extract the traffic streams on a wavelength. One ADM is needed in a node for each wavelength to which traffic is added or dropped. The *grooming factor* C of a WDM network is the number of connections that can share the bandwidth of each wavelength, and an ADM is able to add up to C and drop up to C unitary traffic streams on a given wavelength. Thus, the traffic grooming problem is to minimize the total number of ADMs to be installed in a network to accommodate all of the traffic streams.

For a given fixed set of connection requests and single-hop routing (a connection uses a single wavelength from source to destination), the general traffic grooming problem of minimizing the total number of ADMs is NP-complete [CM00, APS07, SUZ07] and hard to approximate. The corresponding network design and routing problems have been considered by many authors. See the surveys [ML01, DR02, ZM03, BC06], the books [ZZM05, Som06, DKR08], and many articles [GRS00, Hu02, WCLF00, ZQ00].

The traffic grooming problem with multi-hop routing is also NP-complete [HDR06]. With such routing algorithm, a traffic stream can be switched from one wavelength to another at an intermediate node by using an optical-electronic-optical conversion. This requires one ADM for each wavelength at each intermediate node where switching can occur. Multi-hop routing allows further improvement of the usage of the bandwidth.

When the connection requests arrive online, the network design problem is different [KÖ6, CRSL08] because it is impossible to modify the assignment of ADMs to wavelengths for each new connection request. Therefore, the problem is to design an efficient *virtual topology* that minimizes the blocking rate by permanently assigning ADMs to wavelengths at nodes. There is a link in a virtual topology between two nodes for a particular wavelength if each node has an ADM associated with that wavelength.

In this paper, we investigate traffic grooming for directed path networks with online connection requests and distributed routing algorithms. Our objective is to design the best possible virtual topologies with the constraints that each node (i) is equipped with the same number of ADMs, (ii) is connected to wavelengths in a uniform way, (iii) is the source or destination of at most k connection requests and any traffic pattern that satisfies this constraint can be routed without blocking. We first give upper bounds on the size of the network, independently of the routing algorithm. Then, using the distributed greedy routing algorithm proposed in [CRSL08], we give optimal constructions that improve upon the constructions in [CRSL08].

1.1 Notation

P_N**:** unidirectional path with N nodes

C**:** grooming factor

k**:** maximum number of connection requests leaving (origianting) at a node and the maximum number of connectio requests arriving (terminating) at a node. A family of requests that satisfies the constraint that each node originates or terminates at most k requests is said to be *k-allowable.*

r**:** maximum in-degree and out-degree of the directed virtual topology

V**:** set of nodes of the physical and logical topologies. For convenience, we associate an integer in the range $[-p..q]$ to each node of the physical topology. We have $|V| = N = p + q + 1$.

$l_1, l_2, \ldots, l_r$**:** lengths of the links of the virtual topology, such that $l_j < l_{j+1}$. (Note that $l_j = j$ in [CRSL08].)

$S_1, S_2, \ldots, S_r$**:** sum $S_i = l_1 + l_2 + \ldots + l_i$ of the i smallest lengths of the links of the virtual topology

A**:** set of arcs of the virtual topology. We have $A = \{(i, i + l_j) \mid i \in [-p..q],\ j \in [1..r],\ i + l_j \leq q\}$, and $|A| = \sum_{j=1}^{r} j = r(r + 1)/2$.

We say that an arc (u, v) of the virtual topology *crosses 0* if $u \leq 0$ and $v > 0$. In other words, arc (u, v) of the virtual topology requires capacity C on arc $(0, 1)$ of the physical topology. We say that a virtual arc is *saturated* when it is involved in C connections and *available* otherwise.

1.2 Problems and examples

Problem 1 (General Problem) *Given $k \geq 1$, find the admissible values of N, C, $l_1, l_2, \ldots, l_r$ such that any k-allowable set of requests can be routed online in the virtual topology in a way that respects the traffic grooming constraint (at most C connections use the same arc of the virtual topology) and is independent of the routing algorithm.*

Example 2 *Let $k = 1$, $C = 2$, $r = 2$, $l_1 = 1$, and $l_2 = 4$. If $N \geq 6$ we cannot route the three requests $(0, 3)$, $(-1, 2)$, and $(-2, 1)$ as the only possible routes are, respectively, $(0, 1); (1, 2); (2, 3)$, $(-1, 0); (0, 1); (1, 2)$ and $(-2, -1); (-1, 0); (0, 1)$. This routing uses arc $(0, 1)$ three times, contradicting $C = 2$.*

Distributed Greedy Algorithm, `DGA` [CRSL08] . A new request (s,t) is routed online as follows. The current vertex v is initially the start vertex. At the current vertex, $v \neq t$, the available arc of maximum possible length is chosen. More precisely, if the distance from the current vertex v to t is $d(v,t) \geq l_r$ and the arc $(v, v+l_r)$ is available, then the request is routed from v to $v+l_r$ and the current vertex becomes $v+l_r$. Otherwise, the arcs $(v, v+l_{r-1}), (v, v+l_{r-2}), \ldots$ are tried until an available arc is found. If an available arc is not found, then the connection is blocked and the request cannot be routed.

Example 3 *Let $k=1$, $C=2$, $r=3$, and $l_1 = 1, l_2 = 3, l_3 = 6$. The following table shows how four requests are routed using* `DGA`:

Request		Route	Request		Route
$(-12,3)$	$\mapsto$	$(-12,-6); (-6,0); (0,3)$	$(-6,4)$	$\mapsto$	$(-6,0); (0,3); (3,4)$
$(-9,1)$	$\mapsto$	$(-9,-3); (-3,0); (0,1)$	$(-3,2)$	$\mapsto$	$(-3,0); (0,1); (1,2)$

If a fifth request $(0,5)$ arrives, it cannot be routed because the arcs $(0,1)$ and $(0,3)$ have already been used twice and are not available, and $d(0,5) < l_3 = 6$. Note that the network does have the capacity to accommodate all five requests and a different algorithm could find a successful routing.

Problem 4 (General Greedy Problem) *Given $k \geq 1$, find the admissible values of N, C, $l_1, l_2, \ldots, l_r$ such that any k-allowable set of requests can be routed by* `DGA` in the virtual topology.

To address Problems 1 and 4, we concentrate on the following problems:

Problem 5 *Given $k=1$ and an admissible set of values of the parameters C, and $l_1 < l_2 < \ldots < l_r$, find the maximum possible number of vertices $N_{max}(C; l_1, l_2, \ldots, l_r)$ that can route any k-allowable set of requests.*

Problem 6 *Given $k=1$, C and r, find the maximum possible number of vertices $N_{max}(C,r)$ that can route any k-allowable set of requests.*

Problem 7 *Given $k=1$, C and r, find the maximum possible number of vertices $N^{greedy}_{max}(C,r)$ that can route any k-allowable set of requests using* `DGA`.

2 Universal counter-examples

We now provide examples that are independent of the routing algorithm and that will serve as counter-examples.

Lemma 8 *If $l_1 > 1$, then $N < 2$ for any choice of the l_p, $2 \leq p \leq r$.*

If $l_1 > 1$, we cannot route the request $(0,1)$. In the remainder of this paper we assume that $l_1 = 1$.

Lemma 9 *If $l_2 > C+1$, then $N \leq 2C+1$ for any choice of the l_p, $3 \leq p \leq r$.*

Proof. Suppose that $N \geq 2C+2$ and consider the set of $C+1$ requests $(-i, -i+C+1)$ for $0 \leq i \leq C$. The length of each request is $C+1$ and $l_p > C+1$ for $2 \leq p \leq r$, so only arcs of length $l_1 = 1$ can be used. So, each request requires the arc $(0,1)$ and only C of the $C+1$ requests can be routed. Example 2 illustrates the case $C = 2$, $l_2 = 4$ and $N = 2(C+1) = 6$. □

Lemma 10 *If $l_r > CS_{r-1}+1$, then $N \leq 2CS_{r-1}+1$.*

Proof. The proof is similar to the proof of Lemma 9. Suppose that $N \geq 2CS_{r-1}+2$ and consider the set of $CS_{r-1}+1$ requests $(-i, -i+CS_{r-1}+1)$ for $0 \leq i \leq CS_{r-1}$. The length of the request is $CS_{r-1}+1$ and $l_r > CS_{r-1}+1$, so only arcs of length $l_1 = 1, l_2, \ldots, l_{r-1}$ can be used. There are only l_p arcs of length l_p for each $1 \leq p \leq r-1$, l_p, so at most CS_{r-1} of the $CS_{r-1}+1$ requests can be routed. □

Lemma 11 $N_{max}(C; 1, l_2) \leq 2C(l_2+1)+1$.

Proof. The proof is similar to the proof of Lemma 9. Suppose that $N \geq 2C(l_2+1)+2$ and consider the set of $C(l_2+1)+1$ requests $(-i, -i+C(l_2+1)+1)$ for $0 \leq i \leq C(l_2+1)$. Each of these requests uses an arc crossing 0, but there are at most $C(l_2+1)$ available arcs crossing 0, namely C times $(0,1)$ and C times the l_2 arcs $(-i, i+l_2)$ for $0 \leq i < l_2$. □

Lemma 12 $N_{max}(C; 1, l_2, \ldots, l_r) \leq 2CS_r+1$.

Proof. The proof is similar to the proof of Lemma 11. Suppose that $N \geq 2CS_r+2$ and consider the set of CS_r+1 requests $(-i, -i+CS_r+1)$ for $0 \leq i \leq CS_r$. Each of these requests uses an arc crossing 0, but there are at most CS_r available arcs crossing 0, namely C times the l_p arcs $(-i, i+l_p)$ for $0 \leq i < l_p$ for each l_p, $p = 1, 2, \ldots, r$, . □

3 Case $r = 2$

Theorem 13 *For $l_2 > C+1$, $N_{max}(C; 1, l_2) = 2C+1$.*

Proof. By Lemma 9 we have $N_{max}(C; 1, l_2) \leq 2C+1$. Let us now show that any counter-example has at least $2C+2$ nodes. Apply `DGA` to the set of requests of a counter-example. As it does not find a solution, that means that at some point of the algorithm we have a request R routed until a vertex we denote 0 and that we cannot furthermore route it (0 can be the source of R) and in particular the arc $(0,1)$ is no more available ; therefore the arc $(0,1)$ has been already used for routing C requests. But these C requests and the request

R have all a starting node $i \leq 0$ and a destination node $j \geq 1$. As all the sources and all the destinations are different, we have at least $2C + 2$ nodes. □

Theorem 14 *For $l_2 \leq C + 1$, $N_{max}(C; 1, l_2) = 2C(l_2 + 1) + 1$.*

Proof. Similar as for theorem 13. □

Theorem 15 $N_{max}(C, r = 2) = 2C^2 + 4C + 1$ *and this value is attained for* $l_2 = C + 1$

Proof. From Lemma 9 we know that if $l_2 > C + 1$, then $N_{max}(C; 1, l_2) \leq 2C + 1 < 2C^2 + 4C + 1$ and from Lemma 11 we know that if $l_2 \leq C + 1$, then $N_{max}(C; 1, l_2) \leq 2C(l_2 + 1) + 1 \leq 2C^2 + 4C + 1$. The maximum is attained for $l_2 = C + 1$. □

Note that for $C \geq 2$ this value is strictly better than the value of [CRSL08] where they had only $6C + 1$.

4 Case $r = 3$

Theorem 16 *When $l_2 \leq C+1$ and $l_3 \leq 2C+1$, then $N_{max}(C; 1, l_2, l_3) = 2C(1+l_2+l_3)+1$*

Theorem 17 $N^{greedy}_{max}(C, r = 3) = 6C^2 + 6C + 1$ *and this value is attained for $l_2 = C + 1$ and $l_3 = 2C + 1$.*

Proof.

- If $l_2 > C + 1$, then by Lemma 9 we have $N \leq 2C + 1$.
- If $l_2 \leq C+1$ and $l_3 \leq 2C+1$, then by Theorem 16, $N \leq 2C(1+l_2+l_3)+1 \leq 6C^2+6C+1$. The value is attained for $l_2 = C + 1$ and $l_3 = 2C + 1$.
- If $l_2 \leq C + 1$ and $l_3 > C(1 + l_2) + 1$, then by Lemma 10, $N \leq 2C(1 + l_2) + 1 \leq 2C^2 + 4C + 1 < 6C^2 + 6C + 1$.
- If $l_2 \leq C + 1$ and $2C + 1 < l_3 \leq C(1 + l_2) + 1$, then the result follows from the counter-examples involved in Proposition 18.

□

Proposition 18 *Let $l_2 \leq C + 1$ and $2C + 1 < l_3 \leq C(1 + l_2) + 1$, then there exist an $N < 6C^2 + 6C + 1$ and a set of requests $\mathscr{R}$ that cannot be routed by* `DGA`.

Proof.

Idea of the proof: We will construct a set of requests containing $2C$ requests, each with a negative sender and a positive receiver in the set $\{1, 2, \ldots, l_3 - 1\}$ -$\{r_0\}$, such that when we apply DGA, these $2C$ requests are forced to transit in node 0. As the distance from 0 to the receiver is strictly less than l_3, these $2C$ requests use to reach their destination the $2C$ arcs $(0, 1)$ and $(0, l_2)$ and so it will be impossible to route the request $(0, r_0)$ if $r_0 < l_3$. We will choose $r_0 = l_2 - 1$, except for the case $l_3 \leq 6C + 3$ where will choose $r_0 = 2C + 1$. We will first put small requests forced to use arcs of length 1 and define the other requests, dealing first with the special case $l_3 \leq 6C + 3$.

I. Requests of first type: First we have the $l_2 - 2$ requests $(-i, -i + l_2 - 1)$ for $1 \leq i \leq l_2 - 1$. All these requests are of length $l_2 - 1$ and so use only arcs of length 1 and therefore transit in node 0.

II. Requests of second type:

II.a. *Special case $l_3 \leq 6C + 3$:*

We add the requests $R_d = (-(l_2 + dl_3), l_2 - 1 + d)$ for $0 \leq d \leq C + 1 - l_2$ and $R'_d = (dl_3, C + 1 + d)$ for $0 \leq d \leq C - 1$. If we apply DGA, the requests R_d and R'_d use first arcs of length l_3 and each arc carries at mots C requests; so they arrive respectively in vertex $-l_2$ and 0. For the requests arriving in $-l_2$, the distance to their receiver is at most $l_2 + C$ as the largest receiver is C; therefore they cannot use an arc of length l_3 as $l_3 > 2C + 1 \geq l_2 + C$. So we have exactly $2C$ requests transiting in 0, whose receiver is at most $2C < l_3$ and which use all the C arcs of length 1 and all the C arcs of length l_2. Then it is impossible to route the request $(0, 2C + 1)$.

II.b. *Special case $l_3 > 6C + 3$:*

The requests of the second type have a sender $s_{a,b,d} < 0$ and a receiver $r_{a,b,d} > 0$. Senders are of the following form :

$s_{a,b,d} = v_{a,b,d} = -(dl_3 + bl_2 + a)$, where $0 \leq a \leq \alpha - 1$, $0 \leq b \leq \beta - 1$, and $0 \leq d \leq \delta - 1$.

In order all these nodes are different, we suppose

$$\alpha \leq l_2 \quad ; \quad \beta l_2 \leq l_3 \tag{1}$$

We do not have the triple $a = b = d = 0$ (the sender 0 being reserved to the last request $(0, l_2 - 1)$), and some triples with $b = \beta - 1$ (see bellow). In fact, we will miss for $b = \beta - 1$ and any a exactly g triples. More precisely we will have only the $s_{a,\beta-1,d}$, with $0 \leq d \leq \delta - 1 - g$.

We will also not use h triples with $b = 0$. More precisely we will have only the $s_{a,0,d}$ with $0 \leq d \leq \delta - 1 - h_a$ where $h_a \leq \delta - 1$, $h_{\alpha-1} \geq h_{\alpha-2} \geq \ldots \geq h_0$ and $\Sigma_a h_a = h$.

In contrary, when $h_0 = 0$, we might add the sender $s_{0,0,\delta} = -\delta l_3$.

All the parameters $\alpha, \beta, \delta, g, h$ will be defined after in order to force the requests to transit in 0 when using the greedy algorithm and to bound the number of vertices N.

Following the greedy algorithm, if a request is arrived in $v_{a,b,d}$ with $d > 0$, we route this request to $v_{a,b,d-1}$ via the arc of length l_3 if that is possible. Therefore a request starting in

$s_{a,b,d}$ is routed till $v_{a,b,0} = -(bl_2 + a)$, if no arc of length l_3 is used more than C times. The arc the more loaded is the arc $(v_{a,b,1}, v_{a,b,0})$ loaded $\delta - 1$ times except for the arc $(v_{0,0,1}, v_{0,0,0})$ loaded δ times if the sender $s_{0,0,\delta}$ is used. So, no arc of length l_3 is used more than C times if :

$$\delta \leq C \tag{2}$$

Now we route the requests arrived in the nodes $v_{a,b,0} = -(bl_2 + a)$ with $a \geq 0$. We want to force them to arrive in the node $v_{a,0,0} = -a$ by using uniquely arcs of length l_2. For that we need first that the distance from $v_{a,b,0}$ to the receiver is strictly less than l_3.That is the following inequalities (3) should be satisfied for every triple a, b, d :

$$bl_2 + a + r_{a,b,d} < l_3 \tag{3}$$

We should also insure that the arcs of length l_2 are available. To realize that (and also for other reasons), we will do in sort that exactly C requests arrive in $v_{a,1,0} = -(l_2 + a)$. Therefore, C requests use the arc $(-l_2 - a, -a)$ and less than C requests use the arcs $(v_{a,b,0}, v_{a,b-1,0})$ for $b > 1$. In the node $v_{a,1,0} = -(l_2 + a)$, arrive all the requests originating at the $s_{a,b,d}$ with $1 \leq b \leq \beta - 1$. For $1 \leq b \leq \beta - 2$ we have δ such requests as $0 \leq d \leq \delta - 1$. For $b = \beta - 1$ we will have $\delta - g$ such requests as $0 \leq d \leq \delta - 1 - g$. So we choose β and g such that the equation (4) is satisfied.

$$(\beta - 1)\delta - g = C \quad ; \quad 0 \leq g < \delta \tag{4}$$

At that point we have exactly C requests arrived in $v_{a,1,0} = -(l_2 + a)$. which are routed to node $-a$. Furthermore,if $a > 0$, in node $-a$ arrive directly $\delta - 1 - h_a$ requests from the $s_{a,0,d}$ with $0 \leq d \leq \delta - 1 - h_a$. In each node $-a$, with $a > 0$, C requests will be routed via the arc of length l_2 to $-a + l_2$. The remaining requests in number $\delta - 1 - h_a$ are routed to $-a + 1$ via the arc of length $l_1 = 1$ and so on till node 0.

In summary we have on each arc $(-a, -a + l_2)$, with $1 \leq a \leq \alpha - 1$, C requests and furthermore, in node 0 are arriving

- C requests originating at the $s_{0,b,d}$, with $b > 0$, and transiting via the vertex $v_{0,1,0} = -l_2$
- plus $\Sigma_a(\delta - h_a) = \alpha\delta - h$ requests from the other $s_{a,b,d}$ (this includes the requests with $a = b = 0$).
- plus the $l_2 - 2$ requests $(-i, -i + l_2 - 1)$ of the first type.

To insure there are exactly $2C$ requests arriving in node 0 we need to have the equation (5) which defines h:

$$l_2 - 2 + \alpha\delta - h = C \tag{5}$$

Note that altogether we have $(\alpha-1)C+2C+1=(\alpha+1)C+1$ requests (the $+1$ comes from the request $(0,l_2-1)$. Therefore we will take the values of the $r_{a,b,d}$ in the interval $[l_2,(\alpha+1)C]$. Indeed recall that among the positive values we already used the values from 1 to l_2-2 for the requests of first type and we reserved l_2-1 for the request $(0,l_2-1)$. In particular the maximum of the $r_{a,b,d}$ is $(\alpha+1)C+1$. The smallest value of the $s_{a,b,d}$ is obtained for $s_{0,0,\delta}=-\delta l_3$ if it exists. So altogether we obtain a bound for the number of vertices

$$N \leq \delta l_3 + (\alpha+1)C + 2 \tag{6}$$

Now we will fix the values of the $r_{a,b,d}$ and see what are the conditions in order inequalities (3) are satisfied. The idea is give to the $r_{a,b,d}$ small values when bl_2+a is large (in view of having the smallest sum $bl_2+a+r_{a,b,d}$ in inequality 3). The values are given in increasing order from l_2 to $(\alpha+1)C+1$ in the opposite order of variation of the bl_2+a. More precisely, for the $\delta-g$ requests originating in $s_{\alpha-1,\beta-1,d}$ the values of $r_{\alpha-1,\beta-1,d}$ are in the interval $[l_2,l_2-1+\delta-g]$; then for the $\delta-g$ requests originating in $s_{\alpha-2,\beta-1,d}$ the values of their destinations are in the interval $[l_2+\delta-g,l_2-1+2(\delta-g)]$ and so on for the $\delta-g$ requests originating in $s_{a,\beta-1,d}$ the values of $r_{a,\beta-1,d}$ are in the interval $[l_2+(\alpha-1-a)(\delta-g),l_2-1+(\alpha-a)(\delta-g)]$. In particular the maximum value of $s_{0,\beta-1,d}$ is $l_2-1+\alpha(\delta-g)$. For the values $0<b<\beta-1$, there are δ requests for given a and b and the intervals are translated by δ. Therefore, for the δ requests originating in $s_{a,b,d}$, with $0<b<\beta-1$, the values of $r_{a,b,d}$ are in the interval of length δ ending in $l_2-1+\alpha(\delta-g)+\alpha(\beta-b-2)\delta+(\alpha-a)\delta$. Note that for a given b when a decreases by 1 , then bl_2+a decreases by 1 and the maximum value of $r_{a,b,d}$ increases by $\delta-g\geq 1$ if $b=\beta-1$ or $\delta\geq 1$ if $b<\beta-1$. So for a given $1\leq b\leq\beta-1$, the maximum of the right part of inequality (3) $bl_2+a+r_{a,b,d}$ is attained for $a=0$ and its value is $M_b=bl_2+l_2-1+\alpha(\delta-g)+\alpha(\beta-1-b)\delta$.

Note that :

$$b\geq 2, \qquad M_{b-1}-M_b=\alpha\delta-l_2 \tag{7}$$

For $b=0$, the maximum is also attained for $a=0$, as all the existing requests $s_{a,0,d}$ go to 0 and as $h_{\alpha-1}\geq h_{\alpha-2}\geq\ldots\geq h_0$. In that case the maximum is $M_0=(\alpha+1)C+1$ (the number of requests).

Note that there are $\alpha\delta-h$ requests with $b=0$ and so the maximum of $r_{0,1,d}$ is $(\alpha+1)C+1-(\alpha\delta-h)$ $M_1=l_2+(\alpha+1)C+1-(\alpha\delta-h)$ and using equation (5) we get:

$$M_0-M_1=\alpha\delta-h-l_2=C+2-2l_2 \tag{8}$$

Now to finish we deal with two cases according the values of l_2.

`Case 1:` $l_2\leq\frac{C+2}{2}$.

The idea for that case is that the maximum will be M_0 and so in order that all the inequalities (3) are satisfied we should have $l_3 > M_0 = (\alpha+1)C+1$. More precisely, let us choose α as follows:

$$(\alpha+1)C+1 < l_3 \leq (\alpha+2)C+1 \tag{9}$$

As $l_3 > 2C+1$, then $\alpha \geq 1$.
As $l_3 \leq C(1+l_2)+1$, then $\alpha \leq l_2$
Now we choose the smallest δ such that $\alpha\delta \geq C+2-l_2$; that is

$$\alpha(\delta-1) < C+2-l_2 \leq \alpha\delta \tag{10}$$

So equation (5) is satisfied and as $l_2 \leq \frac{c+2}{2}$,

$$\alpha\delta \geq C+2-l_2 \geq \frac{c+2}{2} \geq l_2. \tag{11}$$

Therefore by equation (7), for $b > 1$, $M_{b-1} - M_b = \alpha\delta - l_2 \geq 0$ and so $M_1 \geq M_b$, for $b > 1$. Also by equation (8) $M_0 - M_1 = \alpha\delta - h - l_2 = C+2-2l_2 \geq 0$. So the maximum is M_0, and as $l_3 > M_0 = (\alpha+1)C+1$ by (9) all the inequalities (3) are satisfied
As $l_2 \geq 2$ equation (10) implies $\delta - 1 < C$ and so $\delta \leq C$ (inequality (2)).
Now we choose β in order to satisfy equation (4) that is

$$\delta(\beta-2) < C \leq \delta(\beta-1) \tag{12}$$

Using inequality (11, 12, 9) we get $(\beta-2)l_2 \leq (\beta-2)\alpha\delta < \alpha C \leq l_3 - (C+2)$ So $\beta l_2 = (\beta-2)l_2 + 2l_2 \leq l_3 - (C+3) + 2l_2$. But we are in the case $l_2 \leq \frac{c+2}{2}$ and so $\beta l_2 \leq l_3$ implying the last inequality of (1).

So, with these choices, all the conditions are satisfied. By inequality (6), $N \leq \delta l_3 + (\alpha+1)C+2$.
By inequality (9) $N \leq \delta(\alpha+2)C+\delta+(\alpha+1)C+2 = \alpha(\delta-1)C+\alpha C+2\delta C+(\alpha+1)C+2$. Then by inequality (10) $\alpha(\delta-1) \leq C+1-l_2$ and so $N \leq C(C+1-l_2+2\alpha+2\delta+1)+\delta+2$
Finally $\alpha \leq l_2 \leq C+1$ and $\delta \leq C$ imply $N \leq C(C+1+C+1+2C+1)+C+2 \leq 4C^2+4C+2$ proving the theorem in that case.

`Case 2:` $l_2 > \frac{C+2}{2}$.
We first choose β such that:

$$(\beta+3)l_2 - 3 < l_3 \leq (\beta+4)l_2 - 3 \tag{13}$$

So the second inequality of (1) is satisfied. Furthermore, as $l_3 > 6C+3$ (we deal with the other case in the special case), then $\beta \geq 3$. Now we choose δ such that equation (4) is satisfied

$$(\beta-1)(\delta-1) < C \leq (\beta-1)\delta \tag{14}$$

The first part of (14) and $\beta \geq 3$ imply $C > (\beta-1)(\delta-1) \geq 2\delta - 2$ implying a stronger inequality than (2):

$$\delta < \frac{C+2}{2} \tag{15}$$

Finally, we choose α such that equation (5) is satisfied.

$$(\alpha-1)\delta < C + 2 - l_2 \leq \alpha\delta \tag{16}$$

As $l_2 > \frac{C+2}{2}$, then $\alpha \leq l_2$ and the first part of (1) is satisfied. Now it remains to show that all the inequalities (3) are satisfied. Equation (8) and $l_2 > \frac{C+2}{2}$ imply that $M_0 < M_1$. Equations (7) imply that the maximum of the M_b is either M_1 or $M_{\beta-1}$. So it suffices to verify that we have both $l_3 > M_{\beta-1}$ and $l_3 > M_1$.

We have $M_{\beta-1} = (\beta+1)l_2 - 1 + \alpha(\delta - g) \geq (\beta+1)l_2 - 1 + \alpha\delta$. Using the first part of (16), we get $M_{\beta-1} \leq (\beta+1)l_2 - 1 + C + 1 - l_2 + \delta$. As $C + 3 \leq 2l_2$ and by (15 we obtain $M_{\beta-1} \leq (\beta+3)l_2 - 4$. Therefore, by (13) $M_{\beta-1} < l_3$.

We have $M_1 = (\alpha+1)C + 1 + 2l_2 - (C+2)$. Using the first part of (16), we get:

$M_1\delta < (C+2-l_2)C + 2\delta C + \delta(1 + 2l_2 - (C+2)) \leq (C+2-l_2)C + 2\delta l_2 + \delta(C-1)$.

On the other side, by the first part of (13) $l_3\delta > (\beta+3)\delta l_2 - 3\delta$. Then, by the second part of (14)

$l_3\delta > Cl_2 + 4\delta L_2 - 3\delta$.

Finally, $l_2 > \frac{C+2}{2}$ implies $Cl_2 > (C+2-l_2)C$ and $2\delta l_2 - 3\delta > \delta(C-1)$ and so $l_3\delta > M_1\delta$ that is $l_3 > M_1$.

So, all the conditions are satisfied for this choice of values. Let us now compute N. By (6) $N \leq \delta l_3 + (\alpha+1)C + 2$. As $l_3 > M_1 > M_0 = (\alpha+1)C + 1$, we get $N \leq (\delta+1)l_3$.

By the second part of (13), we get $N \leq (\delta+1)((\beta+4)l_2 - 3) \leq (\delta+1)(\beta-1)l_2 + 2(\beta+4)l_2 + 5(\delta-1)l_2 - 3(\delta+1)$.

The inequality $l_3 \leq C(1+l_2)+1$ of the hypothesis of the theorem and first part of (13) give $(\beta+3)l_2 - 3 < l_3 \leq C(1+l_2)+1$ that is as $l_2 \leq C+1 : (\beta+3) \leq C+1$. Using that, $\delta < \frac{C+2}{2}$ (15) and the first inequality of (14) we obtain

$N \leq Cl_2 + 2(C+2)l_2 + 5\frac{C-3}{2}l_2 - 3\frac{C+3}{2} = \frac{11C^2}{2} + \frac{C}{2} - 10$ implying the theorem. □

5 Discussion and future work

In [CRSL08], we have $N \leq \frac{C}{k}r(r+1)$ using $r(r+1)/2$ wavelengths. So in some sense, adding one wavelength allows to add $\frac{2C}{k}$ nodes. In this work, we have $N \leq 2CS_r + 1$ using $S_r = 1 + l_2 + \cdots + l_r$ wavelengths. So, each extra wavelength contribute to $2C$ nodes. So the usage of wavelength is more efficient in our work than in [CRSL08].

Our results are independent of the routing algorithm. It would be interesting to refine them for specific distributed routing algorithms.

References

[APS07] O. Amini, S. Pérennes, and I. Sau Valls. Hardness and approximation of traffic grooming. In *The 18th International Symposium on Algorithms and Computation (ISAAC 2007)*, Sendai, Japan, December 2007.

[BC06] J-C. Bermond and D. Coudert. *Handbook of Combinatorial Designs (2nd edition)*, volume 42 of *Discrete mathematics and Applications*, chapter VI.27, Grooming, pages 494–496. Chapman & Hall- CRC Press, editors C.J. Colbourn and J.H. Dinitz, November 2006.

[CM00] A. L. Chiu and E. H. Modiano. Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks. *IEEE/OSA Journal of Lightwave Technology*, 18(1):2–12, January 2000.

[CRSL08] R.J. Crouser, B. Rice, A. Sampson, and R. Libeskind-Hadas. On-line distributed traffic grooming. In *IEEE International Conference on Communication (ICC)*, pages 5239 – 5246, Beijing, May 2008.

[DKR08] Rudra Dutta, Ahmed E Kamal, and George N Rouskas, editors. *Traffic Grooming for Optical Networks: Foundations, Techniques and Frontiers*. Optical Networks. Springer, August 2008.

[DR02] R. Dutta and N. Rouskas. Traffic grooming in WDM networks: Past and future. *IEEE Network*, 16(6):46–56, November/December 2002.

[GRS00] O. Gerstel, R. Ramaswani, and G. Sasaki. Cost-effective traffic grooming in WDM rings. *IEEE/ACM Transactions on Networking*, 8(5):618–630, 2000.

[HDR06] S. Huang, R. Dutta, and G.N. Rouskas. Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms. *IEEE Journal on Selected Areas in Communications*, 24(4):66–82, April 2006.

[Hu02] J.Q. Hu. Optimal traffic grooming for wavelength-division-multiplexing rings with all-to-all uniform traffic. *OSA Journal of Optical Networks*, 1(1):32–42, 2002.

[K0̈6] Martin Köhn. A new efficient online-optimization approach for SDH/SONET-WDM multi layer networks. In *OFC*, Anaheim/CA, USA, March 2006.

[ML01] E. Modiano and P. Lin. Traffic grooming in WDM networks. *IEEE Communications Magazine*, 39(7):124–129, July 2001.

[Som06] Arun Somani. *Survivability & Traffic Grooming in WDM Optical Networks*. Cambridge Press, January 2006.

[SUZ07] M. Shalom, W. Unger, and S. Zaks. On the complexity of the traffic grooming problem in optical networks. In *4th International Conference on Fun With Algorithms, Castiglioncello (LI), Tuscany, Italy*, June 2007.

[WCLF00] P-J. Wan, G. Calinescu, L. Liu, and O. Frieder. Grooming of arbitrary traffic in SONET/WDM BLSRs. *IEEE Journal of Selected Areas in Communications*, 18(10):1995–2003, October 2000.

[ZM03] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks: Architectures and challenges. *Optical Networks Magazine*, 4(2):55–64, March/April 2003.

[ZQ00] X. Zhang and C. Qiao. An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings. *IEEE/ACM Transactions on Networking*, 8(5):608–617, October 2000.

[ZZM05] Keyao Zhu, Hongyue Zhu, and Biswanath Mukherjee. *Traffic Grooming in Optical WDM Mesh Networks*. Springer, 2005.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399