Dynamic Traffic Grooming in Optical Burst-Switched Networks

Farid Farahmand[‡], Qiong Zhang[†], and Jason P. Jue[†]
[‡]Department of Electrical Engineering
[†]Department of Computer Science
The University of Texas at Dallas, Richardson, TX 75083
email: {ffarid, qzhang77, jjue}@utdallas.edu

***Abstract*— In this paper we address the problem of data burst grooming in optical burst-switched (OBS) networks. In OBS networks IP packets with the same destination are assembled into larger packets called *data bursts*. Depending on the core node's switching technology, data bursts are required to have a minimum length. On the other hand, each IP packet in a burst has a time delay constraint, called *maximum end-to-end delay*, which determines the upper time limit before which the packet must reach its destination. Thus, a data burst cannot wait indefinitely until sufficient number of IP packets are assembled and the minimum burst length requirement is met. In order to satisfy the packet maximum end-to-end delay requirement, many bursts will be *timed out* and released before they reach the minimum length requirement. Under such circumstances, padding overhead must be added to these short bursts, called *sub-bursts*. Excessive padding results in high overhead and high data burst blocking probability. One approach to minimize the amount of padding overhead, while maintaining the end-to-end delay requirement of IP packets, is to *groom* multiple sub-bursts together. That is, sub-bursts with different destinations are aggregated together at the edge node and transmitted as a single burst until they are separated at some downstream node. In this paper we present an edge node architecture enabling burst grooming capability. We also develop two basic grooming approaches, namely No-routing-overhead *(NoRO)* and Minimum-total-overhead *(MinTO)*. Through a comprehensive simulation study we show that, in general, our proposed grooming algorithms can significantly improve the performance compared to the case of no grooming. However, careful considerations must be given to network loading condition and the number of sub-bursts allowed to be groomed together. We show that although simple greedy algorithms can reduce network overhead, they may alter the traffic characteristics and increase its burstiness, resulting in high packet blocking probability.**

***Index Terms*— Burst assembly, dynamic traffic, edge node architecture, grooming, optical burst switching, padding overhead, routing overhead.**

I. Introduction

The amount of raw bandwidth available on fiber optic links has increased dramatically with advances in dense wavelength division multiplexing (DWDM) technology; however, existing optical network architectures are unable to fully utilize this bandwidth to support highly dynamic and bursty traffic. Optical burst switching [1] - [2] has been proposed as a new paradigm to provide the flexible and dynamic bandwidth allocation required to support such traffic. In OBS networks, incoming data is assembled into basic units, referred to as data bursts, which are then transported over the optical core network. Control signaling is performed out-of-band by control packets which carry information such as the length, the destination address, and the QoS requirements of the optical data burst. The control packet is separated from the data burst by an offset time, allowing the control packet to be processed at each intermediate node before the data burst arrives. OBS provides dynamic bandwidth allocation and statistical multiplexing of data. Aggregating IP packets into large sized bursts can compensate for slow switching time at core nodes. Core nodes with slower switching times require larger minimum burst lengths in order to minimize the switching overhead.

An important issue in OBS networks is data burst assembly. Burst assembly is the process of aggregating IP packets with the same destination into a burst at the edge node. The most common burst assembly approaches are *timer-based* and *threshold-based*. In a timer-based burst assembly approach, a burst is created and sent into the optical network when the time-out event is triggered. In a threshold-based approach, a limit is placed on the number of packets contained in each burst. A more efficient assembly scheme can be achieved by combining the timer-based and threshold-based approaches [5] - [8].

IP packets assembled in a data burst have a time delay constraint, called *maximum end-to-end delay*, determining the deadline by which the packet must reach its destination. Thus, the main motivation for implementing the timer-based burst assembly approach is to ensure IP packets don't wait at the edge node's assembly unit indefinitely before its maximum end-to-end delay is violated. If the arrival rate of incoming IP packets with the same destination is low, bursts are timed out and released before they reach their minimum burst length requirement determined by the core node switching time. Under such conditions, the timed out burst is smaller than the minimum length requirement. We refer to these short bursts as *sub-bursts*. Padding overhead must be added to sub-bursts in order to satisfy the minimum length requirement. However, excessive padding results in high link utilization and data burst blocking probability. Furthermore, when data bursts are timed-out, their aggregated IP packets will experience higher average delay. These concepts are illustrated in Fig. 1. In case (a) the data burst reaches its maximum size before it is timed out. Case (b) represents a situation in which the burst is timed out before it reaches its maximum size. In case (c) the data burst is timed out before it reaches the minimum required length and padding overhead must be added. Note that in this paper, we mainly focus on case (c) where the incoming IP packet arrival rate

[0]This work was supported in part by the National Science Foundation (NSF) under grant ANI-01-33899.

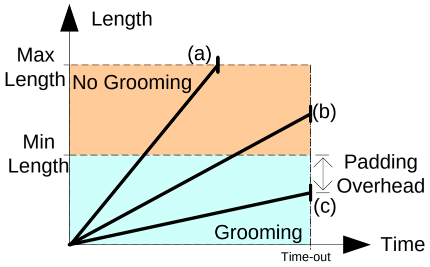

Fig. 1. Illustrating the time-based and threshold-based burst assembly approaches.

is low. Hence, we only assume a timer-based data burst assembly approach.

One approach to minimize the amount of padding overhead, as well as the average end-to-end IP packet delay due to low IP packet arrival rate is to *groom* bursts. Burst grooming is defined as aggregating multiple sub-bursts with different destinations together at the edge node and transmitting them as a single burst until sub-bursts are separated at a downstream node. In this process, some sub-bursts will have to be routed on multiple logical hops, where each logical hop corresponds to sub-burst retransmission.

The problem of aggregating and routing sub-bursts together, as well as determining their wavelength assignment, is referred to as the *data burst grooming problem*. Heuristic algorithms that attempt to solve the data burst grooming problem are referred to as *burst grooming algorithms*. These algorithms differ depending on their aggregation and routing criteria. For example, issues such as which sub-bursts and how many sub-bursts can be groomed together, or how long the accumulated length of the groomed burst should be, can have significant impact on the efficiency of the grooming algorithm under different network loading conditions.

The concept of burst grooming has been extensively studied for various circuit-switched WDM network topologies (ring, mesh, etc.) under different traffic scenarios (static or dynamic) [9] - [12]. The basic idea in all these problems is to share the wavelength dedicated to an established connection. The objective of data burst grooming in OBS over WDM networks, however, is to aggregate multiple sub-bursts to share the data burst created to satisfy a request. Data burst grooming in OBS has not received much attention in the literature. In [13] the authors consider data burst burst grooming at core nodes where several sub-bursts sharing a common path can be aggregated together in order to reduce switching overhead. The aggregated sub-bursts can be separated at a downstream node prior ro reaching their final destinations.

In this paper we address the problem of data burst grooming in OBS networks. In our study, we concentrate on grooming data bursts at the edge nodes. This study is motivated by the following network constraints: (a) the core node switching time is much larger than the average IP packet size; (b) there is a maximum end-to-end delay tolerance for incoming IP packets passing through the network.

The main contribution of this paper is an edge node architecture for enabling burst grooming, as well as several data burst grooming heuristic algorithms. Using simulation we examine the performance of our proposed grooming algorithms under specific network conditions. We compare our results with those obtained without burst grooming in terms of blocking probability and average end-to-end IP packet delay. We show that our proposed burst grooming techniques lead to performance improvement when the IP traffic arrival rate is low.

The remainder of this paper is organized as follows. In Section II, we describe the proposed edge node architecture in OBS networks capable of supporting data burst grooming. Section III formulates the data burst grooming problem and describes issues pertaining to the grooming heuristics. Section IV provides descriptions of two proposed grooming algorithms and details their characteristics. The performance results for each algorithm are presented in Section V. Possible modifications to each algorithm are also discussed and investigated. Finally, Section VI concludes the paper.

II. Node Architecture

The general core node architecture is described in detail in [3] and [4]. We assume that the switching time for core nodes is given as τ, and that the minimum required data burst duration is defined as a function of τ; i.e., $L^{MIN} = f(\tau)$. Throughout this paper, we refer to sub-bursts as the aggregated IP packets with the same destination, whose total length is less than L^{MIN}. Hence, a transmitted burst can contain multiple sub-bursts.

Fig. 2 shows the basic architecture of an edge node supporting data burst grooming. An ingress edge node, which generates and transmits data bursts to core nodes, performs the following operations: (a) burst assembly: aggregating incoming IP packets with the same destination (or other similar characteristics) in a virtual queue (VQ); (b) sub-burst grooming: combining multiple sub-bursts from different VQs into a single burst; (c) burst scheduling: attaching padding and preamble (framing) overhead to the bursts and scheduling them for transmission on an appropriate channel; (d) BHP generation: constructing the header packets and transmitting them prior to their corresponding data bursts.

In the egress path, as shown in Fig. 2, an egress edge node performs two basic functions: burst disassembly and IP routing. Upon receiving a data burst, the edge node initially disassembles the burst. The extracted sub-bursts, which need to be retransmitted to the downstream nodes are sent to the assembly unit, while the remaining sub-burst will be directed to the IP-routing unit. The IP-routing unit is a line card responsible for disassembling each sub-burst and sending its embedded packets to appropriate IP routers in the access layer of the network. We assume that the total IP packet delay in the network must be less that the maximum tolerable end-to-end packet delay, denoted by T_e.

III. Burst Grooming

In this section we first introduce some basic definitions and formulate the edge node grooming problem in OBS network, and then describe our proposed grooming algorithms.

A. Data burst grooming

We denote a sub-burst as b. Each sub-burst b consists of multiple IP packets with the same destination and can be characterized by its source, destination, and length: S_b, D_b, and L_b. As soon as an IP packet with destination D_b arrives to a node, a timer

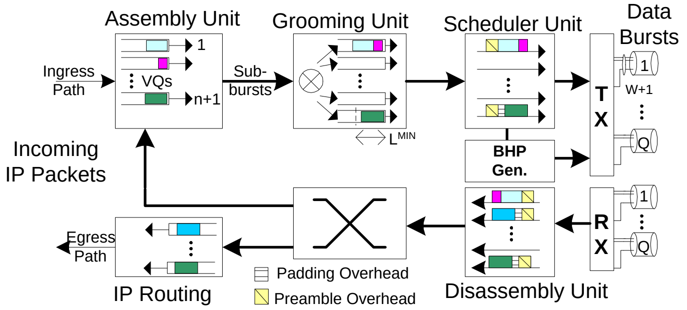

Fig. 2. An edge node architecture supporting burst grooming with Q ports and W data channels and one control channel on each port.

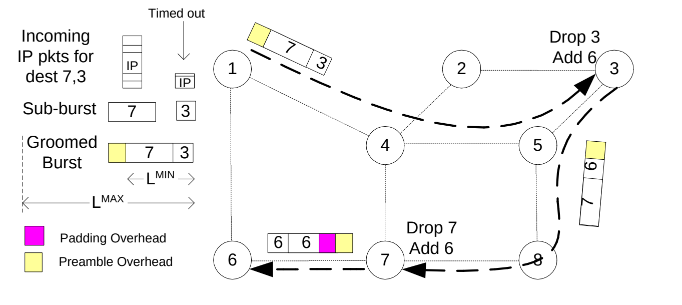

Fig. 3. A simple network carrying groomed data bursts.

is set for sub-burst b. The sub-burst will be released when it is timed out. The time-out value for data bursts in each virtual queue is bounded by the difference between the maximum tolerable end-to-end packet delay, T_e, and the sum of source-destination propagation delay and node processing delays, which includes the burst disassembly time at the destination node. In addition to the aforementioned parameters, each sub-burst, b, has a *remaining slack time*, denoted as δ_b. The remaining slack time is defined as the remaining tolerable end-to-end delay the sub-burst can tolerate before it reaches its destination.

We represent a groomed data burst by $\mathbf{G} = \{b_0, b_1, b_2, ...\}$, which is constructed by aggregating a number of sub-bursts with different destinations. We consider the first element (sub-burst) in the grooming set (b_0) as the timed-out sub-burst, which must be routed on a single hop. Hence, the first hop for all sub-bursts in $\mathbf{G}$ will be the node corresponding to the destination D_{b_0}. In our notation $|\mathbf{G}|$ indicates the number of sub-bursts groomed together. Clearly if $|\mathbf{G}|$=1 no grooming has been performed. Furthermore, we refer to G^{MAX} as the maximum number of sub-bursts which are allowed to be groomed together prior to transmission, hence $|\mathbf{G}| \leq G^{MAX}$.

We define the *hop-delay* as the delay time imposed on an incoming sub-burst due to electronic processing. In our study, we only consider the maximum hop-delay, expressed as T_h, and assume it is the same for all nodes. It is clear that the timed out sub-burst can only be groomed with any other sub-burst, b_i, whose remaining slack time satisfies the following expression:

$$T_p(S_{b_0}, D_{b_0}) + T_p(D_{b_0}, D_{b_i}) + T_h \leq \delta_{b_i} \leq T_e. \qquad (1)$$

In the above expression, $T_p(s, d)$ is the propagation delay from node s to node d. Note that δ_b for any given sub-burst is bounded by T_e.

When $\mathbf{G}$ reaches its first destination node, D_{b_0}, sub-burst b_0 is dropped. Then, each remaining sub-burst, b_i, in the grooming set $\mathbf{G}$, is directed to its proper virtual queue and its slack time is reduced by $T_h + T_p(S_{b_0}, D_{b_0})$. Incoming sub-bursts may be aggregated with the existing IP packets waiting in the corresponding virtual queue. In this case, the remaining slack time of the *combined* sub-burst is set to the remaining slack time of the earliest packet in the queue.

We illustrate the above concepts using the example shown in Fig. 3. In this example, The sub-burst at Node 1 going to Node 3 is timed out and it is groomed with another sub-burst with destination Node 7, in order to meet the minimum length requirement. At Node 3, the sub-burst with destination Node 3 is dropped. The remaining sub-burst going to Node 7 will be groomed with another sub-burst with destination Node 6. At Node 7, the sub-burst going to Node 6 is sent to the proper virtual queue and combined will all existing IP packets in the queue. When the timer is expired, the combined sub-burst going to Node 6 must be transmitted. In this case, since the minimum length is not met, padding overhead is added.

When a sub-burst b_0 is timed out, the burst grooming algorithm finds the appropriate $\mathbf{G}$ ($b_0 \in \mathbf{G}$) among all possible grooming combinations. Selection of the grooming set is based on the optimization objective of the grooming algorithm. Aggregating multiple sub-bursts reduces the *padding overhead* and consequently the network utilization, which in turn, can improve the blocking probability. However, this can potentially result in routing the groomed sub-bursts over longer physical paths. This phenomena, referred as the *routing overhead*, can impact the network throughput.

For example, consider Fig. 3, where at Node 1 the timed-out sub-burst going to Node 3 is groomed with the sub-burst going to Node 7. We denote the physical hop distance between node pair (s, d) by $H_p(s, d)$. In this case, the sub-burst going to Node 7, will be traveling over $H_p(1, 3) + H_p(3, 7) = 3 + 3 = 6$ physical hops, where as the shortest path between Node 1 and Node 7 includes only 2 physical hops: $H_p(1, 7) = 2$. This example, demonstrates that simple greedy aggregation of sub-bursts can have adverse effects. Consequently, an effective grooming policy must minimize both the padding and the routing overhead while minimizing additional hop-delay.

B. Problem formulation

In an OBS mesh network, data burst grooming can be performed at the edge node. In this case, each individual edge node must decide how to aggregate individual sub-bursts with durations smaller than the minimum length requirement, in order to optimize the throughput and reduce the probability of burst dropping. Hence, we can formulate the data burst grooming problem at the edge node as follows. *Given* the entire network information (including the physical network topology and full routing knowledge between all node pairs), the minimum required data burst duration, the maximum end-to-end delay that each IP packet can tolerate, and that a given sub-burst with duration smaller than the minimum required length has timed-out, *find* the available sub-bursts, b_i, which can be aggregated with the timed-out sub-burst, b_0, in order to minimize blocking probability.

We consider the following assumptions: all edge nodes have full grooming capability with no wavelength converters, and all transmitters and receivers are tunable to all wavelengths; all incoming IP packets have arbitrary lengths and a single destination; data bursts with durations shorter than the minimum burst length requirement will be subject to padding overhead; all IP packets in a virtual queue must be transmitted together. In addition, in this study, we focus on networks with low IP traffic arrival rate; thus, only a timer-based triggering scheme is assumed. We assume source routing, where the source node knows the entire path for all sub-bursts.

C. Description of grooming algorithms

An intuitive approach to reduce packet blocking probability is to develop effective grooming algorithms in order to reduce overall network overhead. The efficiency of grooming algorithm can be affected by several parameters, including the number of sub-bursts which can be groomed together, the accumulated length of the groomed sub-bursts, and the way groomed sub-bursts with different destinations are routed. These parameters can have conflicting impacts under different network conditions. For example, under light loading condition, having fewer constraints on the above parameters may considerably reduce the network overhead, resulting in higher network throughput. On the contrary, under moderate loading condition, asserting no constraints on the above parameters may notably alter the traffic characteristics and increase traffic burstiness, resulting in higher packet blocking.

We distinguish grooming algorithms by the way the source node calculates the padding and routing overheads due to burst grooming. Since the source node has no knowledge about the traffic between other node pairs, its padding overhead calculations are based on worst case *local* estimations. In our study, we consider two grooming algorithms: No-routing-overhead *(NoRO)* and Minimum-total-padding-overhead *(MinTO)*. In the first approach, we perform burst grooming only if no routing overhead is added and all sub-bursts travel through their shortest paths. In the MinTO algorithm, we relax the overhead routing constraint and assume that grooming can be implemented as long as the combined padding and routing overheads is reduced or maintained the same.

No-routing-overhead algorithm (NoRO): The main objective in this grooming algorithm is to select the grooming set, $\mathbf{G} = \{b_0, b_1, b_2, ...\}$, such that there is no routing overhead. The relative routing overhead for each sub-burst b_i in the grooming set $\mathbf{G}$ is calculated as follows:

$$Roh(b_i) = \frac{H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_i})}, \tag{2}$$

where b_0 is the timed-out sub-burst and $H_p(s, d)$ represents the number of physical hops on the shortest path between node pair (s, d). Having $Roh(b_i) = 1$, indicates that the destination of the timed-out sub-burst, D_{b_0}, is on the shortest path to the destination of the groomed sub-burst, D_{b_i}. The total relative routing overhead for a given grooming set $\mathbf{G}$ will be

$$TRoh(\mathbf{G}) = \sum_{b_i \in \mathbf{G}, b_i \neq b_0} Roh(b_i). \tag{3}$$

Consequently, the NoRO algorithm only considers the grooming sets with $TRoh(\mathbf{G}) = |\mathbf{G}| - 1$.

We now describe the details of the NoRO grooming algorithm as sub-burst b_0 is timed out. We denote all available sub-bursts in $n+1$ different virtual queues as a set of $\mathbf{S} = \{b_0, b_1, ..., b_i, ...\}$ and assume the length of b_0 is denoted by L_{b_0}.

```
• Step 0: Let G = $\{b_0\}$, Š = $\emptyset$, |Š| =0, and
  S = $\{b_0, b_1, ..., b_i, ...\}$
• Step 1: For each $b_i \in S$, $i \in [1, n]$:
    – If $\delta_{b_i}$ satisfies eqn. (1) continue;
      else, delete $b_i$ from S and go back to
      step 1;
    – If $Roh(b_i) = 1$, save $b_i$ as a feasible
      grooming solution: $b_i \rightarrow$Š; else, delete
      $b_i$ from S and go back to step 1;
• Step 2: For each $b_j \in$ Š, $j \in [1, m]$, $m$=|Š|
  and $m \leq n$:
    – Select $b_j$ with the largest length in
      Š: $b_j \rightarrow$ G
    – Set $L_G = L_G + L_{b_j}$ and remove $b_j$ from Š.
    – If $L_G < L^{MIN}$ and |G| $< G^{MAX}$ and
      go to Step 2; else, terminate the
      algorithm.
```

Minimum-total-overhead algorithm (MinTO): The NoRO algorithm is very *strict* in the sense that it only allows grooming along the shortest paths and it allows no *route deflection*. We define route deflection distance, $\Delta(b_0, b_i)$ as the number of *additional* physical hops a sub-burst, b_i, must traverse, when compared to its shortest path, before it reaches its destination:

$$\Delta(b_0, b_i) = (H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})) - H_p(S_{b_0}, D_{b_i}). \tag{4}$$

For example, referring to Fig. 3, the sub-burst going to Node 7 from Node 1 will have to tolerate a route deflection distance of (6-2)=4. The route deflection constraint imposed in the NoRO algorithm can be relaxed by allowing sub-bursts to be groomed as long as the combined relative routing and padding overhead is less or equal than the padding overhead resulting when no grooming is implemented. We define the relative routing and padding overhead, $RPoh(b_i)$, when sub-burst set $\mathbf{G}$ is groomed with b_i, where b_i does not belong to $\mathbf{G}$, as follows:

$$\begin{aligned} RPoh(b_i) = \{&max(L^{MIN}, L_G + L_{b_i}) \cdot H_p(S_{b_0}, D_{b_0}) + \\ &\sum_{b_j \in \mathbf{G}}^{b_j \neq b_0} max(L^{MIN}, L_{b_j}) \cdot H_p(D_{b_0}, D_{b_j}) + \\ &max(L^{MIN}, L_{b_i}) \cdot H_p(D_{b_0}, D_{b_i})\} / \\ \{&\sum_{b_j \in \mathbf{G}} max(L^{MIN}, L_{b_j}) \cdot H_p(S_{b_0}, D_{b_j}) + \\ &max(L^{MIN}, L_{b_i}) \cdot H_p(S_{b_0}, D_{b_i})\}. \end{aligned} \tag{5}$$

Details of the MinTO grooming algorithm as sub-burst b_0 with length L_{b_0} is timed out are as follow:

```
- Step 0: Let G = $\{b_0\}$, Š = $\emptyset$, |Š| =0, and S = $\{b_0, b_1, ..., b_i, ...\}$
- Step 1: For each $b_i \in S$, $i \in [1, n]$:
    – If $\delta_{b_i}$ satisfies eqn. (1) continue; else, delete $b_i$ from S and go back to step 1.
    – If $\Delta(b_0, b_i)$ is less than maximum allowable route deflection distance continue to the next step; else, delete $b_i$ from S and go back to Step 1.
    – If $RPoh(b_i) \leq 1$, save $b_i$ as a feasible grooming solution: $b_i \rightarrow$Š; else, delete $b_i$ from S and go back to Step 1.
- Step 2: Find $b_j$ with the smallest $RPoh(b_j)$ and largest length where $RPoh(b_j) \leq 1$; $b_j \in$Š, $j \in [1, m]$, $m$=|Š|, and $m \leq n$:
    – If $b_j$ exists, update the grooming set: $b_j \rightarrow$ G, $L_G = L_G + L_{b_j}$, remove $b_j$ from Š, and continue; otherwise terminate the algorithm.
    – If $L_G < L^{MIN}$ and |G| $< G^{MAX}$ and go to Step 2; otherwise terminate the algorithm.
```

D. Algorithm analysis and comparison

In this section we take a closer look at the MinTO algorithm and examine its performance under three different loading conditions. For simplicity we assume that maximum number of sub-bursts that can be groomed in a single burst is two, $G^{MAX} = 2$.

(a) $L_G, L_{b_0}, L_{b_i} < L^{MIN}$: In this case (5) will be reduced to

$$RPoh(b_i) = \frac{H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) + H_p(S_{b_0}, D_{b_i})}, \tag{6}$$

which must be less than unity for $\mathbf{G} = \{b_0\}$ to be groomed with b_i. Using (4), the necessary condition for $RPoh(b_i)$ to be less than unity can be expressed as

$$\Delta(b_0, b_i) \leq H_p(S_{b_0}, D_{b_0}). \tag{7}$$

If the route deflection distance is zero, $\Delta = 0$, under the low loading assumption, (6) is reduced to

$$RPoh(b_i) = \frac{H_p(S_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) + H_p(S_{b_0}, D_{b_i})}, \tag{8}$$

which is always less than unity. In this case, $RPoh(b_i)$ will be smaller for sub-bursts with shorter hop distance from S_{b_0} to D_{b_i}: $H_p(S_{b_0}, D_{b_i})$.

(b) $L_G \geq L^{MIN}, L_{b_0}, L_{b_i} < L^{MIN}$: In this case (5) will be reduced to

$$RPoh(b_i) = \frac{H_p(S_{b_0}, D_{b_0}) \cdot (L_G/L^{MIN}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) + H_p(S_{b_0}, D_{b_i})}. \tag{9}$$

Rewriting the above expression in terms of Δ, we obtain

$$\Delta(b_0, b_i) \leq H_p(S_{b_0}, D_{b_0})(1 - \epsilon) \quad where \quad 0 \leq \epsilon < 1. \tag{10}$$

The parameter ϵ is proportional to $r = L_G/L^{MIN}$ and it is defined such that $1 + \epsilon = min(1, r)$. Comparing (6) and (9), suggests that as long as $L_G < L^{MIN}$ and $H_p(D_{b_0}, D_{b_i}) < H_p(S_{b_0}, D_{b_i})$, the timed-out sub-burst can be groomed with b_i. However, as the load increases and $L_G > L^{MIN}$, less grooming can be expected.

(c) $L_{b_i} \approx L_G \geq L^{MIN}, L_{b_0} < L^{MIN}$: In this case (5) can be expressed as

$$RPoh(b_i) = \frac{H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) \cdot (L_G/L^{MIN}) + H_p(S_{b_0}, D_{b_i})}. \tag{11}$$

Using the definition for Δ, the above expression can be rewritten as

$$\Delta(b_0, b_i) \leq H_p(S_{b_0}, D_{b_0}) \cdot \epsilon \quad where \quad 0 < \epsilon \leq L^{MIN}/L^{MAX}, \tag{12}$$

with L^{MAX} being the maximum allowed burst length.

In the above discussion we can clearly see that, in order to minimize routing and padding overhead, MinTO continuously attempts to groom multiple small sub-bursts, whose destinations are closest to D_{b_0}. On the contrary, the NoRO algorithm mainly attempts to find the largest available sub-burst traveling along the timed-out sub-burst's path. We refer to these characteristics as *grooming aggressiveness* and *packet aggregation aggressiveness*. An interesting observation in comparing (7), (10), and (12) is that as the network load increases smaller route deflection distance will be allowed and hence, less grooming opportunities will be provided by MinTO. Furthermore, the above relationships show that under certain network conditions, MinTO reduces the overall overhead in the network by introducing minimum routing overhead, $\Delta \neq 0$. This is different from NoRO, which aggressively attempts to search for the largest available sub-bursts to be groomed, regardless of the network load.

We illustrate the behavior of the NoRO and MinTO using the example shown in Fig. 4, where a 5-node network with a single optical channel between each node pair is considered. We assume at Node a sub-burst b_y is timed out and can be groomed with one of the available sub-bursts: b_w, b_x, or b_z. Using the NoRO algorithm, if we groom sub-burst b_y with b_z, the lowest Roh value can be obtained. On the other hand, using the MinTO algorithm, the grooming choice changes depending on the length ratio of the available sub-bursts, namely, b_w, b_x, and b_z, over L^{MIN}. For example, assuming the length of b_z is much larger than L_{b_x} and L_{b_w}, the value of $RPoh$ for b_x, b_z and b_w varies depending on the length of the timed-out sub-burst, b_y, as shown in Fig. 5. It can be seen, that for high values of L_{b_y}/L^{MIN}, $RPoh(b_x)$ will be the smallest and hence, b_x will be selected to be groomed with b_y. This shows, that under special circumstances, the MinTO algorithm prefers to groom with an available sub-burst which results in larger route deflection distance. Fig. 6 demonstrates the range where the value of $RPoh(b_w)$, with $\Delta(b_0, b_w) = 2$ is smaller than $RPoh(b_x)$ with $\Delta(b_0, b_x) = 1$.

IV. Performance results

In this section we present the simulation results obtained by implementing the NoRO and MinTO algorithms and examine

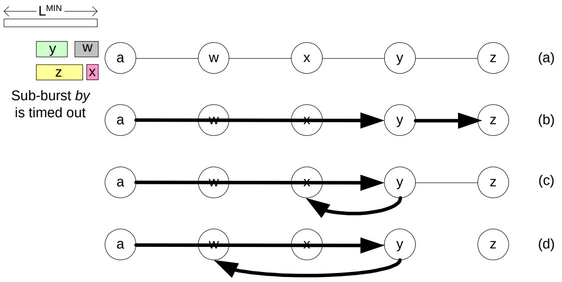

Fig. 4. An example of a 5-node network where sub-burst b_y going to Node y is timed out and it can be groomed with any one of the available sub-bursts: b_w, b_x, or b_z. Note that we assume the size of the grooming set is limited to $G^{MAX} = 2$.

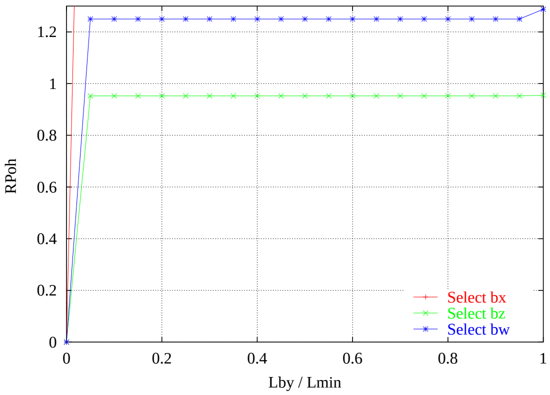

Fig. 5. Calculating the minimum routing and padding overhead for $C_x = \{b_y, b_x\}$, $C_z = \{b_y, b_z\}$, and $C_w = \{b_y, b_w\}$ as a function of L_{b_y}.

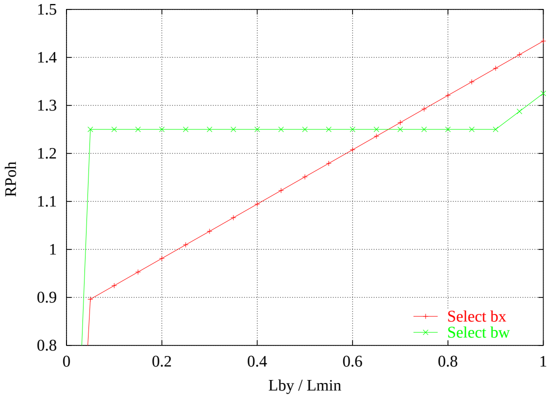

Fig. 6. Calculating minimum routing and padding overhead for $C_x = \{b_y, b_x\}$ and $C_w = \{b_y, b_w\}$ as a function of L_{b_y}.

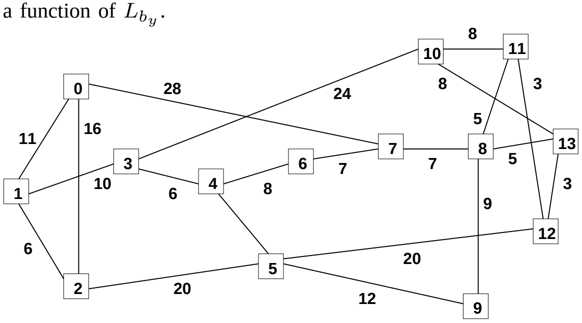

Fig. 7. The NSF network with 14 nodes and 21 bi-directional links.

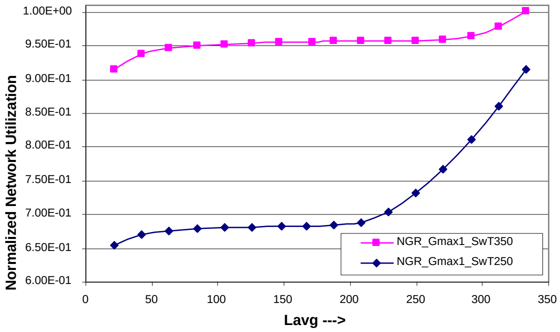

Fig. 8. IP Packet probability of blocking for different minimum burst length requirements, L^{MIN}=250 and 350, when no grooming is implemented.

different modifications to each. We have chosen the NSFNet backbone, shown in Fig. 7, as our test network. In this network, we assume each link is bi-directional with a fiber in each direction and the transmission rate is 10 Gbps. Our simulation model was developed based on the following assumptions: IP packet arrivals into the OBS network are Poisson with λ denoting their arrival rate and they are uniformly distributed over all sender-receiver pairs; IP packet length is fixed with 1250 bytes; the end-to-end allowed IP Packet delay is 50 ms; the switching time at the core node is 250 μs, requiring a minimum burst length of 250 packets for each data burst; each data burst can carry maximum of 2500 IP packets; and the data burst preamble size is 16 bytes. We also assume all nodes support data burst grooming capacity and are equipped with no wavelength convertors, and that each link has 8 wavelengths. We adopt the latest available unscheduled channel (LAUC) algorithm to schedule data bursts at the core nodes. Furthermore, we only consider timed-based assembly and assume all sub-bursts can be groomed as long as their accumulated length is less than the minimum required length. In our simulation study we mainly focus on the light traffic load scenario where sub-bursts typically time out before they reach their minimum required length, L^{MIN} and hence, their average length, L^{AVG} is less than L^{MIN}. Recall that the minimum required length (in terms of number of IP packets) is determined by the core node's switching time (in μs) and hence, we use these terms interchangeably. Also note that the average burst length between a node pair among N nodes, when no grooming is applied, can be calculated in terms of the network offered load as follows:

$$L^{AVG} = \frac{\lambda}{N \cdot (N-1)} \cdot T_o, \qquad (13)$$

where T_o is the tim-out value and we assume it is the same for all sub-bursts.

In our C-based simulation model we used confidence interval accuracy as the controlling factor. For each case of interest, the simulation was run until a confidence interval level of 90% was observed and an acceptably tight confidence interval were achieved. Calculations of the confidence interval were based on the variance within the collected observations [14]. All simulations were performed on a UNIX-based multiprocessor machine.

We first justify the importance of grooming as the core node's switching time increases. Fig. 8 compares the normalized network

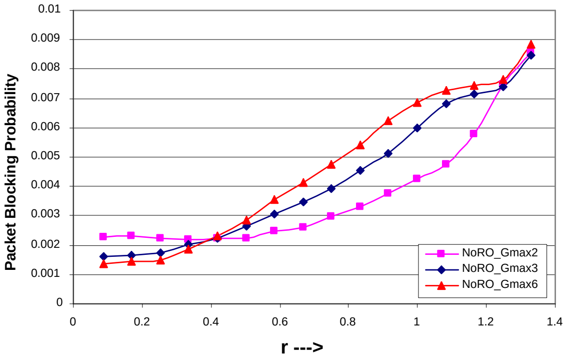

Fig. 9. Probability of blocking using NoRO with different G^{MAX} values: 2, 3, and 6.

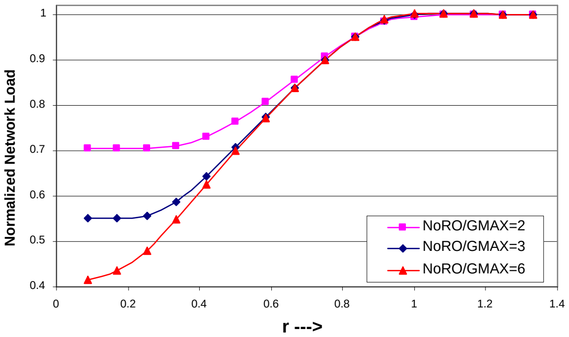

Fig. 10. Normalized link load carried in the network, including padding overhead obtained by implementing NoRO for G^{MAX}=2, 3 and 6

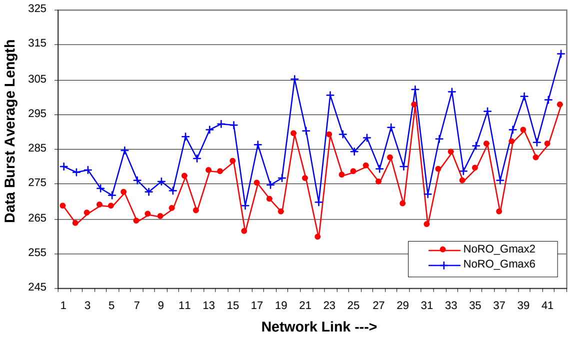

Fig. 11. Average data burst length per link (in number of IP packets) for G^{MAX}=2 and 6 when r =0.6 using NoRO.

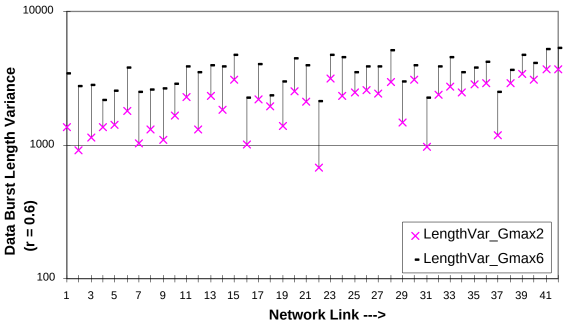

Fig. 12. Variance of data burst length per link for G^{MAX}=2 and 6 when r =0.6 using NoRO.

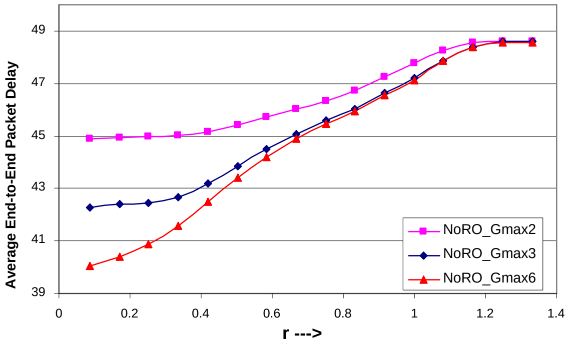

Fig. 13. Average end-to-end packet delay using NoRO with different G^{MAX} values: 2, 3, and 6.

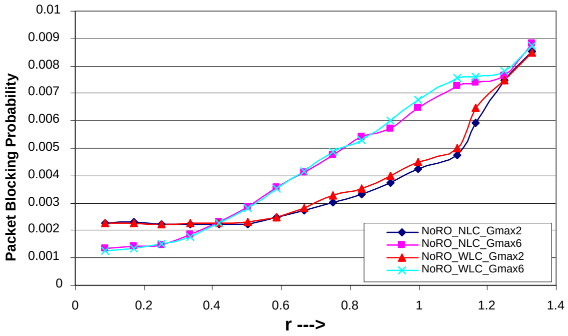

Fig. 14. Implementing NoRO with and without length constraint for G^{MAX} =2 and 6.

utilization when L^{MIN} changes from 250 to 350 IP packets, as the network load increases. In this figure we represent the network load as the average number of IP packets assembled in each timed-out sub-burst, denoted as L^{AVG}. Note that if the burst length is smaller than L^{MIN}, padding overhead will be added. Hence, for a given IP packet arrival rate, as L^{MIN} becomes larger, more sub-bursts will be timed out before reaching their minimum length requirement. Consequently, more padding overhead will be generated and the link utilization is increased, resulting in higher packet blocking probability. We continue this section by first characterizing each grooming algorithm in details and then comparing them together and to the case in which no grooming is implemented.

A. Characterizing the NoRO algorithm

Fig. 9 shows the IP packet blocking as a function of average length ratio, $r = L^{AVG}/L^{MIN}$, when the NoRO grooming algorithm is implemented for different maximum grooming set sizes: G^{MAX} =2, 3, and 6. As this figure suggests, under very light traffic condition, $r < 0.45$, as more data bursts are allowed to be groomed together, lower IP packet blocking probability can be achieved. Note that under our simulation assumptions, further increase in the maximum number of sub-bursts which can be groomed in a single burst, $G^{MAX} > 6$, does not result in further performance improvement. This is due to practical limitations on the number available sub-bursts in virtual queues.

Under higher loading conditions, when $0.45 \leq r < 0.85$, as G^{MAX} increases and more sub-bursts are allowed to be groomed together, IP packet blocking probability increases. As the load continues to increase, $r \geq 0.85$, only a small percentage of sub-bursts are shorter than the minimum required length and hence, less grooming will take place. Under this loading scenario, as Fig. 9 suggests, the performance with different G^{MAX} values, tend to result in similar packet blocking probability.

Our simulation results, as shown in Fig. 10, indicate that in general, when $r < 1$, as G^{MAX} increases, lower link load can be achieved. This contradicts the results in Fig. 9, which suggests that when $0.45 \leq r < 0.85$ for higher G^{MAX} values higher packet blocking is obtained. In order to understand this paradox, we examine the traffic characteristics throughout the network due to burst grooming. Fig. 11 shows the mean burst length generated on each link for r =0.6. Note that under light loading condition, the difference between the average length is slightly higher when $G^{MAX} = 6$ compared to when $G^{MAX} = 2$. It can also be shown that the mean interarrival time on each link as G^{MAX} changes from 2 to 6 slightly increases. Fig. 12 compares the variance of data burst length transmitted on links 1 through 42 for G^{MAX} =2 and 6 when $r = 0.6$. This figure shows that as G^{MAX} increases from 2 to 6, the variance of burst length considerably increases as well. This is one possible reason that IP blocking is higher for larger values of G^{MAX} under light network loading. Note that that when the traffic loading is very low, $Var(L^{AVG})|_{G_{MAX}=6}$ and $Var(L^{AVG})|_{G_{MAX}=2}$ are much smaller and relatively close to each other.

The average end-to-end packet delay obtained from NoRO is shown in Fig. 13. Note that, as G^{MAX} increases, lower average delay can be achieved. This is due to the fact that by allowing higher number of sub-bursts to be groomed in a single burst, fewer sub-bursts will have to wait until they are timed out. Consequently, fewer sub-bursts experience the maximum end-to-end delay. As we mentioned before, increasing the value of G^{MAX} beyond 6, due to practical limitations of the number of available sub-bursts in virtual queues, will not result in further improvement in average end-to-end packet delay.

A possible modification to the NoRO algorithm is to disallow the length of the groomed data sub-burst to become larger than L^{MIN}, $L_G \leq L^{MIN}$. We call this approach the NoRO algorithm with length constraint, NoRO-WLC. Fig. 14 compares the performance of NoRO with no length constraint, NoRO-NLC, and NoRO-WLC for G^{MAX}=2 and 6 in terms of IP packet blocking probability. Note that at low loads the two approaches are relatively comparable. However, as the network load increases, the NoRO-NLC tends to be more aggressive and more sub-bursts will be subject to grooming. Hence, a slightly better packet blocking can be obtained for NoRO-NLC. For the remaining of this paper we ignore NoRO-WLC and only focus on NoRO-NLC and refer to it as NoRO.

The above results obtained for different network loading conditions, indicate that when the network loading is very light, the data burst blocking probability of success is almost *independent* of the average data burst length and closely related to the *number* of bursts generated in the network. Therefore, by allowing more sub-bursts to be groomed together, lower blocking and average end-to-end packet delay can be expected. On the other hand, at higher loads, although having higher G^{MAX} values result in fewer number of bursts generated into the network, the expected average burst length will be higher. Under such conditions, data burst probability of success becomes more correlated to the traffic behavior. By allowing higher number of sub-bursts to be groomed in a single burst, the traffic behaves more bursty, which in turn can increase the IP packet blocking probability.

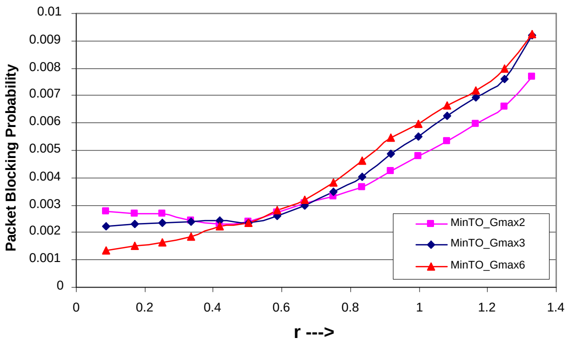

Fig. 15. IP packet blocking probability using MinTO with different G^{MAX} values: 2, 3 and 6.

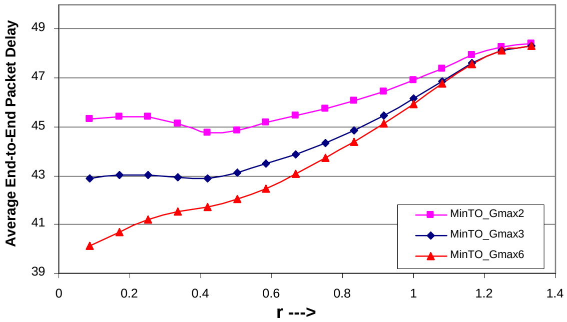

Fig. 16. Average end-to-end IP packet delay using MinTO with different G^{MAX} values: 2, 3, and 6.

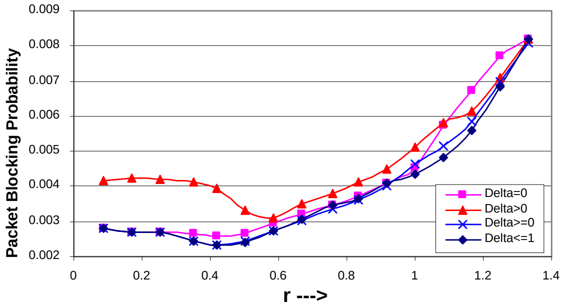

Fig. 17. Probability of blocking using variations of MinTO (MinRO-NRO and MinTO-WRO) with different route deflection distance, Δ, constraints when $G^{MAX} =$ 2.

B. Characterizing the MinTO algorithm

We now examine the performance of the MinTO algorithm. Fig. 15 compares the performance of MinTO in terms of packet blocking probability for G^{MAX}=2, 3, and 6. Note that in general, the performance results in terms of packet blocking and delay obtained for MinTO follow similar trends as discussed for NoRO.

In order to gain an insight into the MinTO algorithm operation, we investigate its performance under different route deflection distance, Δ, constraints. Hence, we develop two variations to

MinTO, which differ in the way available sub-bursts are selected for grooming with a timed-out sub-burst and the way they are routed. Note that in the following cases, data burst grooming can only be allowed if (5) is less than or equal unity:

- *MinTO with no routing overhead,* $\Delta = 0$ *(MinTO-NRO):* This is the case in which the timed-out sub-burst, b_0, can only be groomed with sub-bursts whose shortest path overlaps the shortest route between S_{b_0} and D_{b_0}.

- *MinTO with routing overhead only,* $\Delta > 0$ *(MinTO-WRO):* In this case the timed-out sub-burst, b_0, only grooms with other sub-bursts whose shortest path *does not* overlap the shortest path between S_{b_0} and D_{b_0}. Therefore, in this case any attached sub-burst, b_i, is expected to detour from its shortest path by Δ hops.

Fig. 17 compares the IP packet blocking probability achieved by MinTO-NRO and MinTO-WRO, when G^{MAX} is limited to 2. It can be shown that in general, under very light traffic conditions, $r < 0.45$, MinTO-NRO provides more grooming opportunities. As the network load increases, $0.45 \leq r < 0.85$, MinTO-NRO continues to outperform MinTO-WRO. However, this performance improvement is less significant. This is mainly due to the fact that, in general, as the network load increases, the impact of additional padding overhead is less significant and hence, the relative efficiency of MinRoh-NRO in terms of reducing the padding overhead becomes less notable. Under higher loading condition, $r \geq 0.85$ the performance of MinTO-NRO starts degrading when compared to MinTO-WRO. Such decline in performance is the direct result of having high variance due to relatively higher aggregation aggressiveness of MinTO-NRO.

MinTO combines the results obtained from MinTO-NRO and MinTO-WRO. In fact, Fig. 17 suggests that when MinTO is not constrained ($\Delta \geq 0$), the overall packet blocking probability is slightly improved compared to the lowest performance from either MinTO-NRO or MinTO-WRO. The results obtained for MinTO indicate that at lower loads, the improvements are mainly due to having no routing overhead. On the other hand, at higher loads, such improvements are primarily due to minimizing the padding overhead while reducing the traffic burstiness through a less aggressive grooming approach.

A major drawback of MinTO is that it can potentially send some sub-bursts through long paths, causing significant route deflection distance. Consequently, these sub-bursts will be more vulnerable to blocking at intermediate nodes. One way to avoid excessive route deflection is to impose an upper bound on the maximum route deflection distance, for example $\Delta \leq 1$. Fig. 17 shows that under such constraint, at higher loads, slightly lower packet blocking can be achieved. The tradeoff for such constraint is, of course, higher average end-to-end packet delay.

As a final note, we highlight the fact that the results described for MinTO-NRO and MinTO-WRO, can also be verified by (7), (10), and (12). Since $\Delta = 0$ in MinTO-NRO, the above expressions are always satisfied. However, in case of MinTO-WRO, as the network load increases, less grooming will be performed. Furthermore, it is interesting to mention that generally, in terms of aggregating more IP packets into a data burst by means of grooming, MinTO-NRO is less aggressive than NoRO. This is because NoRO attempts to groom the timed-out sub-bursts with the largest available sub-burst. On the other hand, the objective of MinTO-NRO is to minimize the padding and hence, it tends to find the smallest available sub-burst for grooming. In the rest of this section, we only consider MinTO where $\Delta \geq 0$.

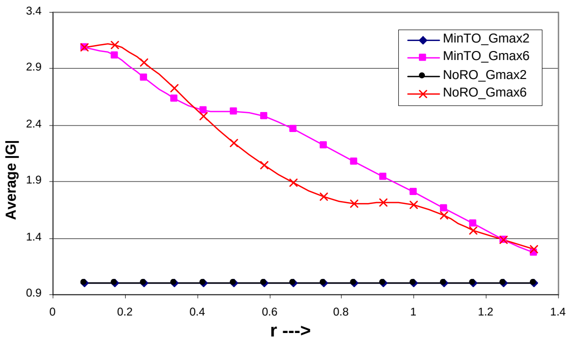

Fig. 18. Comparing the average number of sub-bursts groomed in a single burst using NoRO and MinTO for G^{MAX}=2 and 6.

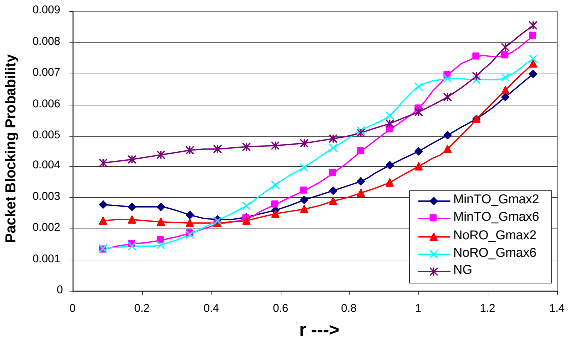

Fig. 19. Comparing the blocking probability using NoRO and MinTO for G^{MAX}=2 and 6.

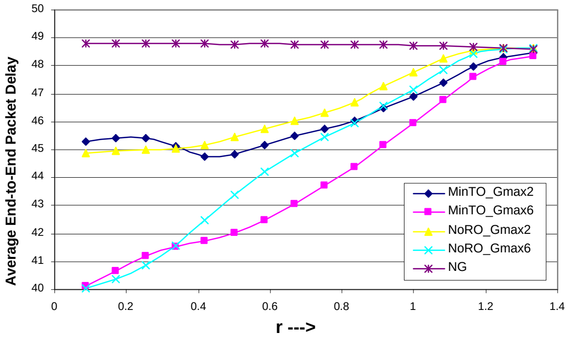

Fig. 20. Comparing the average end-to-end packet delay using NoRO and MinTO for G^{MAX}=2 and 6.

C. Grooming algorithm comparison

In this section we compare the performance of NoRO and MinTO with the case when no grooming is applied. Fig. 18 shows the average size of the grooming set, $\overline{|\mathbf{G}|}$, obtained for G^{MAX} =2 and 6 when NoRO and MinTO algorithms are implemented. When G^{MAX} =2, $\overline{|\mathbf{G}|}$ is limited to 2 for both grooming approaches. On the other hand, when G^{MAX} =6, there is no practical limitations in terms of $\overline{|\mathbf{G}|}$. In this case, under very light traffic condition, NoRO provides higher grooming opportunities. As the network load increases, MinTO tends to allow more sub-bursts to be groomed in a single burst. Eventually, as the network load becomes large enough, the grooming capability of both algorithms becomes the same until no grooming is required anymore.

Fig. 19 shows the packet blocking probability obtained by implementing the NoRO and MinTO for G^{MAX} =2 and 6. Under very light loading condition, $r < 0.45$ as we mentioned earlier, the performance of the grooming algorithm closely follows the link utilization. Hence, a more aggressive approach, where more sub-bursts with longer average lengths are groomed can further improve the performance. When G^{MAX} is large, G^{MAX} =6, and there is no practical limitation on how many sub-bursts can be groomed, NoRO results in higher average grooming size, $|\overline{\mathbf{G}}|$. Similarly, when the maximum number of sub-bursts which can be groomed is limited, say G^{MAX} =2, NoRO still outperforms MinTO. In this case, although each timed-out sub-burst can only be groomed with a single sub-burst, NoRO tends to select the sub-burst with the largest length.

As the load increases beyond $0.45 \leq r < 0.85$, MinTO tends to allow more sub-bursts to be packed into a single burst. This is primarily due to the fact that NoRO tends to groom with the largest available sub-burst, whereas, MinTO attempts to minimize the accumulated overhead and hence it tends to groom with shorter sub-bursts. Consequently, as Fig. 19 suggests, when G^{MAX} is large, say 6, under such traffic condition, MinTO outperforms NoRO in terms of packet blocking probability. On the other hand, when G^{MAX} is limited to 2, $|\overline{\mathbf{G}}|$ will be the same for both grooming approaches. Recall that the performance of MinTO under light traffic regime is governed by MinTO-NRO, which, as we explained, is less aggressive than NoRO in terms of packet aggregation. Therefore, NoRO results in slightly lower overall padding and blocking, as shown in Fig. 19.

An interesting observation in Fig. 19 is that under moderate traffic load, $r \geq 0.85$, when G^{MAX} is not limited, the performance of both grooming algorithms become slightly *worse* than when no grooming is implemented. In such cases data burst grooming results in high variant traffic characteristic and thus, the network performance degrades. On the other hand, with limited G^{MAX}, at moderate loads, both grooming approaches considerably outperform the case when no grooming is implemented. Note that under such traffic scenario, MinTO tends to outperform NoRO in terms of packet blocking probability. This is due to higher variance and bursty traffic behavior resulted by NoRO at moderate loading condition, due to its packet aggregation aggressiveness.

As a final note, it must be pointed out that at $r \approx$0.45, the performance of both algorithms is the same. This is due to the fact that at this crossing point, with uniform traffic, all statistical parameters, including the average length, $|\overline{\mathbf{G}}|$, utilization, etc. are the same.

Fig. 20 shows the average end-to-end packet delay obtained by implementing NoRO and MinTO. As this figure suggests, in general, the average end-to-end packet delay due to grooming is much less than the case in which no grooming is implemented. Under very low loads, $r < 0.45$, the MinTO slightly outperform NoRO regardless of the G^{MAX} value. However, as the load increases, $0.45 \leq r < 0.85$, NoRO results in higher average delay compared to MinTO. This is due to the fact that in the NoRO algorithm, timed out sub-bursts tend to be groomed with longer sub-bursts. Therefore, on average, IP packets will be spending longer time in the assembly unit. Clearly, as the load increases, the impact of burst grooming is reduced and T_e^{AVG} tends to become the same as the case with no grooming.

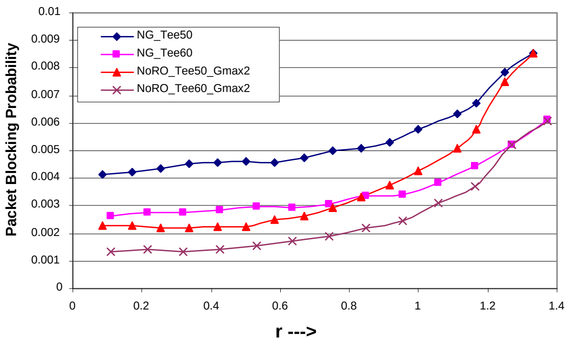

Fig. 21. Comparing the packet blocking probability using NoRoh and no grooming for different T_e values.

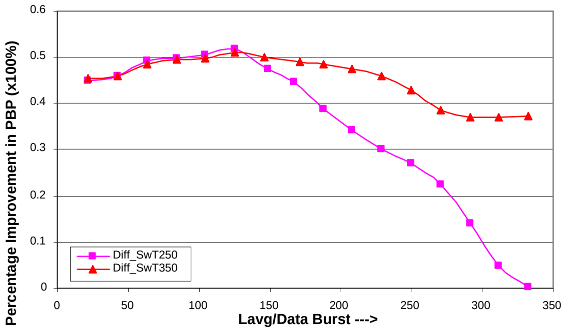

Fig. 22. The percentage PBP improvement with and without NoRoh as L^{MIN} changes from 250 to 350 for $G^{MAX} = 2$

The above results indicate that in general, when burst grooming is applicable, both NoRO and MinTO grooming approaches can reduce the link load as well as the average end-to-end packet delay. On the other hand, these grooming techniques tend to increase traffic burstiness on network links. Depending on the network loading condition, NoRO and MinTO perform differently. When the load is very low, allowing unlimited number of sub-bursts to be groomed together results in better performance, both in terms of packet blocking and average end-to-end delay. Under light loading condition, the NoRO algorithm results in higher IP packet aggregation than the MinTO algorithm. As the load increases, G^{MAX} must be limited to 2 in order to avoid altering the traffic characteristic and becoming highly bursty.

D. *Performance of NoRO under different network parameters*

In this section we investigate the performance of the grooming algorithms as the maximum end-to-end packet delay, T_e, and the minimum burst length requirement, L^{MIN}, which is equivalent to core node switching time, vary. Since both NoRO and MinTO behave similarly under such changes, we only focus on performance of the NoRO grooming algorithm.

In general, for a given switching time and average network load, as T_e decreases, data bursts time out earlier and hence, the average data burst length tends to become smaller. Consequently, more overhead will be generated and higher packet blocking probability is expected. Fig. 21 shows the packet blocking probability using NoRO for G^{MAX}=2 and T_e=50 and 60 ms. This figure suggests that for a given network load and switching time, NoRO is more effective in terms of packet blocking probability for smaller values of T_e.

It can be shown that in general, the behavior of NoRO in terms packet blocking probability and average end-to-end delay for different values of maximum grooming sizes, G^{MAX} =1, 2, 3, and 6, remains the same, regardless of the minimum burst length requirement, L^{MIN}. Hence, under low loading conditions, for larger values of L^{MIN}, allowing more sub-bursts to be groomed in a single burst, will be more effective. Fig. 22 shows the percentage performance improvement of NoRO with $G^{MAX} = 2$ compared to no grooming, as L^{MIN} changes from 250 to 350. This figure is plotted as a function of network load represented in terms of average burst length. Fig. 22 suggests that as L^{MIN} increases, NoRO become more effective for higher network loads, until average burst size becomes comparable with the minimum burst length requirement. Similarly, it can be shown that for a given network load and T_e, as L^{MIN} increases burst grooming can become more effective in terms of lowering the average end-to-end packet delay.

V. Conclusion

In this paper we discussed the problem of data burst grooming in optical burst-switched networks. The main motivation for this study is improving network performance when the traffic load is light and the core node's switching time is larger than the average size of sub-bursts. Under such assumptions, sub-bursts will time out before they reach their minimum required length, and hence, padding overhead must be added. We developed two grooming algorithms, namely MinTO and NoRO, which aggregate multiple small sub-bursts together in order to reduce the padding overhead, while minimizing any added routing overhead.

Through a comprehensive simulation study we investigated the performance of the MinTO and NoRO algorithms in terms of packet blocking probability and average end-to-end delay. Our results show that, in general, the proposed grooming algorithms can significantly improve the performance when compared with the case with no grooming. However, careful considerations must be given to network loading condition and the number of sub-bursts allowed to be groomed together. We showed that simple greedy algorithms will not perform sufficiently due to the fact that they alter the network traffic characteristics negatively and make it more bursty.

In this study, we demonstrated that with limited grooming the network packet blocking probability can be considerably improved, and the average end-to-end packet delay throughout the OBS network can be decreased. Under particular network conditions, the performance can farther be improved if higher number of sub-bursts are allowed to be groomed.

One area of future work would be to extend the proposed burst grooming framework such that it can support service differentiation and QoS. Another problem is to study the data burst grooming under static traffic scenario, where the average traffic between each node pair is known in advance.

References

[1] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet," *Journal of High Speed Networks*, vol. 8, no.1, pp.69-84, Jan. 1999.

[2] J.S. Turner, "Terabit Burst Switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3-16, Jan. 1999.

[3] Y. Xiong, M. Vanderhoute, and H.C. Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838-1851, Oct. 2000.

[4] F. Farahmand, V. M. Vokkarane, and J. P. Jue, "Practical Priority Contention Resolution for Slotted Optical Burst Switching Networks," *Proceedings, IEEE/SPIE First International Workshop on Optical Burst Switching (WOBS) 2003*, Dallas, TX, Oct. 2003.

[5] V.M. Vokkarane, K. Haridoss, and J. P. Jue, "Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks," *Proceedings, SPIE OptiComm 2002*, Boston, MA, vol.4874, pp. 125-136, July 2002.

[6] M. Casoni, E. Luppi, M. Merani "Impact of Assembly Algorithms on End-to-End Performance in Optical Burst Switched Networks with Different QoS Classes," *Proceedings, IEEE/SPIE First Workshop on Workshop on Optical Burst Switching (2004*, San Jose, CA, Oct. 2004.

[7] X. Yu, Y. Chen, and C. Qiao, "A study of Traffic Statistics of Assembled Burst Traffic in Oprical Burst Switched Networks," *Proceedings, SPIE OptiComm 2002*, Boston, MA, vol.4874, pp. 149-159, July 2002.

[8] S. Oh and M. Kang, "A Burst Assembly Algorithm in Optical Burst Switching Networks," *Proceedings, Optical Fiber Communication Conference and Exhibit (OFC) 2002*, pp. 771-773, March, 2002.

[9] R. Dutta and G.N. Rouskas, "Traffic grooming in WDM networks: past and future," *IEEE Network,* vol. 16, no. 6, pp. 46 -56, Nov.-Dec. 2002.

[10] P. J. Lin and E. Modiano, "Traffic grooming in WDM networks," *IEEE Communications Magazine*, vol. 39, no. 7, pp. 124 -129, July 2001.

[11] C. Xin, C. Qiao, and S. Dixit, "Analysis of single-hop traffic grooming in mesh WDM optical networks," *Proceedings, SPIE OptiComm* 2003, Dallas, TX, pp.91-101, Oct. 2003.

[12] F. Farahmand, X. Huang, and J. P. Jue, "Efficient Online Traffic Grooming Algorithms in WDM Mesh Networks with Drop-and-Continue Node Architecture," *Proceedings, BroadNets 2004*, Dallas, TX, November, 2004.

[13] S. Sheeshia, C. Qiao "Burst Grooming in Optical-Burst-Switched Networks," *Proceedings, IEEE/SPIE First Workshop on Traffic Grooming in WDM Networks (2004*, San Jose, CA, Oct. 2004.

[14] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley - Interscience, New York, NY, April 1991.