Resource Allocation Strategies for a non-Continuous Sliding Window Traffic Model in WDM Networks

Ying Chen, Arunita Jaekel and Ataul Bari
School of Computer Science, University of Windsor
401 Sunset Ave., Windsor, ON N9B 3P4, Canada
E-mail: {chen13r, arunita, bari1}@uwindsor.ca

Abstract*—In recent years, a number of papers have shown that the *scheduled traffic model*, which exploits knowledge of the connection holding times of traffic demands, can lead to significant improvements in resource utilization in WDM networks. In such a traffic model, the setup and the teardown times of the scheduled demands may be known in advance (*fixed window model*) or may be allowed to slide within a larger window (*sliding window model*). In both fixed and sliding window models, once the transmission of a demand is started, it continues until the entire data has been transmitted. However, there are many applications where such continuous data transmission is not strictly required. In this paper, we introduce a new model, the *non-continuous sliding window model*, where a demand may be decomposed into two or more components and each component can be sent separately. We first present an integer linear program (ILP) formulation for resource allocation under the non-continuous sliding window model and show that both the fixed and the traditional sliding window models can be treated as a special case of our generalized non-continuous sliding window model. Our formulations can accommodate fixed, sliding, and non-continuous demands, or any combination of these demand types. We also provide a heuristic algorithm that can be used for practical sized networks with a large number of overlapping demands. Simulation results on various networks, using different demand sets, show that our model results in significant performance improvements, even over results obtained using traditional scheduled traffic models, which already outperform holding-time-unaware models.

I. INTRODUCTION

Wavelength division multiplexing (WDM) allows optical backbone networks use to transport huge amounts of data quickly and reliably over long distances. Depending upon the specific applications, traffic flowing over the network may be characterized using different models; the two most commonly used traffic models are the static model (demands are relatively stable over long periods of time) and the dynamic model (demands arrive randomly, with the arrival rates and durations of demands typically following a specified distribution). The *scheduled traffic model* [1], [2], [3], has been introduced in recent years for applications that require periodic use of bandwidth (e.g. once per day) at predefined times. The setup and the teardown times of such scheduled demands may be known in advance (*fixed window model*) or may be allowed to slide within a larger window (*sliding window model*). Resource allocation techniques for the scheduled traffic model, allow sharing of resources among time-disjoint demands [2], [3], [4], [5], and it has been well established that such connection *holding time aware* strategies lead to considerable improvements over traditional *holding time unaware* approaches [4].

A significant number of recent papers have investigated the resource provisioning problem for scheduled lightpaths, under both the fixed and sliding window models. However, all these approaches typically assume that once transmission of a demand starts, it continues until all the data has been transmitted. We will refer to this model as the *continuous* scheduled traffic model. Such a model can be appropriate for applications such as a daily "virtual classroom", where bandwidth is required continuously for several hours, as long as the class is "in session". On the other hand, let us consider an application where a bank has to transfer its data nightly to a central location. The actual data transfer requires 1 hour and must be completed some time between 1am and 4am. In this case it is not necessary to send the data continuously; instead the data may be divided into several smaller components and each component sent separately, as long as the entire data is transferred within the specified time window between 1am and 4am. We will refer to this type of data transmission model as the *non-continuous* scheduled traffic model.

The *non-continuous* scheduled traffic model adds another degree of flexibility to the existing *sliding* window model, which can be exploited to generate more resource efficient solutions to the network design problem, or to accommodate more traffic for a given set of resource constraints. The advantages of the new *non-continuous* scheduled traffic model proposed in this paper are illustrated by the example in Fig. 1. We consider a single fiber link and for simplicity, we assume that the link can accommodate only one WDM channel. We also consider two demands d_1 and d_2, where d_1 (d_2) requires the entire WDM channel for 2 hours (3 hours), within time window 1 - 4 (0 -5). Clearly, under the *continuous* sliding window model, it will not be possible to accommodate both of these demands. However, as shown in Fig. 1b, both demands can be easily handled using the *non-continuous* scheduled traffic model, by dividing d_2 into two components d_{2-1} and d_{2-2} respectively.

In this paper, we propose a new traffic model called the non-continuous scheduled traffic model, which can lead to more efficient network design. In addition to the usual routing and wavelength assignment (RWA) [10] issues involved in scheduling lightpath demands, design strategies under the non-continuous model also need to take into consideration a number of other important factors such as: i) which demands

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249

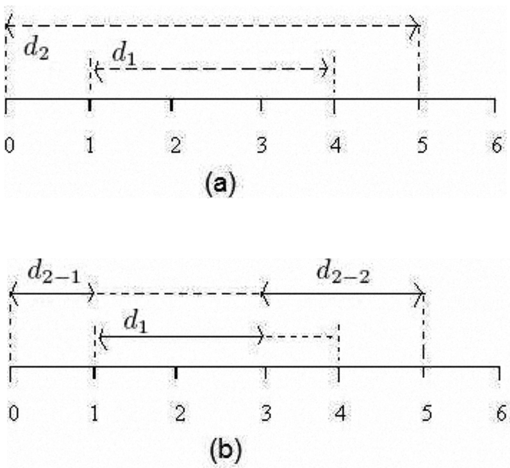

Fig. 1. An example of routing under the sliding window and the non-continuous scheduled traffic model. (a) Two demands, d_1 and d_2, with holding times 2 and 3 hours, and the time window 1 - 4 and 0 - 5 hours, respectively. Both demands cannot be accommodated under continuous sliding window model. (b) Using the non-continuous scheduled traffic model, both demands are accommodated by dividing d_2 into two components, d_{2-1} and d_{2-2}, with holding time 1 hour (in time wondow 0 - 1 hour) and 2 hours (in time window 3 - 5 hours), respectively.

(if any) should be divided into segments, ii) the number and sizes of the segments for each demand, and iii) how to schedule the individual segments to optimize resource utilization. Therefore, resource allocation under the new model can be viewed as a complex optimization problem, and we have provided an integer linear program (ILP) formulation, as well as a heuristic for solving this problem.

We note that the non-continuous model will lead to some additional costs at higher levels, in terms of reassembling multiple segments. For this paper, we focus primarily on bandwidth resources in the fiber links, and do not consider the additional costs at higher layers. Demands for which multiple segments are not acceptable are treated as a special case and can be easily handled by our formulation as discussed in section III-C. The main contributions of this paper are:

i) We propose a new non-continuous scheduled traffic model, which is more flexible than the existing sliding window model and can be used for a number of applications requiring periodic use of bandwidth.
ii) We present an ILP formulation that optimally solves the complete design problem and show that the traditional fixed and sliding window models can be treated as a special case of our proposed model.
iii) We present an efficient heuristic that can be used for larger networks with many scheduled demands.
iv) We demonstrate, through simulations that significant improvements can be achieved using our approach, even compared to existing *holding time aware* models, which already outperform *holding-time-unaware* models.

The remainder of the paper is organized as follows. Section II reviews the scheduled traffic model. Sections III and IV present our ILP formulation and heuristic for resource allocation under the non-continuous scheduled traffic model. We discuss and analyze our results in Section V and present our conclusions in Section VI.

II. Related Work

A. Scheduled Traffic Model

Each demand in the fixed window scheduled traffic model is represented as (s, d, η, st, et), where s and d denote the source and the destination of the demand, η represents the number of requested lightpaths for the demand and st (et) is the setup (teardown) time of the demand [1]. The fixed window traffic model can be augmented so that the setup and teardown times are no longer fixed, but can slide within a larger window [2], [3]. This is referred to as the *sliding* scheduled traffic model. For the sliding scheduled traffic model [2], [3] the demand setup and teardown times (st and et) are not known beforehand. Instead, a larger window of time (α, ω) and a demand holding time are specified for each demand. The scheduled demands in the sliding scheduled traffic model are represented by $(s, d, \eta, \alpha, \omega, \tau)$, where s, d and η have the same meaning as above, α, ω are the start and end times of the larger window during which the demand must be met and $\tau (0 < \tau \leq \omega - \alpha)$ is the demand holding time [2].

Resource allocation for lightpaths under the scheduled traffic model have received considerable research attention in recent years [4], [5], [6], [7], [8], [9], and it has been shown that connection holding time aware approaches consistently outperform traditional RWA algorithms for scheduled lightpath demands [4]. In [1], the authors have presented a branch and bound algorithm and a tabu search based approach to solve the routing problem in WDM networks. In [4], [6] the authors have proposed optimal ILP formulations for the design of survivable wavelength convertible networks, under the fixed window scheduled traffic model. Heuristic solutions for the same problem have been presented in [7], [8], [9]. In [5], an ILP formulation and a heuristic are presented for prioritized demands under the fixed scheduled traffic model where the nodes are capable of wavelength conversion. In [2] the authors have provided a heuristic algorithm for scheduling the demands and solving the RWA problem for a fault-free network, without wavelength conversion. In [11] the authors have presented an optimal ILP formulation for resource allocation under the sliding window model. In [3], the authors have investigated the relationship between the wavelength efficiency and the time flexibility of the scheduled demands. Zheng *et al* [12] present bandwidth reservation model and signaling control model to provide advance reservation service in Generalized Multi-Protocol Label Switching (GMPLS) based WDM networks.

III. Resource allocation for non-continuous scheduled traffic model

In this section, we present our ILP formulation for scheduling non-continuous demands. The objective is to accommodate as many demands as possible, under a given set of network constraints. Our approach is to divide the entire time period

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249

into a sequence of *intervals* (i) of equal duration, as shown in Fig. 1. If a demand is active for any amount of time within an interval, it is assumed to be active during the entire interval. The duration of an interval may be selected by the designer, based on expected traffic patterns and may be made as coarse or as fine as desired. For example, in our experiments we have set each interval to 15 minutes and the entire time period to be 12 hours or 24 hours. Clearly, a finer granularity allows greater control at the expense of increased computation. The formulation presented here allocates each demand within its time window (separating them into non-continuous segments if necessary) and performs RWA for each demand that can be handled by the network.

A. Notation used

In our ILP formulations, we will use the following notation for input data:

- E_P: The set of directed edges in the physical topology, each edge representing a fiber in the network.
- V_P: The set of end-nodes in the network.
- s_q (d_q): The source (destination) end-node of demand q.
- n_q: The number of lightpaths required by demand q.
- α_q (ω_q): The start (end) time of the larger window of demand q during which the demand must be met, expressed in terms of the first (last) interval during which the demand may be active.
- τ_q: The holding time of demand q, expressed in terms of the number of intervals during which the demand is active (fractional values are rounded up) $0 < \tau_q \leq \omega_q - \alpha_q$.
- $\mathcal{Q}$: The set of all traffic demands. Each element $q \in \mathcal{Q}$ is represented as $(s_q, d_q, n_q, \alpha_q, \omega_q, \tau_q)$.
- $\mathcal{K}$: The set of channel numbers on each fiber.
- $\mathbb{R}$: The number of edge-disjoint routes through the physical topology to be considered for RWA between each ordered pair of end-nodes.
- $d^e_{r,q}$: A matrix where each cell is defined as follows:

$$d^e_{r,q} = \begin{cases} 1 & \text{if the } r^{th} \text{ physical route of demand } q \\ & \text{from } s_q \text{ to } d_q \text{ includes edge } e, \\ 0 & \text{otherwise.} \end{cases}$$

We also define a number of *binary* variables as follows:

- $b_{q,i} = \begin{cases} 1 & \text{if demand } q \text{ is active during the interval } i \\ 0 & \text{otherwise.} \end{cases}$
- $x^i_{r,q} = \begin{cases} 1 & \text{if the } r^{th} \text{ physical route is selected to} \\ & \text{route demand } q \text{ during interval } i, \\ 0 & \text{otherwise.} \end{cases}$
- $\beta^i_{k,q} = \begin{cases} 1 & \text{if channel } k \text{ is assigned to} \\ & \text{demand } q\text{, during the interval } i, \\ 0 & \text{otherwise.} \end{cases}$
- $y_q = \begin{cases} 1 & \text{if demand } q \text{ is accommodated,} \\ 0 & \text{otherwise.} \end{cases}$

We also define a number of *continuous* variables as follows:

- $\alpha^i_{e,q} = \begin{cases} 1 & \text{if demand } q \text{ uses edge } e \text{ during interval } i, \\ 0 & \text{otherwise.} \end{cases}$
- $\gamma^i_{k,e,q} = \begin{cases} 1 & \text{if channel } k \text{ on edge } e \text{ is assigned to} \\ & \text{demand } q\text{, during the interval } i, \\ 0 & \text{otherwise.} \end{cases}$

B. ILP formulation for non-Continuous Demands

$$\textbf{Maximize} \sum_{q \in \mathcal{Q}} y_q \tag{1}$$

If a demand q is accommodated, then $y_q = 1$. Hence, the objective function (1) maximizes the the number of demands that can be accommodated in the network, by maximizing $\sum_{q \in \mathcal{Q}} y_q$. It is also possible to assign a *weight* to each demand q, either in terms of its bandwidth requirement n_q, or a preassigned priority level p_q. In that case the objective function would be changed to Maximize $\sum_{q \in \mathcal{Q}} n_q \cdot y_q$ or Maximize $\sum_{q \in \mathcal{Q}} p_q \cdot y_q$

Subject to:

a) An active demand must use only one route during any specific interval i.

$$\sum_{r, 0 \leq r < \mathbb{R}} x^i_{r,q} = b_{q,i}, \qquad \forall q \in \mathcal{Q}, \forall i, \alpha_q \leq i < \omega_q \tag{2}$$

b) The value of $\alpha^i_{e,q} = 1$ only if the r^{th} route is used by the demand q during the interval i (i.e. $x_{r,q} = 1$) and includes edge e (i.e. $d^e_{r,q} = 1$) .

$$\alpha^i_{e,q} = \sum_{r, 0 \leq r < \mathbb{R}} x^i_{r,q} \cdot d^e_{r,q}, \qquad \forall q \in \mathcal{Q},\ \forall e \in E_P \tag{3}$$

Constraint (3) must be satisfied $\forall i,\ \alpha_q \leq i < \omega_q$.

c) The sum of the allocated intervals for each demand q equals to its holding time. A demand is allocated resources only if it is accommodated (i.e., $y_q = 1$).

$$\sum_{i, \alpha_q \leq i < \omega_q} b_{q,i} = y_q \cdot \tau_q, \qquad \forall q \in \mathcal{Q} \tag{4}$$

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249

d) Total channels used on a link must not exceed $|\mathcal{K}|$.

$$\sum_{q\in\mathcal{Q}} n_q \cdot \alpha^i_{e,q} \leq |\mathcal{K}|,\ \forall e \in E_P, \forall i, \alpha_q \leq i < \omega_q \tag{5}$$

e) No interval is active for demand q outside its larger window.

$$b_{q,i} = 0, \qquad \forall q \in \mathcal{Q},\ \forall i, i < \alpha_q,\ \forall i, i \geq \omega_q \tag{6}$$

f) Exactly n_q channels are allocated to demand q if it is active during interval i ($b_{q,i} = 1$), and no resources are allocated if it is inactive during interval i ($b_{q,i} = 0$).

$$\sum_{k\in\mathcal{K}} \beta^i_{k,q} = n_q \cdot b_{q,i},\ \forall q \in \mathcal{Q}, \forall i, \alpha_q \leq i < \omega_q \tag{7}$$

g) Set the value of $\gamma^i_{k,e,q}$ to either 0 or 1.

$$\beta^i_{k,q} + \alpha^i_{e,q} - \gamma^i_{k,e,q} \leq 1 \tag{8}$$

$$\beta^i_{k,q} \geq \gamma^i_{k,e,q} \tag{9}$$

$$\alpha^i_{e,q} \geq \gamma^i_{k,e,q} \tag{10}$$

Constraint (8) - (10) must be satisfied $\forall q \in \mathcal{Q},\ k \in \mathcal{K},\ \forall e \in E_P$ and $\forall i$ such that $\alpha_q \leq i < \omega_q$.

Constraints (8) - (10) are used to define the variable $\gamma^i_{k,e,q}$. If either $\beta^i_{k,q} = 0$ or $\alpha^i_{e,q} = 0$, then $\gamma^i_{k,e,q} = 0$. The value of $\gamma^i_{k,e,q}$ is set to 1 only if both $\beta^i_{k,q} = 1$ and $\alpha^i_{e,q} = 1$. Therefore, although $\gamma^i_{k,e,q}$ is a continuous variable, it acts as a binary variable by assuming values from $\{0, 1\}$, thus reducing the complexity of the ILP formulation.

h) A channel k, on link e, can be assigned to at most 1 demand during an interval i.

$$\sum_{q\in\mathcal{Q}} \gamma^i_{k,e,q} \leq 1 \tag{11}$$

Constraint (11) must be satisfied $\forall k \in \mathcal{K},\ \forall e \in E_P$ and $\forall i$ such that $\alpha_q \leq i < \omega_q$.

C. Handling Continuous Demands

As mentioned in the introduction, the non-continuous model may not be suitable for all applications. In such cases, the corresponding demands should not be split into multiple segments, but transmitted as one single segment. We refer to these demands as *continuous* demands, and discuss below how they can be handled by our formulation.

For a demand q with fixed setup and teardown times (fixed window model), no additional modifications are needed. As long as the start and end times (α_q and ω_q) satisfy the condition $\tau_q = \omega_q - \alpha_q$, the entire demand will be sent as a single segment.

For the sliding window model, the start time is not fixed, but once transmission starts we must ensure that all segments are sent in consecutive intervals, along the same route. For this case, we add the following binary variables and constraints.

- $st_{q,i} = \begin{cases} 1 & \text{if transmission of demand } q \text{ starts during the interval } i, \\ 0 & \text{otherwise.} \end{cases}$
- $x_{r,q} = \begin{cases} 1 & \text{if the } r^{th} \text{ physical route is selected to route demand } q, \\ 0 & \text{otherwise.} \end{cases}$
- $\beta_{k,q} = \begin{cases} 1 & \text{if channel } k \text{ is assigned to demand } q, \\ 0 & \text{otherwise.} \end{cases}$

The following constraints are needed to be added to the above formulation:

$$\begin{aligned}
\sum_{i,\alpha_q\leq i<\omega_q} st_{q,i} &= y_q,\ \forall q \in \mathcal{Q} && (12)\\
b_{q,i+j} &\geq st_{q,i},\ \forall q \in \mathcal{Q}, \forall i, \alpha_q \leq i < \omega_q, && (13)\\
&\quad \forall j, 0 \leq j < \tau_q &&\\
\sum_{r,0\leq r<\mathbb{R}} x_{r,q} &= y_q,\ \forall q \in \mathcal{Q} && (14)\\
x^i_{r,q} &\leq x_{r,q},\ \forall q \in \mathcal{Q}, \forall i, \alpha_q \leq i < \omega_q, && (15)\\
&\quad \forall r, 0 \leq r < \mathbb{R} &&\\
\sum_{k\in\mathcal{K}} \beta_{k,q} &= n_q \cdot y_q,\ \forall q \in \mathcal{Q} && (16)\\
\beta^i_{k,q} &\leq \beta_{k,q},\ \forall q \in \mathcal{Q}, \forall i, \alpha_q \leq i < \omega_q, && (17)\\
&\quad \forall r, 0 \leq r < \mathbb{R} &&
\end{aligned}$$

Constraint (12) ensures that there is exactly one starting interval for a demand q, which is routed over the network (i.e. if $y_q = 1$). Constrait (13) sets $b_{q,i} = 1$ for τ_q cosecutive intervals starting with $st_{q,i} = 1$; this ensures that the segments for demand q are sent continuously in τ_q consecutive intervals, starting from interval $st_{q,i}$. Constraints (14) - (15) ensure that all segments for demand q are sent along the same route $x_{r,q}$ during each interval, (16) - (17) ensure that all segments for demand q are sent using the same n_q channels during each interval.

Constraints (14) - (17) can be used in conjunction with non-continuous demands as well, ensuring all segments use the same route and channels, to reduce the chances of out-of-order segments arriving at the receiver.

IV. RWA HEURISTIC FOR NON-CONTINUOUS SCHEDULED DEMANDS

In this section we outline our heuristic for RWA for Non-continuous Scheduled Demands (H-NSD) without any wavelength conversion. For a given fiber network with a specified number of channels per fiber and a set of non-continuous scheduled demands, the heuristic allocates resources to carry as many demands as possible. We have used the following additional notation to describe the heuristic:

- $S_{e,i}$: The set of free channels (channels not allocated or temporarily locked to any lightpath) on link e at interval i.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249

- $\mathcal{P}_q$: The set of potential active intervals i, for demand q, where $\alpha_q \le i < \omega_q$.
- $\mathcal{A}_q$: The set of active intervals i, for demand q, where $\alpha_q \le i < \omega_q$.
- $\mathcal{O}_e$: The set of intervals during there may be insufficient channels available on link $e \in E_P$.
- $\mathcal{Q}_{alloc}$: The set of successfully allocated demands.
- $\mathcal{Q}_{unalloc}$: The set of unallocated demands.
- p_q^r: The set of edges included in the r^{th} route of demand q in the path from s_q to d_q.
- $p_q^{r_i}$: The set of edges included in the r^{th} route of demand q in the path from s_q to d_q during the interval i.
- $\mathcal{W}_{q,i}^r$: The set of free channels that can be allocated to the r^{th} route of demand q at interval i.
- $\mathcal{C}_{q,i}^r$: The maximum congestion (in terms of the number of channels) on any edge for the r^{th} route of demand q at interval i.
- $\alpha_{e,i}$: The set of channels (wavelengths) on each link $e \in E_P$ that have been allocated or temporarily locked for a lightpath at interval i.
- β_q: The set of intervals during which demand q uses a congested link.

The heuristic given below allows different segments of a demand to use different routes, but constraints similar to those in section III.C may be easily incorporated here as well. Using the above notations, we describe our approach in heuristic H-NSD.

In the heuristic, when allocating resource to a demand q, the following constraints must be met:

- The number of WDM channels allocated on a link does not exceed $|\mathcal{K}|$, in any interval i.
- The summation of the active intervals for a demand q equals its total demand holding time τ_q.

Initially, none of the demands have been allocated resources, so $\mathcal{Q}_{alloc} \leftarrow \emptyset$ and $\mathcal{Q}_{unalloc} \leftarrow \mathcal{Q}$. From line 3 to 16, we identify the set of intervals $\mathcal{P}_q$ during which demand q may be active, select a route and temporarily "lock" a set of n_q channels for demand q for each interval $i \in \mathcal{P}_q$. The selected route is the least congested route (based on currently available information) that can accommodate demand q. We also identify (lines 10-11) the intervals $\mathcal{O}_e$ during which there may be insufficient channels on link e. If $i \in \mathcal{O}_e$ then we say that link e is congested during interval i.

Heuristic: H-NSD

Input: $G = (V_P, E_P)$, $\mathcal{K}$, $\mathcal{Q}$ and a set of pre-computed $\mathbb{R}$ edge-disjoint paths over the physical topology between each pair of end-nodes.

Value returned: A set of successful demands, $\mathcal{Q}_{alloc}$, and an appropriate RWA for $\mathcal{Q}_{alloc}$.

1: $\mathcal{Q}_{alloc} \leftarrow \emptyset$; $\mathcal{Q}_{unalloc} \leftarrow \mathcal{Q}$.
2: while (*demand can be added to* $\mathcal{Q}_{alloc}$) do
3:   $\mathcal{O}_e \leftarrow \emptyset,\ \forall e \in E_P$; $\mathcal{P}_q \leftarrow \emptyset,\ \forall q \in \mathcal{Q}_{unalloc}$.
4:   for Each $i \in \mathcal{I}$ do
5:     for Each $q \in \mathcal{Q}_{unalloc}$ do
6:       if $\alpha_q \le i < \omega_q$ then
7:         for Each $r,\ \forall r, 0 \le r < \mathbb{R}$ do
8:           $\mathcal{W}_{q,i}^r \leftarrow \bigcap_{e \in p_q^r} S_{e,i}$.
9:           $\mathcal{C}_{q,i}^r \leftarrow \max_{e \in p_q^r}\{|\alpha_{e,i}| + n_q\}$.
10:         if $\mathcal{C}_{q,i}^r > |\mathcal{K}|,\ \forall r, 0 \le r < \mathbb{R}$ then
11:           $\mathcal{O}_e = \mathcal{O}_e \cup \{i\},\ \forall e \in p_q^r$.
12:         else
13:           Select route r_i during interval i such that
14:             the value of $\mathcal{C}_{q,i}^{r_i}$ is minimum.
15:           Lock n_q wavelengths from $\mathcal{W}_{q,i}^{r_i}$.
16:           $\mathcal{P}_q \leftarrow \mathcal{P}_q \cup \{i\}$
17:   for Each $q \in \mathcal{Q}_{unalloc}$ do
18:     for Each $i \in \mathcal{P}_q$ do
19:       if $i \in \bigcup_{e \in p_q^{r_i}} \mathcal{O}_e$ then
20:         $\beta_q \leftarrow \beta_q \cup \{i\}$.
21:     if $\sum_{i \in \mathcal{P}_q} i - \sum_{i \in \beta_q} i \ge \tau_q$ then
22:       $\mathcal{Q}_{alloc} \leftarrow \mathcal{Q}_{alloc} \cup \{q\}$.
23:       $\mathcal{Q}_{unalloc} \leftarrow \mathcal{Q}_{unalloc} \backslash q$.
24:       Select $\mathcal{A}_q$ from $\mathcal{P}_q \backslash \beta_q$ such that $\sum_{i \in \mathcal{A}_q} i = \tau_q$.
25:       Release unused resources for demand q
26: if $|\mathcal{Q}_{unalloc}| > 1$ then
27:   for Each $q \in \mathcal{Q}_{unalloc}$ do
28:     $\mathcal{A}_q \leftarrow \mathcal{P}_q \backslash \beta_q$.
29:     $\mathcal{P}_q \leftarrow \emptyset$.
30:   $\mathcal{Q}_{sort} \leftarrow$ demands in $\mathcal{Q}_{unalloc}$, sorted based on
31:       either criteria 1 or criteria 2.
32:   for Each $q \in \mathcal{Q}_{sort}$ do
33:     for Each $i,\ \alpha_q \le i < \omega_q$ do
34:       if $i \notin \mathcal{A}_q$ then
35:         Do steps from line 7 to 16, except the statement
36:             in line 16 is replaced by $\mathcal{A}_q \leftarrow \mathcal{A}_q \cup \{i\}$.
37:         if $\sum_{i \in \mathcal{A}_q} i = \tau_q$ then
38:           $\mathcal{Q}_{alloc} \leftarrow \mathcal{Q}_{alloc} \cup \{q\}$.
39:           $\mathcal{Q}_{unalloc} \leftarrow \mathcal{Q}_{unalloc} \backslash q$.
40:           *Break;*
41:     if $q \notin \mathcal{Q}_{alloc}$ then release all resources for demand q.
42: return $(\mathcal{Q}_{alloc})$.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249

In the next phase (lines 17 - 25), we first identify the intervals β_q during which a demand q may need to use channels on a "congested" link e (lines 18 - 20). Then (lines 21 - 24), we allocate resources to those demands which can be accommodated using only those intervals during which the congested links can be avoided, i.e., the number of such intervals ($\sum_{i \in \mathcal{P}_q} i - \sum_{i \in \beta_q} i$) is greater than or equal to the demand holding time τ_q. Finally, any locked resources for these demands are released (line 25). This whole process (lines 2- 25) continues until no more demands can be allocated using "uncongested" links only.

Finally, the remaining demands (if any) are processed using a "greedy" scheme (lines 26 - 41). The demands are first sorted, based on either criteria 1 or criteria 2 as follows:

- **criteria 1**: the unallocated demand set $\mathcal{Q}_{unalloc}$ is sorted in ascending order of the number of intervals needed.
- **criteria 2**: the unallocated demand set $\mathcal{Q}_{unalloc}$ is sorted in descending order of the demand priority. The priority of a demand q is calculated as Priority of demand q = (*number of intervals still needed for demand q*)/(*number of intervals remaining until* ω_q).

Then, each demand from the sorted list $\mathcal{Q}_{sort}$ is processed one by one and resources are allocated to the demand if available. If the demand traverses a congested link during interval i, and a sufficient number of channels are available, these are allocated to the demand; otherwise the demand cannot be accommodated and all resources for this demand (during all other intervals) are released (line 41). The algorithm returns a set of demands Q_{alloc} that contains all the demands that can be handled under the given resource constraints.

V. Experimental Results

We discuss our experimental results in this section. We have tested our approach on different size networks, varied from 6 to 20 nodes, and for each size network, we varied the number of demands from 32 to 300. The demand sets were constructed randomly such that each demand in a demand set was assigned a value for the holding time between 3 and 6 hours (inclusive). For each size of demand set, 5 sets of demands were randomly generated and tested using different schemes. Each individual result presented here are the averages of these 5 sets.

First, we present the performance of our ILP formulations, which were able to produce results on small size networks (up to 6 nodes). The results were obtained using ILOG CPLEX solver [13]. Table I summarizes the results of the ILP formulation (with wavelength converters) on a 6 nodes network. For these experiments, the number of demands were varied from 32 to 300 (indicated in the column $|\mathcal{Q}|$), and the number of channels were varied from 4 to 16 (indicated in the column $|\mathcal{K}|$). The column "Fixed" indicates the total number of demands that were accommodated by using the the *fixed* window approach, under the demand set and the number of channels given in the corresponding row. The columns "NSD +2hrs", "NSD +4hrs" and "NSD +6hrs", indicate the total number of demands accommodated, while using our ILP approach on non-continuous sliding window traffic model, with the window size extended by 2 hours, 4 hours and 6 hours, respectively, over the fixed window demand holding time, for each demand. As shown in the table, the ILP outperforms the fixed window approach in all cases. Also, the scheme was able to accommodate more demands as the window size is increased, which was expected.

TABLE I

Comparison of total number of demands accommodated using the fixed window approach and the ILP formulation (with wavelength converters) under different window size.

		# of demand accommodated under the scheme			
$\lvert\mathcal{Q}\rvert$	$\lvert\mathcal{K}\rvert$	Fixed	NSD +2hrs	NSD +4hrs	NSD +6hrs
32	4	24.8	29.2	31.2	31.6
50	4	32.6	39.4	42.2	43
80	8	64.2	74.4	77.2	78.2
100	8	76.8	88.2	92	94.2
150	16	126.8	141.8	146.8	149
200	16	156.6	174.6	182	187
250	16	181.2	204.2	212.6	218.8
300	16	205.8	236.4	246.6	253.6

Now we present the performance of the heuristic H-NSD on a 14-node NSFNET and a 20-node ARPANET network [14]. For these experiments, the number of demands were varied from 100 to 300. For each demand set, we have compared the performance of our approach (using H-NSD) to both the *fixed* window approach (denoted by "Fixed"), and the traditional *sliding* window approach (adapting the algorithm in [11]), with the window size extended by 2hrs, 4hrs and 6hrs (denoted by "Sliding +2hrs", "Sliding +4hrs" and "Sliding +6hrs", respectively) over the fixed window demand holding time for each demand.

Fig. 2 compares the *percent improvement* for the number of demands successfully handled by the heuristic H-NSD over Fixed, Sliding +4hrs and Sliding +6hrs schemes, on a 14-node network, under different load and resource constraints. The first group is the results with 100 demands and 8 channels (indicated as "100x8" in the label. Others labels follow the same convention). In each group of bars (corresponding to a specific demand size and link capacity), the first bar indicates the percent improvement over Sliding +4hrs approach, followed by the improvement over Sliding +6hrs approach, which is followed by the percent improvement over fixed window approach. For this set of results, the criteria 2 has been used for sorting the unallocated demands (results with criteria 1 follow a similar pattern). Fig. 3 compares the *percent improvements* on a 20-node network, when criteria 1 has been used for sorting the unallocated demands (results with criteria 2 follow a similar pattern).

As shown in Figs. 2 and 3, our approach consistently outperforms existing *connection holding time aware* approaches. The improvements range from 10% to 30%, and vary depending on the available resources and load on the network. As expected, the improvements over the fixed window model are significantly higher compared to traditional sliding window models. We observe that sorting by criteria 1 performs better in the case of less number of available channels per fiber and

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249

smaller window size, while the other provides better results when the number of channels per fiber and window size are increased.

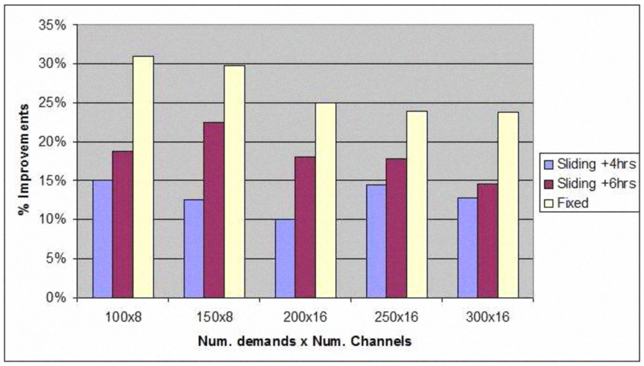

Fig. 2. Comparison of the percentage improvements using H-NSD over other approaches on a 14-node network when criteria 2 was used for sorting.

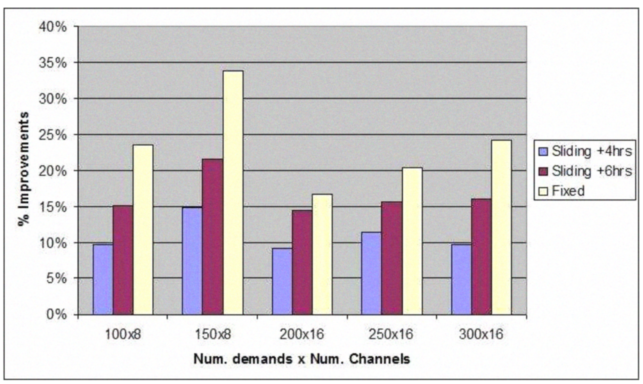

Fig. 3. Comparison of the percentage improvements using H-NSD over other approaches on a 20-node network when criteria 1 was used for sorting.

VI. Conclusions

In this paper, we have introduced a new *non-continuous* scheduled traffic model, which is more flexible than the existing sliding window model and can be used for a number of applications requiring periodic use of bandwidth. We have proposed an ILP formulation that optimally schedules demands (in time), divides a demand into multiple segments to be sent separately, if necessary and performs RWA for each scheduled demand. We have also shown how the traditional fixed and sliding window models can be treated as a special case of our formulation and presented a simple heuristic that can be used for larger networks with a high number of demands. It has already been well established in the literature that connection holding time aware algorithms offer significant advantages compared to holding time unaware strategies. In this paper, we show that our model can lead to additional improvements over existing holding time aware models. Simulation results show an average improvements of around 25% and 13% over the fixed window and sliding window models respectively. We are currently investigating ways to extend the formulation by incorporating higher layer costs and penalizing excessive segmentation of demands.

Acknowledgment

The work of A. Jaekel has been supported by research grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro and R. Douville. Routing and wavelength assignment of scheduled lightpath demands. IEEE JSAC, vol. 21, no. 8, pp. 1231-1240, Oct. 2003.
[2] B. Wang, T. Li, X. Luo, Y. Fan and C. Xin. On service provisioning under a scheduled traffic model in reconfigurable WDM optical networks. BROADNETS 2005, PP. 15-24, Oct. 2005.
[3] W. Su and G. Sasaki. Scheduling of periodic connections with flexibility. in 41st Allerton Conf. on Communications, Control and Computing, Oct. 2003.
[4] T. Li and B. Wang. On optimal survivability design in WDM optical networks under a scheduled traffic model. DRCN05, pp. 23-30, Oct. 2005.
[5] A. Jaekel and Y.Chen. Quality of service based resource allocation for scheduled lightpath demands. Computer Communications, vol. 30, no. 18, pp. 3550-3558, Dec. 2007.
[6] B. Wang, T. Li, C. Xin, and X. Zhang. On survivable service provisioning in WDM networks under a scheduled traffic model. GLOBECOM 2005, pp. 1900-1904, Nov.-Dec. 2005.
[7] T. Li and B. Wang. On optimal survivability design under a scheduled traffic model in wavelength routed optical mesh networks. 4th Annual Communication Networks and Services Research Conference (CNSR06), 2006.
[8] T. Li and B. Wang. Path protection based routing and wavelength assignment in wavelength division multiplexing optical networks under a scheduled traffic model. J. of Optical Networking, vol. 5, pp. 575-588, 2006.
[9] M. Tornatore, C. Ou, J. Zhang, A. Pattavina, and B. Mukherjee. Efficient shared-path protection exploiting the knowledge of connection-holding time. OFC 2005, pp. 65-72, Mar. 2005.
[10] H. Zang, J.P. Jue and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. Optical Networks Magazine, vol. 1, pp. 47-60, Jan. 2000.
[11] A. Jaekel and Y. Chen. Resource Provisioning for Survivable WDM Networks under a Sliding Scheduled Traffic Model. Optical Switching and Networking, Vol. 6, No. 1, pp.44-54, Jan. 2009.
[12] J. Zheng, B. Zhang and H. T. Mouftah. Towards automated provisioning of advance reservation service in next-generation Internet. IEEE Communications Magazine, Vol. 44, No. 12, pp. 68-74, Dec. 2006.
[13] Ilog cplex 9.1. Documentation available at http://www.columbia.edu/dano/resources/cplex91man/index.html.
[14] M. Sridharan, M.V. Salapaka, A. Somani. A practical approach to operating survivable WDM networks. IEEE JSAC, Vol. 20, No. 1, pp. 34-46, Jan. 2002.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7249
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7249