Dynamic Circuits with Lightpath Switching over Wavelength Routed Networks

Neal Charbonneau and Vinod M. Vokkarane
Department of Computer and Information Science, University of Massachusetts, Dartmouth, MA
Email: {ncharbonneau,vvokkarane}@ieee.org

***Abstract*—In this paper we examine provisioning holding-time-aware dynamic circuits using a technique called lightpath switching (LPS). Instead of using the same lightpath for the duration of the data transmission, in LPS we allow a request to switch lightpaths over time. Data transmission may begin on one lightpath from the source to destination, then at a later time a different lightpath from the source to the destination may be selected to continue data transmission. The lightpath switches are transparent to the user and are managed by the network. Allowing LPS creates a number of segments that can use independent lightpaths. We compare the performance of traditional routing and wavelength (RWA) assignment to routing and wavelength assignment with LPS. We show that LPS can significantly reduce blocking compared to traditional RWA.** [1]

***Index Terms*—WDM, RWA, holding-time-aware lightpath**

I. INTRODUCTION

Optical wavelength-routed WDM networks are a potential candidate for future wide-area backbone networks as well as scientific Grid networks. In WDM networks, each fiber is partitioned into a number of wavelengths, each of which is capable of transmitting data simultaneously. An optical WDM network consists of fibers connected by switches, or optical cross connects (OXCs). In order to transmit data over the network, a dedicated circuit is established when a user submits a connection request. When a connection request arrives at the network, the request must be routed over the physical topology and also assigned a wavelength. This is known as the *routing and wavelength assignment* (RWA) problem. The combination of a route and wavelength is known as a *lightpath*. The RWA problem is NP-complete so heuristics are typically used. As requests are accepted into the network, no two requests can use the same wavelength on the same link. As more requests arrive over time new lightpaths must be allocated as long as there are enough wavelengths to establish them. If an arriving request cannot find a lightpath, the request is rejected and it is said to be blocked. In an all-optical WDM system, once a path is setup, the signal is transmitted all-optically through the network.

A common traffic model for WDM networks is the dynamic traffic model. Requests are assumed to arrive sequentially, according to a stochastic process, and have finite holding times. The goal is to minimize request blocking, where a user's request is denied due to lack of resources. We can consider two types of dynamic models. One is dynamic with unknown duration, where each request uses network resources for an unspecified amount of time, and the other is dynamic with known duration where requests specify a holding time when they arrive. This is also known as *holding-time-aware* (HTA) traffic [1]. There are classes of applications that are able to specify holding times, such as video distribution and large file transfers. This extra information allows the network to better optimize its resources and increase efficiency.

In this work, we consider holding-time-aware traffic with *lightpath switching* (LPS). With LPS, a series of lightpath switches occur during the request's duration. For example, a request may use some lightpath x from time t_1 to t_5, then switch to a different lightpath y from time t_5 to t_8. We note, this is not the same as multihop routing. We still use single-hop routing, but the physical lightpath connecting the source to destination changes temporally. It is already established that HTA can improve the performance of traditional dynamic traffic with unknown durations [1]. Here we show that LPS can further improve the performance of HTA demands. We discuss the related work in Section II. We formally define the problem in Section III. Our LPS heuristic is proposed in Section IV. We present our performance evaluation in Section V and finally conclude in Section VI.

II. RELATED WORK

In this section we provide a brief overview of the related work. There are a number of papers on HTA traffic, among others see [1], [2]. Advance reservation, or scheduled demands, is related to HTA traffic. With advance reservation, in addition to specifying a holding times, the requests book-ahead, i.e., they reserve network resources in-advance for use at a later time [3], [4].

To the best of our knowledge, we are the first to propose *lightpath switching* for HTA demands. Path switching for flexible advance reservation in electronic networks is proposed in [5]. In this work we must consider both routing *and* wavelength assignment (with wavelength continuity). Non-continuous advance reservation was proposed in [6]. They consider the *static* traffic problem where all of the demands are given and each demand may be broken into smaller segments that can use different lightpaths. The authors of [7] proposed flexible reservations that could be segmented. They do not consider routing and wavelength assignment. They approach the problem strictly as a scheduling problem of bandwidth and assume the lightpaths are already established. The above work is for advance reservation, not HTA demands.

III. PROBLEM DEFINITION

In this section we formally define the problem. We will discuss the network assumptions used in our problem definition in the following subsection. We consider the case of dynamic traffic where user requests arrive according to some stochastic process. We are given a network, $G = (V, E, W, H)$, where V is the set of nodes, E is the set of edges, and W is the fixed number of wavelengths. We assume a time-slotted network with fixed-size timeslots. We define the *horizon*, H, to be the number of future timeslots for which state information is maintained. This value will limit how large holding times can be. There is a centralized scheduler that maintains the

---

[1] This work was supported in part by the National Science Foundation (NSF) under grant CNS-0626798 and Department of Energy (DOE) COMMON project under grant DE-SC0004909.

state information, which is updated for every new request. The state information consists of which timeslots are used on all wavelengths on all edges. It can be thought of as a three-dimensional array $U[E, W, H]$. The user requests, R, can be defined as a three-tuple, (s, d, τ), where $s \in V$ is the source node, $d \in V$ is the destination, and τ is the duration in timeslots. Upon arrival of a dynamic circuit request, the scheduler must allocate resources to the request. The scheduler will then return a vector of *segments*, S, called the *schedule*. We define a *segment* as a lightpath used to transfer data between a specified start and end time. A segment can be defined as a four-tuple, (t, d, P, W), where t is the start time, d is the duration, P is the path, and W is the wavelength. We assume that t refers to a specific timeslot and d is specified in number of timeslots. The start time is inclusive, so the segment transmits data from $[t, t+d-1]$. Each segment follows the wavelength continuity constraint on all links. Each new segment constitutes a lightpath-switch.

Instead of generating a single route and wavelength for a given request as in traditional RWA problems, our heuristics can generate a schedule of one or more segments. We will call this *routing, wavelength, and segment assignment* (RWSA).

Definition *RWSA*: Given a network, $G = (V, E, W, H)$, its current state, $U[E, W, H]$, and an incoming request, $R = (s, d, \tau)$, we must return a *schedule*, $S = \{(t_i, d_i, P_i, W_i)\}$ if the request can be accommodated, or $BLOCKED$ otherwise. The segments should be selected in a way that they reduce blocking of future requests.

We have the following constraints for the schedule, S:

$$1 \leq |S| \leq \tau. \quad (1) \qquad t_1 = t_{now}. \quad (2)$$

$$\sum d_i = \tau. \quad (3) \qquad t_i + d_i = t_{i+1}. \quad (4)$$

Assume the schedule has n elements. (1) specifies there must be at least one segment and there can be at most τ segments. (2) states that the first segment must start when the request arrives. (3) states that the summation of the segment durations is equal to the request's duration. (4) states that each segment must start when the previous one ends

A. Network Architecture and Assumptions

We consider HTA demands over Grid networks to support e-Science applications. We assume that there are no wavelength converters in the network, so any given segment must use the same wavelength on all links. It is possible, however, that different segments of a request can use different wavelengths.

We assume the network is under centralized control by a network resource manager. This assumption is reasonable in the case of Grid networks because of the relatively small size. For more general networks, we assume the scheduler is implemented at the domain level for scalability. We consider the time-domain to be broken into discrete timeslots of fixed-size. There is no need for synchronization between the networking elements because the scheduler controls the directly. The storage complexity for the network state, U, is $\Theta(EWH)$, which can be stored as a bit-vector.

A common architecture for Grids is to have a centralized resource broker that provides APIs for Grid applications or Grid middleware [8]. The lightpath switching is controlled by the resource broker. The API between the resource broker and grid application that can inform the application when it should transmit on different lightpaths. We assume that users request a single wavelength of bandwidth. The user (or client application) will determine how many timeslots are required given this bandwidth request.

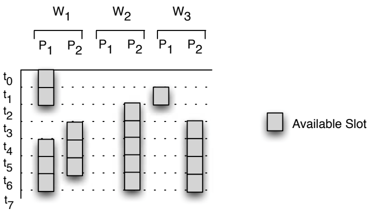

Fig. 1. Representation of the timeslot availability of different lightpaths. We assume that there are two precomputed paths (P_1 and P_2) and three available wavelengths. This leads to a total of six lightpaths, where each lightpath is either available or unavailable for any of the timeslots.

There will be a small overhead for requests that are segmented because of the time it takes to reconfigure the OXCs. We assume that the switching can be done in sub-second time. The actual reconfiguration of OXCs can be done using existing protocols, such as RSVP-TE [2]. We note that our algorithms are independent of the timeslot duration.

IV. LIGHTPATH SWITCHING

In this section we discuss RWSA with static route computation. For each source-destination pair, we have precomputed k-shortest-paths using Yen's algorithm [9], which finds k loopless paths (not necessarily disjoint). We have a total of $W * k$ lightpaths for each request and τ timeslots that can be used on any of the lightpaths for any given request. We can visualize the state information as shown in Fig. 1, where shaded blocks represent the available timeslots.

In the figure, assume there are two pre-computed paths (P_1 and P_2) and three wavelengths (W_1, W_2, W_3) in the network for a total of six lightpaths between some source-destination pair. This can be computed for each arriving request based on the wavelength and timeslot availability information stored in the scheduler. In this example, let the transmission window starts at $\alpha = t_0$ and $\tau = 7$. From the figure, there is no single lightpath available starting at t_0 that is also available for seven slots. With lightpath switching, however, we could select P_1 on W_1 during timeslots $[t_0, t_2)$, and P_2 on W_2 during $[t_2, t_7)$, creating schedule $S = \{(t_0, 2, P_1, W_1), (t_2, 5, P_2, W_2)\}$. Lightpath switching allows us to provision this request that would otherwise be blocked.

Algorithm 1: All-Segments (AS)

```
Input: $R = (s, d, \tau)$, $G = (V, E, W, H)$, $U[E, W, H]$
Output: Schedule, $S = \{(t, \tau, P, W)\}$
$schedule = \phi$
for $w = 1$ to $W$ do
    for $k = 1$ to $K$ do
        if $available(P_k, w, t_{now}) \geq \tau$ and lowest index then
            $schedule = (t_{now}, \tau, P_k, w)$
return $schedule$
```

A. HTA reservation with no Lightpath Switching

In order to evaluate the performance of lightpath switching, we compare our proposed heuristics with a simple HTA

---

[2] RFC 3209

heuristic that does not allow lightpath switching. The heuristic scans all path and wavelength combinations and records all of the available segments with duration of τ. If there are multiple segments found, it will select the segment on the lowest index wavelength. The algorithm, All-Segments (AS), is shown in Algorithm 1. The *available* function determines how many consecutive slots are available starting at the current time (t_{now}) on the specified lightpath.

Determining the number of consecutive slots that are available takes $O(V\tau)$. The complexity of AS is then $O(WkV\tau)$.

B. Lightpath Switching (LPS)

In this section we propose a lightpath switching heuristic (LPS) that fills voids on wavelengths in increasing order of wavelength index. LPS starts with the lowest-index wavelength and scans it for unused slots, which are turned into segments. Once all paths on the current wavelength are scanned, it moves to the next higher index wavelength again looking for unused slots that do not overlap in time with previously selected slots and adds these to the schedule. The algorithm maintains a list of currently selected segments in the $schedule$. Because we assume simultaneous transmission on multiple lightpaths is not possible, any new segments that will be added to $schedule$ cannot overlap in time with anything currently in $schedule$. The *findFreeTimes* function on line 4 (Algorithm 2) returns the gaps in time between segments already in $schedule$. For example, if the current schedule is $S = \{(t_3, 2, P_1, W_1), (t_5, 2, P_1, W_1)\}$, the free times are $[\alpha, t_3), [t_7, \alpha + \tau)$. The algorithm scans these free times for unused slots on the next wavelength, adding new segments to $schedule$ as it finds them.

Algorithm 2 Lightpath Switching (LPS)

```
  Input: $R = (s, d, \alpha, \tau, \omega)$, $G = (V, E, W, H)$, $U[E, W, H]$
  Output: Schedule, $S = \{(t_i, d_i, P_i, W_i)\}$
1 $schedule = \phi$
2 for $w$ in $W$ do
3     for $k$ in $K$ do
4         $validTimes = findFreeTimes(schedule)$
5         for $v$ in $validTimes$ do
6             find segments for $P_k, w$ between $[v.start, v.end]$
7             insert segments into $schedule$
8 return $schedule$
```

For each lightpath, the *findFreeTimes* function can execute in $O(\tau)$, since there are at most τ segments in the schedule. There are also at most τ valid timeslots to scan on at most V links of the path. This leads to a runtime of $O(WkV\tau)$, which is the same as the AS heuristic.

V. PERFORMANCE EVALUATION

We now provide simulation results for our proposed heuristics. We use the following parameters. The arrival process is a Poisson process. The holding time distribution is exponential. The horizon is large enough so that no requests are blocked due to their duration. The selection of timeslot size is beyond the scope of this paper and largely depends on the type of traffic a network operator expects.

Our primary performance metric will be blocking probability, which is defined as the ratio of the number of blocked requests to the total number of requests. We simulate 10^6 requests and take the average of ten runs. We evaluate our heuristics on the 14-node NSFnet. The results are similar for other networks. We use $k = 3$ pre-computed shortest paths.

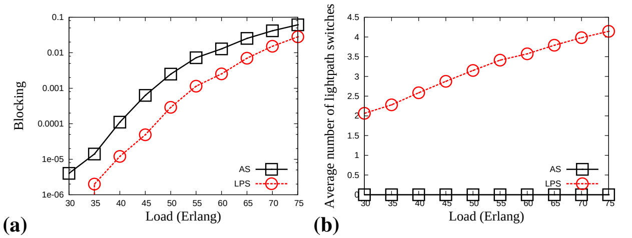

Fig. 2. (a) blocking probability vs. load. (b) Average number of lightpath switches vs. load.

We show the performance of the two heuristics in Fig. 2. Fig. 2(a) shows that the blocking probability is significantly reduced when lightpath switching is allowed. Fig. 2(b) shows the average number of lightpath switches that occur for each request. The mean holding time for each request is twelve timeslots. The number of lightpath switches ranges from two to four, meaning each segment is on average four timeslots at low loads and over two timeslots at higher loads. These results show there is a tradeoff between reduced blocking and increased network signaling (number of lightpath switches).

We also ran simulations for different k values and for different networks. With $k = 1$, the performance improvement between AS and LPS is not as significant because here only wavelength switching would be possible. For all other values of k, the relative improvement of LPS to AS is about the same. As the value of k increases, the number of lightpath switches also increases while the blocking probability decreases with LPS. These results are not included due to space limitations. We also note that depending on the network's average nodal degree, there is a maximum value of k for which no further improvement occurs.

VI. CONCLUSION

In this paper we propose lightpath switching for holding-time-aware demands. We show that allowing a request to use multiple lightpaths over its duration can significantly decrease blocking probability. Areas of future work include dynamic routing heuristics as well as distributed signaling techniques.

REFERENCES

[1] M. Tornatore *et al.*, "Holding-time-aware dynamic traffic grooming," *IEEE JSAC*, vol. 26, no. 3, pp. 28 –35, Apr. 2008.

[2] D. Lucerna *et al.*, "Dynamic routing of connections with known duration in WDM networks," in *IEEE GLOBECOM*, Nov. 2009, pp. 1 –7.

[3] J. Kuri *et al.*, "Routing and wavelength assignment of scheduled lightpath demands," *IEEE JSAC*, vol. 21, no. 8, pp. 1231–1240, Oct. 2003.

[4] J. Zheng *et al.*, "Toward automated provisioning of advance reservation service in next-generation optical Internet," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 68–74, Dec. 2006.

[5] R. Cohen *et al.*, "Path switching and grading algorithms for advance channel reservation architectures," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1684 –1695, Oct. 2009.

[6] Y. Chen *et al.*, "Resource allocation strategies for a non-continuous sliding window traffic model in WDM networks," in *Proceedings, ICST BroadNets*, Sep. 2009, pp. 1 –7.

[7] S. Naiksatam *et al.*, "Elastic reservations for efficient bandwidth utilization in LambdaGrids," *Future Gener. Comput. Syst.*, vol. 23, no. 1, pp. 1–22, 2007.

[8] C. Guok *et al.*, "A user driven dynamic circuit network implementation," in *IEEE GLOBECOM Workshop*, Dec. 2008, pp. 1 –5.

[9] J. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.