Knapsack-based Drop-and-Continue Traffic Grooming for Power and Resource Efficiency

M. Masud Hasan[1], **Farid Farahmand**[2], **Jason P. Jue**[3] **and Zuqing Zhu**[4]

[1]*Dept. of Mathematics and Computer Science, Elizabeth City State University, NC*
Email: mmhasan@mail.ecsu.edu
[2]*Dept. of Engineering Science, Sonoma State University, CA*
Email: farid.farahmand@sonoma.edu
[3]*Dept. of Computer Science, The University of Texas at Dallas, TX*
Email: jjue@utdallas.edu
[4]*School of Info. Science and Tech., University of Science and Technology of China, Anhui*
Email: zqzhu@ieee.org

Abstract: We propose a “knapsack grooming” algorithm for drop-and-continue lightpath routing, which not only saves energy, but also optimizes network performance in terms of blocking probability, wavelength-link usage, and bandwidth utilization.

OCIS codes: (060.4251) Networks, assignment and routing algorithms; (060.4256) Networks, network optimization

1. Introduction

In addition to reduce carbon-footprint on the environment, energy-conserving protocols are necessary to reduce operational expenditures (OPEX) for sustainable growth of wavelength-division multiplexing (WDM) networks. To conserve energy, traffic grooming can be used where smaller traffic requests are aggregated to fill up wavelength capacity, thereby reducing the usage of active devices in the networks.

A traditional traffic grooming architecture often requires add-drop devices, which implements optical-electrical-optical conversion of the bypassing signals. Such a solution is known to be power-hungry. To overcome this issue, the paper in [1] considers a drop-and-continue (DAC) architecture where the local traffic from a lightpath of groomed requests is optically dropped via splitters. The advantage is that expensive and power-hungry electronic devices are not required at the intermediate nodes where traffic needs to be dropped, but not added. However, as reported in [2], energy-aware routing schemes may exhibit higher blocking probability if longer lightpaths are chosen to facilitate grooming and/or to maximize the re-use of already active components.

In this paper, we propose an intelligent traffic grooming protocol to achieve energy-efficiency as well as to improve network performance with regard to blocking of requests, bandwidth utilization, and wavelength-link usage. In particular, we propose knapsack grooming inspired by the well-known knapsack problem of combinatorial optimization [3]. In the original problem, a knapsack needs to be optimally filled up by items having their own values and weights. In our case, the knapsack is one wavelength capacity which needs be filled up optimally by traffic requests having their values (e.g., bandwidth demand and number of common destinations) and weights (e.g., hop-distance). Accounting such values and weights in grooming along with DAC routing leads to power and resource optimization as described in subsequent sections. We also justify the approach by comparing it with other strategies through simulation.

2. Problem Statement and Architecture

Given a network and a static set of requests, we have to maximize traffic grooming by considering requests from the same source node and from different source nodes, and allocate requests using DAC lightpath routing obeying wavelength continuity constraint. We assume that, in DAC optical cross-connects, an incoming optical signal can be split unequally using, e.g., tap-and-continue passive devices. Thus, a fraction of the optical power can be dropped and processed electronically, while the remaining power can travel all-optically to the next node with negligible degradation and, if required, with optical compensation.

3. Knapsack Grooming in Drop-and-Continue Lightpath

For the proposed knapsack grooming approach, we use two values to prioritize grooming of requests: total bandwidth demand and common destination ratio (CDR). For two requests r and s, total bandwidth demand is $BW(r)$ and $BW(s)$, where $BW()$ denotes bandwidth demand of a request. Requests r and s are compatible to groom if $(BW(r)+BW(s)) \leq$ knapsack capacity. For two compatible requests, let c be the number of common destinations (i.e., both requests want to reach these destinations) and let d be the number of destinations that are not in common. We define common destination ratio, $CDR(r,s) = c/(c+d)$. It can be seen that $0 \leq CDR(r,s) \leq 1$. Candidate request-pairs for knapsack grooming fall in two weight groups based on hop-distance: requests from the same source in a higher weight group (kept in priority queue Q_1) and requests from different sources in a lower weight group (kept in priority queue

```
Input: a network, a set of requests, a node $u$
Output: a priority queue $Q_1$ of request-pairs from $u$ and a priority queue $Q_2$ of request-pairs considering neighboring nodes; allocate requests if needed
Pseudo-code:
$Q_1 \leftarrow Q_2 \leftarrow \emptyset$
$K \leftarrow$ wavelength capacity
for each request $r$ from $u$
    for each request $s$ from $u$ or its neighbors; $r \neq s$
        if $BW(r) + BW(s) \leq K$
            if $r$ and $s$ are from $u$
                insert $r$-$s$ pair in $Q_1$ in order of total bandwidth, then in order of $CDR(r, s)$
            else if $CDR(r, s) = 1$
                insert $r$-$s$ pair in $Q_2$ in order of total bandwidth
    if $r$ cannot be groomed with any requests, allocate $r$
```

Fig. 1. Algorithm for priority queues and routing.

```
Input: a network, a set of requests
Output: groomed request-pairs
Pseudo-code:
while there is an unallocated request
    for each node $u$ with unallocated requests
        call the function to produce $Q_1$ & $Q_2$ for $u$
        while ($Q_1$ is not empty)
            select the first pair ($r$, $s$) from $Q_1$ and groom
            if any pair in $Q_1$ & $Q_2$ refers $s$, remove the pair
            if any pair in $Q_1$ & $Q_2$ refers $r$, update new total demand and $CDR(r, s)$
        while ($Q_2$ is not empty)
            select the first pair ($r$, $s$) from $Q_2$, find route for $s$
            if route for $s$ toward $r$ is not blocked
                groom ($r$, $s$) pair
                if any pair in $Q_1$ & $Q_2$ refers $s$, remove the pair
                if any pair in $Q_1$ & $Q_2$ refers $r$, update new total demand and $CDR(r, s)$
```

Fig. 2. Algorithm for selecting request-pairs to groom.

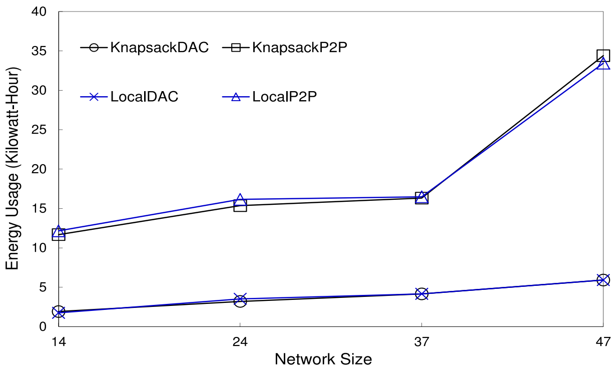

Fig. 3. Power consumption for different networks.

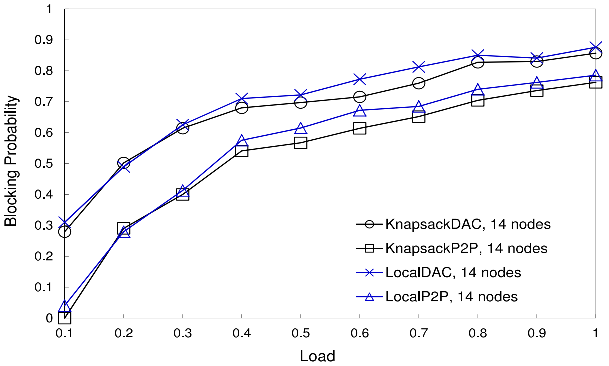

Fig. 4. Blocking probability for the 14-node network.

Q_2). Q_1 sorts request-pairs in order of total bandwidth, where ties are broken by CDR values, and then randomly. The reason to value total bandwidth more than CDR is to maximize bandwidth utilization, which will require fewer wavelength-links and reduce blocking probability. If no request-pair is available in Q_1, Q_2 will be considered for grooming. Since grooming request-pairs from different sources requires active transceivers (as opposed to grooming request-pairs from the same source where no transceivers are needed), we prioritize Q_1 over Q_2. This will reduce usage of active components, saving energy. To optimize further, Q_2 includes request-pairs only if CDR = 1 (i.e., having the same set of destinations). Q_2 sorts such request-pairs in order of total bandwidth, and then randomly. That is, during the grooming of requests between neighboring nodes, CDR is valued highly to ensure the same route toward destinations is used to fill up the knapsack as much as possible.

The knapsack grooming approach is outlined in Fig. 1 and Fig. 2. In Fig. 1, priority queues for a node u are generated by considering all other requests from u and its neighboring nodes. As discussed above, total bandwidth and CDR are used in ordering request-pairs. If a request r cannot be groomed with any other requests, r will be immediately routed using DAC and resources will be allocated. In Fig. 2, each node with unallocated requests repeatedly invokes the previous algorithm to form queues of candidate request-pairs. Then, request-pairs are groomed in order. After each grooming, queues and the set of requests reflect the new total bandwidth, destinations, and CDR values. Once a knapsack is filled up or a request cannot be groomed, all destinations in the request are routed using a DAC lightpath. In DAC routing, we choose the destination that has the minimum hop-distance from the current source node and allocate resource. We repeat this process until all destinations are reached, otherwise the request is blocked. The worst case computation complexity of knapsack grooming is polynomial, $O(N^2R^3)$, where N the number of nodes in a network and R is the maximum number of requests in a node.

4. Numerical Results

We study the performance of the proposed approach in terms of energy-efficiency (i.e., number of active devices such as transceivers), blocking probability, number of wavelength-link used, and bandwidth utilization. We use the energy consumption model of the Cisco Catalyst 6500 series as published in [4], where a grooming module (transmitter or receiver, including the E/O and O/E conversion) consumes 160 watts. An unused module (i.e., not processing traffic) is in idle mode and consumes negligible amount of energy. We experiment on four well-known mesh topologies: 14-node 21-link NSF network, 24-node 43-link NSF network, 37-node 57-link pan-European fiber-optic network, and 47-node

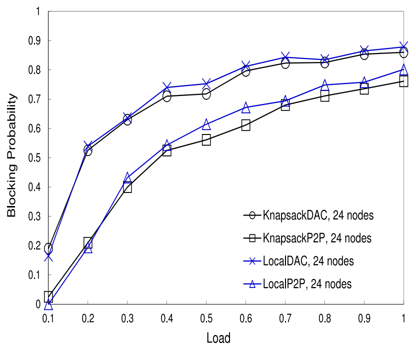

Fig. 5. Blocking probability for 24 nodes.

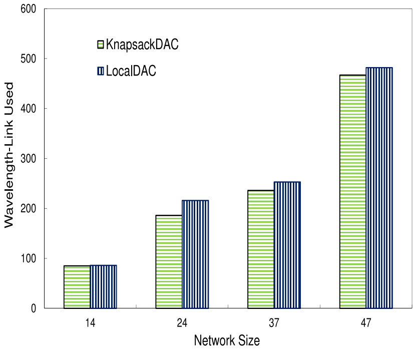

Fig. 6. Number of wavelength-links used.

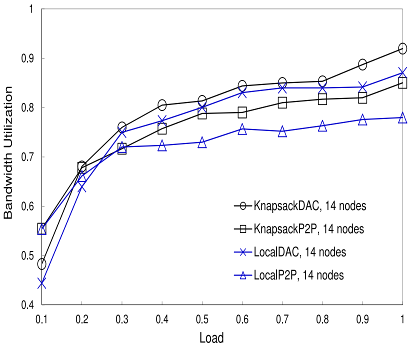

Fig. 7. Bandwidth utilization for 14 nodes.

98-link network from Paul Baran's historical paper [5]. Links (L) are bidirectional and the number of wavelengths (W) per link is 4. Wavelength capacity is OC-192 and traffic requests are of sub-wavelength granularity (i.e., OC-1, OC-3, OC-12, or OC-48). We assume that static traffic requests uniformly distributed across networks. We offer traffic load based on network capacity, which is calculated as $2 \times L \times W \times$ OC-192. As for benchmarks, we implement multiple grooming and point-to-point (P2P) routing strategies other than the proposed scheme. The proposed scheme is called *KnapsackDAC* which follows knapsack grooming and DAC lightpath routing as described earlier. *KnapsackP2P* is a strategy that follows the proposed knapsack grooming, but uses traditional P2P routing (with O/E/O conversion) instead of DAC routing. This benchmark will reveal the advantages of DAC. *LocalDAC* is a strategy that grooms requests only within a source node based on total traffic demand and uses DAC routing. This benchmark will reveal the advantages of knapsack grooming. Finally, *LocalP2P* is another strategy that also grooms within a source node and uses P2P routing. This benchmark will reveal the advantages of both knapsack grooming and DAC routing.

Figure 3 compares total energy usage in kilowatt-hour by all strategies at a non-blocking load (= 0.1). Regardless of network size, DAC routing clearly outperforms P2P routing in terms of total active devices in networks. Since DAC routing relies on passive splitters to serve intermediate destinations in a lightpath, an average of 75% energy is saved compared to P2P routing. Although *KnapsackDAC* and *LocalDAC* exhibit similar energy consumption patterns in Fig. 3, *KnapsackDAC* has lower blocking probability than *LocalDAC* as shown in Fig. 4 for 14 nodes and in Fig. 5 for 24 nodes under varying load from 0.1 to 1. For both networks with a particular routing strategy, knapsack grooming shows better blocking probability (i.e., *KnapsackDAC* is better than *LocalDAC*, and *KnapsackP2P* is better than *LocalP2P*). One of the reasons is that knapsack grooming intelligently grooms requests to utilize the same route for common destinations, whereas the other grooming strategy uses a route for more heterogenous destinations. Consequently, a lightpath in knapsack grooming has to reach fewer destinations on average. This attribute can be perceived from Fig. 6, which shows number of wavelength-links used in *KnapsackDAC* and *LocalDAC* at 0.1 load. For all network sizes, *KnapsackDAC* requires fewer wavelength-links to satisfy requests, hence experiences less blocking. Another reason for lower blocking is due to the potential for grooming of requests from neighboring nodes in the knapsack based scheme, as opposed to restricted grooming of requests only within the same node in the local grooming scheme. Finally, Fig. 7 depicts bandwidth utilization (ratio of actually used bandwidth to total allocated bandwidth) for requests in the 14-node network under varying load. As expected, for the reasons mentioned above, a lightpath using knapsack grooming carries more useful data and less waste of bandwidth. Among all schemes, *KnapsackDAC* is the most efficient in resource utilization.

5. Conclusion

In this work, we introduce an intelligent traffic grooming scheme to be used with drop-and-continue lightpath routing. The purpose is to address the problem of higher blocking of requests found in energy-efficient routing protocols. The proposed knapsack grooming prioritizes request-pairs based on total bandwidth demand and number of common destinations. It also considers request-pairs between neighboring nodes if local grooming is not possible. Simulation results exhibit power saving as well as reduced blocking probability, fewer wavelength-link usage, and higher bandwidth utilization compared to other approaches.

References

1. F. Farahmand, I. Cerutti, M. Hasan, and J. Jue, "Energy-efficiency of Drop-and-Continue Traffic Grooming," In *IEEE/OSA OFC*, (2011).
2. M. M. Hasan, F. Farahmand, A. N. Patel, and J. P. Jue, "Traffic Grooming in Green Optical Networks," In *Proc. of IEEE ICC*, (2010).
3. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness* (W. H. Freeman and Co., NY, 1979).
4. Cisco® Power Calculator, Available online: http://tools.cisco.com/cpc, Accessed: Sep 2011.
5. P. Baran, "On Distributed Communications Networks," IEEE Trans. on Communications **12,** 1–9 (1964).