Dynamic On-demand Lightpath Provisioning Using Spectral Defragmentation in Flexible Bandwidth Networks

Ke Wen, Yawei Yin, David J. Geisler, Shuo Chang, S. J. B. Yoo
Department of Electrical and Computer Engineering, University of California, Davis, One Shields Ave., Davis, California 95616, USA
{kwen, yyin, djgeisler, suchang, sbyoo}@ucdavis.edu

Abstract: We propose a spectral defragmentation algorithm and corresponding node architectures with wavelength conversion to support on-demand lightpath provisioning in flexible bandwidth optical networks. Simulation results demonstrate six-fold reductions in blocking rates with defragmentation.

OCIS codes: (060.4264) Networks, wavelength assignment; (060.1155) All-optical networks

1. Introduction
Exponentially increasing demand for communication capacity is already approaching the limit of single mode fiber capacity in many links around the world. At the same time, dynamically varying traffic demand is requiring an efficient and agile utilization of the optical spectrum. Flexible bandwidth networking emerged recently as a promising paradigm for assigning elastic spectral bandwidth to traffic demands with various modulation formats and spectral efficiencies [1]. Current research includes use of many low-speed, orthogonal subcarriers to generate super-wavelength and sub-wavelength connections. While existing spectrum allocation algorithms for flexible bandwidth networks use static planning models with the assumption that all the connection requests are known a priori and are constant over time [2], there is an increasing need for rapid and dynamic end-to-end bandwidth (lightpath) provisioning due to the inherent dynamic nature of real network demands. This requires intelligent optical networks that can find a block of contiguous spectra available on all links of the desired lightpath in real time. However, inevitable spectral fragmentation (Fig.1(a)) in flexible bandwidth networks increases the complexity of dynamic spectrum allocation and results in a high blocking probability. Fig.1(b) shows a typical case where a new connection request (blue) from node A to node E finds available spectrum on all links in its path except for link 3, which requires spectral defragmentation at node C to satisfy this new connection request.

This paper proposes a spectral defragmentation algorithm to enable on-demand connection requests in flexible bandwidth networks. It also presents two scalable wavelength cross connect architectures capable of spectral defragmentation and demonstrate significant reductions in blocking probabilities by spectral defragmentation.

Fig.1 (a) Scattered spectral fragments in five links of NSFNET in a flexible bandwidth networking scenario. (b) A typical case showing when spectral defragmentation is needed. (c) Pre-switch and (d) post-switch flexible bandwidth wavelength cross connect architectures with defragmentation capability. Example shows the defragmentation operations in node C in (b).

2. Enabling Technology and Switch Node Architectures
Spectral defragmentation can be realized by four-wave mixing (FWM) based wavelength conversion [3]. As Fig 1 indicates, network nodes equipped with FWM elements and wavelength selective switches (WSS) support spectral defragmentation functionality. With $M$ parallel FWM convertors, a link has the capability to convert $M$ connections’ spectra simultaneously. Spectral defragmentation can be performed in either a pre-switch (Fig.1(c)) or post-switch (Fig.1(d)) fashion. The pre-switch architecture (Fig.1(c)) requires that defragmentation will not result in any spectrum overlap on both the output link and the fiber between the $M \times 1$ WSS and the $N \times N$ WSS. In the post-switch architecture (Fig.1(d)), if satisfying a new request requires spectral defragmentation, the new request is routed to the
lower port of the corresponding output link. After defragmentation is performed on the upper link, the required spectrum is freed. Since the non-overlapping constraint must be satisfied only on the output link during the defragmentation process, the post-switch architecture provides more flexibility for spectrum rearrangements. However, this flexibility requires an $N \times 2N$ WSS as well as a sufficient number of FWM elements on a single link, in order to perform multiple conversions simultaneously; while in the pre-switch case, these operations are distributed to multiple input links.

3. Spectral Defragmentation Algorithms

3.1 Minimum Cost Search

With spectral defragmentation, the spectrum of a link can be rearranged and spectral fragments “glued” into a continuous block, thus freeing continuous swaths of unallocated bandwidth to serve new connection requests. However, using different spectral ranges to embed the new lightpath may result in different amounts of existing connections to be moved out of the “Evacuate zone” (Fig.2(a)) as well as different amounts of defragmentation operations needed. It is straightforward and reasonable to set the cost of spectral defragmentation as the number of connections affected and the objective of finding the best spectrum range as minimization of this cost. Here, we use greedy minimum-cost search to find the best range: Initialize the starting frequency $f_s$ of the new lightpath as 0 (measured by subcarrier index); Count the number of existing connections affected if the new lightpath were to be embedded in $[f_s, f_s+w]$, where $w$ is the bandwidth required by the new connection. Increase $f_s$ by 1 each time, until $f_s = f_{\text{max}} - w$, where $f_{\text{max}}$ means the maximum capacity of a link. Starting from the $f_s$ with the smallest number of affected connections, verify whether these affected connections can be moved to alternative spectrum blocks with the available defragmentation capability.

![Fig.2 (a) Affected connections ($C_1, C_2, C_3$) and their possible alternative spectrum blocks (in colored circles). The evacuate zone is the spectral range to be freed via defragmentation to enable a new request. (b) Bipartite matching graph for scenario in (a). (c) 14-node NSFNET topology.](image)

3.2 Search and Verification of Spectra for the Affected Connections

When attempting to fit in a new lightpath within a candidate spectrum $[f_s, f_s+w]$, all connections affected by defragmentation operations must not be blocked. If any existing connection is blocked, then $f_s$ cannot be used. In other words, a one-to-one matching between the affected connections and the alternative void spectrum blocks is needed. During this process, there are two main constraints. First, existing connections may be affected on multiple links, in which case the number of wavelength conversions for a particular connection needs to be minimized to maximize signal integrity. Second, to ensure non-overlapping spectral allocation for all affected connections, the control plane needs to handle all necessary defragmentation decisions jointly due to their mutual influence. These two criteria make the problem a “chain reaction” and is in fact an integer linear programming (ILP) problem, which is generally NP-hard. In this case, we can reduce the defragmentation scenario to a graph problem called bipartite matching [4]. This kind of problem can be solved by any maximum network flow algorithm (e.g., the Ford-Fulkerson algorithm [5]) in $O(\mathbf{E} \cdot \mathbf{C})$ time, where $\mathbf{E}$ is the number of edges in the graph and $\mathbf{C}$ is the maximum cardinality of matching.

In bipartite matching problems, the matching objects are modeled as two columns of nodes (each having $n_L$ and $n_R$ nodes). A directed edge is formed if a node in the left column can become a partner with a node in the right column. The matching constraint is that any node, either in the left column or the right column, can have at most one partner. The goal is to find the maximum-cardinality matching between the two columns of nodes. If the cardinality is equal to the minimum of $n_L$ and $n_R$, a perfect matching is found. The defragmentation problem can be mapped into a bipartite matching problem as follows:

**Step (i):** For each affected connection $C_i$, from its own perspective, find alternative spectrum blocks (colored circles in Fig.2(a)) which are commonly available on all touched links. The set of available blocks is denoted as $B_i$. 

Mo.2.K.4.pdf 2

7/27/2011 4:18:23 PM  
ECOC Technical Digest © 2011 OSA
Step (ii): If any two spectrum blocks in two different sets have any overlap, then merge them together and add it into a new set $B_{total}$. If a block does not overlap any, it is still added into $B_{total}$. A common property of the blocks in $B_{total}$ is that each element can be assigned to only one connection at one time. At this point, the defragmentation problem becomes finding the best match between the affected connections and the spectrum blocks in $B_{total}$.

Step (iii): Construction of the bipartite matching graph $G$: represent each affected connection by a node in the left column and each spectrum block in $B_{total}$ by a node in the right column. If it is possible for a connection $C_i$ to fit in a block $b_k \in B_{total}$, then add a directed edge from $C_i$ to $b_k$. Add source node $S$ and sink node $T$. Connect $S$ with each node in the left column by a directed edge with a capacity of 1, for the guaranty that each connection will pick up only one block in the right column as its new habitat. Connect each node in the right column with $T$ by a directed edge with a capacity of 1, for the guaranty that each block will be used by only one connection.

Step (iv): Run any maximum network flow algorithm on graph $G$ to find the maximum flow from $S$ to $T$. Here, the value of each flow is restricted to non-negative integers. If the value of the total flows out of $S$ equals to the number of nodes in the left column, then a perfect matching is found. The flow paths (red arrows in the Fig.2(b)) indicate the selection of non-overlapping spectrum blocks for each affected connection. Otherwise, if the value of the total flows out of $S$ is smaller than the number of nodes in the left column, then no one-to-one matching is found and the candidate spectrum range $[f_s, f_s+w]$ cannot be used, another range should be tried.

4. Performance Evaluation and Simulation Results

We evaluate the performance of spectral defragmentation and the bipartite matching algorithm in terms of blocking probability in a dynamic traffic scenario. The 14-node NSFNET (Fig.2(c)) was used as the simulation topology, with the assumption that each fiber has 5 THz total spectrum and the bandwidth of each subcarrier is 12.5 GHz. The required bandwidth for connections is uniformly distributed from the smallest granularity (12.5 GHz) to 500 GHz. New connection requests arrive according to a Poisson process at a rate of $\lambda$, and their holding time conforms to a negative exponential distribution with parameter $\mu$. The average holding time $h$, which equals $1/\mu$, is 730 time units in the simulation. The load of the network is measured by “offer load”, which is the product of $\lambda$ and $h$. From the results, the blocking probability is greatly reduced by spectral defragmentation. In particular, the pre-switch architecture results in a 50% reduction in the blocking probability with only one FWM element for one link. Increasing to four FWM elements, the blocking probability remains low (0.07) even with high network load ($\lambda=0.9$). Due to the minimum-cost search, the average number of affected connections is less than 1.8 for each new connection request.

Fig.3 (a) Blocking probability using pre-switch architecture. (b) Blocking probability using post-switch architecture. (c) Number of four-wave mixing blocks needed on each link in the pre-switch architecture.

5. Conclusion

This paper proposes a highly efficient spectral defragmentation algorithm and two scalable wavelength cross connect architectures to enable on-demand lightpath provisioning in flexible bandwidth networks. Our simulation results demonstrate significant reductions in blocking probabilities enabled by spectral defragmentation.

6. References