This paper was presented as part of the High-Speed Networks 2011 (HSN 2011) Workshop at IEEE INFOCOM 2011

Optimal Node Hardware Module Planning for Layer-One Optical Transport Networks

Gangxiang Shen, Yunfeng Shen, and Harshad P. Sardesai

Abstract—**Most of the studies on traffic grooming focus on minimizing network link capacity and providing serving-relationship between client services and link capacity. Subsequent to this step, it is important to plan for adding/dropping client services over client service ports and setting up end-to-end lightpaths over network ports, which is however seldom investigated. We call such effort *node hardware module planning*. This is an industrially practical problem aiming to minimize node hardware cost since hardware modules are the most expensive components in a network. Based on a link-based traffic grooming result that provides information on end-to-end capacity units incident to nodes and aggregation relationship between client services and capacity units, we develop an Integer Linear Programming (ILP) model to optimally plan hardware modules. To overcome the computation difficulty of the ILP model under large-sized planning scenarios, we also develop a fast sub-optimal heuristic for hardware module planning. Simulation studies indicate that the heuristic is efficient to achieve a design close to an optimal solution obtained by the ILP model. Also, the evaluation of the impact of switch backplane size shows that given a certain set of network modules, an optimal switch backplane size exists, which achieves the lowest hardware cost.**

Index Terms—**Hardware module planning, switch backplane, hardware module, client service tree**

I. Introduction

We have witnessed extensive research effort on subwavelength traffic grooming in different types of networks such as SDH/SONET networks [1] [2], IP over WDM networks [3], and Optical Transport Networks (OTN) [4]. One of major motivations of traffic grooming is to efficiently fill large wavelength capacity pipes with fine subwavelength client services. The subwavelength services can be traditional SONET/SDH services, pure IP traffic, or Optical Transport Network (OTN) client services. The wavelength capacity pipes can be 10 Gb/s, 40 Gb/s, or up to 100 Gb/s based on today's optical transmission technology. Most of the existing traffic grooming studies focus on minimizing required end-to-end lightpath capacity given that all the subwavelength client services are provisioned [1], or maximizing served client services subject to a limited amount of predefined wavelength capacity [2]. Minimizing required end-to-end lightpath capacity helps reduce the total number of optical transponders, thereby reducing network cost. Similarly, maximizing served client services [2] best caters to users' traffic demands, thereby increasing network operational revenue.

In parallel with link-based capacity planning, there is another important problem that is industrially practical, but not well explored: *how to plan hardware modules on each node subsequent to a link-based traffic grooming design*. We need to consider how many switch shelves and hardware modules should be deployed at each node, such that all the add-drop client services and end-to-end lightpaths at the node can be fully supported. The problem can essentially be modeled as a bin-packing optimization process [5] with several system-related constraints taken into account. These constraints, for example, include a limited size of switch backplane (which allows only a limited number of hardware modules commonly connected to), limited numbers of client service ports and network ports on each hardware module, and so on.

Because dark fibers are often pre-laid underground and optical amplifiers are shared by a large number of wavelength channels, hardware modules and switch backplanes at each node are actually the most expensive components in an optical transport network. It is of importance to optimally plan and organize them at each node such that the most significant amount of cost can be saved. We employ the Integer Linear Programming (ILP) technique to optimally model the problem. By extending the heuristic proposed in our previous work [6], we also develop a general heuristic that can consider different hardware module types. The major contributions in this paper compared to [6] include a new ILP optimization model and a more generalized heuristic for hardware module planning. This study focuses on the layer-one Optical Transport Networks (OTN) owing to its wide industrial support and capability of carrying various types of client services. However, the approaches and models are extendable to plan for other types of networks.

The rest of the paper is organized as follows. In Section II, we have a literature review on link-capacity-based traffic grooming. We also introduce the layer-one Optical Transport Networks and discuss how hardware modules are organized and laid out at a node. In Section III, we present an ILP model for optimal hardware module planning. Also, an efficient heuristic is developed for large-sized hardware module planning scenarios. Simulations are performed and results are analyzed in Section VI. Section V concludes the paper.

Gangxiang Shen is with College of Electronics and Information, Soochow University, P. R. China. (e-mail: shengx@suda.edu.cn). Yunfeng Shen and Harshad P. Sardesai are with Ciena Corp., Linthicum, Maryland, USA.

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

II. BACKGROUND

A. Literature Review

Link-based traffic grooming has been investigated for many years. Ring-based traffic grooming was first studied to groom SDH/SONET tributaries onto lightpath channels, wherein minimizing the required number of optical transponders was considered a key objective [1]. Traffic grooming was later extended to mesh networks. The authors in [2] maximized SDH/SONET client services provisioning in a mesh network given a predefined network link capacity and transceivers at each node. In [7], analytical models were developed for dynamic traffic grooming in a mesh network. Studies were also performed to consider sparse traffic grooming in a network [8][9][10]. Recently, traffic grooming considering the physical-layer impairments [10][11][12] was investigated to take advantage of traffic grooming capability of a signal regeneration node and traffic grooming with energy minimization also received a wide interest [13][14].

It is important to optimally plan hardware modules subsequent to link-based traffic grooming since node hardware is the most expensive in a network. Recently, we made a preliminary study on this problem [6] and proposed an efficient heuristic to plan hardware modules under the assumption of a single type of switch backplane, client service module, and network module. This paper extends this study to consider a more general design scenario by allowing for multiple types of switch backplanes, client service and network modules. We consider hardware modules whose client service ports support all type client services, as well as modules whose client service ports are dedicated to a certain protocol.

B. Optical Transport Networks

ITU Standard G. 709 defines Optical Transport Network (OTN) [4], which is also called "digital wrapper." OTN is a type of *layer-one* transport network, directly sitting on the Optical Channel layer. OTN currently supports four line rates, namely OTU1, OTU2, OTU3 and OTU4, which correspond to data rates 2.5 Gb/s, 10 Gb/s, 40 Gb/s and 100 Gb/s, respectively.

Optical transport networks provide a converged layer-one network transport platform to support various types of client services. They include traditional time-slot-based SDH/SONET services (e.g., STM-4 and OC-3), today's popular Carrier Ethernet services (e.g., Gigabit Ethernet), traditional Storage Area Network (SAN) services (e.g., Fiber Channel), and even video services. Virtual concatenation [15] plays a key role in OTN to enable it to widely support these different types of services. Specifically, each client service is mapped to different numbers of basic time slots (i.e., virtual containers). Each time slot is a basic granularity that is transported and switched in OTN. At a receiver side, these time slots are gathered and reassembled back to corresponding client services.

C. Node Hardware Layout

A switch node consists of multiple groups of hardware modules, called "*module groups*." Modules in each group are commonly connected to a non-blocking switch backplane. These modules can directly communicate with each other through the backplane. However, modules in different groups (on different switch backplanes) generally cannot directly communicate with each other except that there is an external link (such as a short-reach fiber link) connecting the groups. Fig. 1 (a) shows an *example* of a switch node, in which a total of 12 hardware modules are divided into three groups. Each group contains four hardware modules; they are connected to a common switch backplane. The node contains three independent switch backplanes. Modules 2 and 3 (counted from the left to the right) can directly communicate with each other as they are on the same module Group 1. In contrast, because modules 4 and 5 are on different switch backplanes (i.e., Group 1 and Group 2), they cannot communicate with each other though they are physically neighboring. Note here we just show an example of how hardware modules are organized in groups and connected to common switch backplanes. Different switch node products can have different switch backplane sizes and total numbers of contained hardware modules.

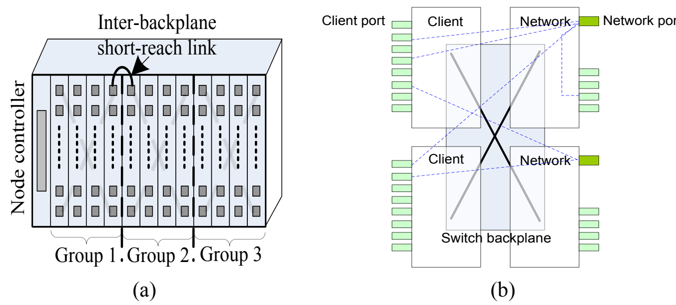

Fig. 1. (a) A switch node with multiple groups of hardware modules; (b) Modules connecting to a common switch backplane.

Hardware modules may be classified into two categories, namely *client service modules* and *network modules*. A client service module has a fixed number of ports for adding-dropping client services. The ports can be either general to support client services of any type of protocol, or protocol-dedicated to support client services of a certain protocol type[1]. We call the former "*general modules*," and the latter "*protocol-dedicated modules*." For example, a protocol-dedicated client service module can be a module that only supports SDH/SONET client services on all its ports. General modules are flexible in supporting various types of client services. They are however more expensive than protocol-dedicated modules. To minimize a node hardware cost, we should maximally deploy protocol-dedicated modules by gathering as many client services of the same protocols onto common hardware modules as possible.

Different from client service modules, network modules are equipped with network ports. Each network port is connected to an end-to-end lightpath capacity unit. For better module port density, a network module may also be equipped with a small

[1] For example, Ciena's CN4200 hardware modules can be softly configured to generally support any type of services or support services of a certain protocol type up to 2.5 Gb/s on all its client service ports [16].

number of additional client service ports. The number of client service ports is usually smaller than that on a dedicated client service module. Likewise, depending on the client service supporting capability of the client service ports, network modules can be general or protocol-dedicated.

Fig. 1 (b) shows an *example* of hardware modules connecting to a common switch backplane, on which there are two network modules and two client service modules. Each of the network modules carries one network port and four client service ports, and each of the client service modules carries six client service ports. The dotted lines represent how client services are switched to network ports over the switch backplane. In the Optical Transport Networks (OTN) [4], client-service switching between modules is performed on a time-slot basis, which is based on the virtual concatenation technique and whose minimal granularity for example can be at the level of STM-1/OC-3 or ODU-0.

III. Hardware Module Planning

A. Problem Statement

The problem of hardware module planning can be stated as follows. Given a link-based subwavelength traffic grooming result at a node, we plan hardware modules and organize them in groups with each group corresponding to a switch backplane. The objective of this planning is to minimize the total hardware cost. The constraints that the design is subject to include (i) all the add-drop client services are provisioned over client service ports, (ii) all the network capacity units incident to the node are supported by network ports, (iii) a limited switch backplane size, limited numbers of client service ports and network ports on hardware modules.

In this planning, we only consider the ports of the traffic demands that are locally added/dropped and the related network ports. For all those bypass client services, which traverse the node via the enclosed switch backplanes from ingress network ports to egress network ports, we do not need to assign client service ports.

We consider two categories of hardware modules, namely *protocol-dedicated* and *general* modules. In general, protocol-dedicated modules are cheaper than general modules as the former dedicatedly support a single type of protocol, e.g., SDH/SONET, while the latter should be powerful enough to support any type of client services. To achieve a cost-minimized planning, we should use protocol-dedicated modules to support client services whenever possible. For example, when all the occupied ports on a module carry client services of the same protocol type, we always use a corresponding protocol-dedicated module. A more expensive general module is only used when client services of different protocols are served on the module.

As modules connected to different backplanes cannot directly communicate with each other (except that a short-reach fiber link is set up between two network ports of the backplanes), in order to avoid wasting network ports, it is strategically efficient to accommodate all the client services under a capacity unit onto a common switch backplane. This is because if a partial set of client services are on a switch backplane different from that of the corresponding network port, to aggregate these *isolated* client services onto the network port, we have to set up an inter-backplane short-reach link. Fig. 1(a) just shows such an example, where an inter-backplane link crossing module Groups 1 and 2 connects two network ports respectively from modules 4 and 5, if some client services are in module Group 1 while their corresponding network port is in module Group 2. This causes a waste of two network ports compared to a design that accommodates all the client service ports and their network port on the same module group. Thus, to avoid such a waste, it is important to accommodate all the client services and their corresponding network port on the same switch backplane when carrying out hardware module planning.

B. ILP Optimization Model

The hardware module planning problem can be formulated by an Integer Linear Programming (ILP) model. In this section, we present this model by taking into account all the constraints and concerns discussed in Section-A. We first introduce related sets, parameters, and variables. We then present optimization formulae and explain the objective and constraints. The model considers a single node. For a complete network with multiple nodes, we repeat the same optimization process for each of the nodes.

The **sets** of the model are as follows:

R is the set of data rates of network capacity units (network ports), which for instance can be 10 Gb/s, 40 Gb/s, and 100 Gb/s.

P is the set of protocol types of client services, which can be protocols such as SONET/SDH, Carrier Ethernet, etc.

T_r is the set (or number) of network capacity units at data rate *r*, $r \in \mathsf{R}$. Each unit provides capacity and corresponds to a network port on a network module.

X is the set of switch backplane types. Different types of switch backplanes maximally hold different numbers of hardware modules.

B_x is the set (or number) of switch backplanes of type *x*, $x \in \mathsf{X}$.

M_p is the set of module types, whose client service ports are dedicated to the services of protocol type *p*, $p \in \mathsf{P}$.

M_G is the set of general module types, whose client service ports can support client services of any type of protocol. Here subscript *G* is not an index, but denotes a general module type.

The **parameters** of the model are as follows:

$\beta_{t,r}^{p}$ is the number of client services of protocol *p* aggregated on the t^{th} network trunk whose data rate is *r*, $t \in \mathsf{T}_r$.

α_x is the number of hardware modules that can be maximally accommodated by switch backplane type *x*, $x \in \mathsf{X}$.

ξ_m^p is the number of client service ports on the m^{th} module type that is dedicated to protocol p, $p \in \mathbf{P}$ and $m \in \mathbf{M}_p$.

$\mathrm{N}_{m,p}^r$ is the number of network ports that support data rate r on the m^{th} module type that is dedicated to protocol p, $r \in \mathbf{R}$ and $m \in \mathbf{M}_p$.

ξ_m^G is the number of client service ports on the m^{th} type of general module, $m \in \mathbf{M}_G$.

$\mathrm{N}_{m,G}^r$ is the number of network ports that support data rate r on the m^{th} type of general module, $r \in \mathbf{R}$ and $m \in \mathbf{M}_G$.

χ_m^p is the cost of the m^{th} module type that is dedicated to protocol p.

χ_m^G is the cost of the m^{th} type of general module.

δ_p is the cost of the p^{th} type of switch backplane.

Δ represents a large value.

The **variables** of the model are as follows:

$\theta_{b,x}^{t,r}$ is a binary variable, which takes the value of one if the t^{th} network capacity unit at data rate r (and all its contained client services) is accommodated on the b^{th} backplane of type x; zero, otherwise.

$y_{b,x}$ is a binary variable indicating whether the b^{th} switch backplane of type x should be activated. If so, it takes the value of one; zero, otherwise.

$z_{b,x}^{m,p}$ is the number of the m^{th} module type that is dedicated to protocol p and accommodated on the b^{th} switch backplane of type x.

$x_{b,x}^{m,G}$ is the number of the m^{th} type of general module, accommodated on the b^{th} switch backplane of type x.

$\Phi_{b,x}^p$ is an intermediate variable, indicating the maximal number of client services of protocol p, not yet served by protocol-dedicated modules on the b^{th} switch backplane of type x.

The objective and constraints of the model are given as follows:

Objective: Minimize

$$\sum_{b\in\mathbf{B}_x, x\in\mathbf{X}} y_{b,x}\cdot\delta_x + \sum_{\substack{m\in\mathbf{M}_p, p\in\mathbf{P};\\ b\in\mathbf{B}_x, x\in\mathbf{X}}} z_{b,x}^{m,p}\cdot\chi_m^p + \sum_{\substack{m\in\mathbf{M}_G;\\ b\in\mathbf{B}_x, x\in\mathbf{X}}} z_{b,x}^{m,G}\cdot\chi_m^G$$

Subject to:

$$\sum_{b\in\mathbf{B}_x, x\in\mathbf{X}} \theta_{b,x}^{t,r} \le 1 \quad \forall t\in\mathbf{T}_r, r\in\mathbf{R} \tag{1}$$

$$\sum_{t\in\mathbf{T}_r} \theta_{b,x}^{t,r} \le \sum_{m\in\mathbf{M}_p, p\in\mathbf{P}} z_{b,x}^{m,p}\cdot\mathrm{N}_{m,p}^r + \sum_{m\in\mathbf{M}_G} z_{b,x}^{m,G}\cdot\mathrm{N}_{m,G}^r \quad \forall b\in\mathbf{B}_x, x\in\mathbf{X}; r\in\mathbf{R} \tag{2}$$

$$\sum_{t\in\mathbf{T}_r, r\in\mathbf{R}} \theta_{b,x}^{t,r}\cdot\beta_{t,r}^p - \sum_{m\in\mathbf{M}_p} z_{b,x}^{m,p}\cdot\xi_m^p \le \Phi_{b,x}^p \quad \forall b\in\mathbf{B}_x, x\in\mathbf{X}; p\in\mathbf{P} \tag{3}$$

$$\sum_{p\in\mathbf{P}} \Phi_{b,x}^p \le \sum_{m\in\mathbf{M}_G} z_{b,x}^{m,G}\cdot\xi_m^G \quad \forall b\in\mathbf{B}_x, x\in\mathbf{X} \tag{4}$$

$$\sum_{m\in\mathbf{M}_p, p\in\mathbf{P}} z_{b,x}^{m,p} + \sum_{m\in\mathbf{M}_G} z_{b,x}^{m,G} \le \alpha_x \quad \forall b\in\mathbf{B}_x, x\in\mathbf{X} \tag{5}$$

$$\Delta\cdot y_{b,x} \ge \sum_{m\in\mathbf{M}_p, p\in\mathbf{P}} z_{b,x}^{m,p} + \sum_{m\in\mathbf{M}_G} z_{b,x}^{m,G} \quad \forall b\in\mathbf{B}_x, x\in\mathbf{X} \tag{6}$$

The objective of the model is to minimize the total sum cost of switch backplanes and hardware modules, in which the first sum is the cost of switch backplanes, the second sum is the cost of protocol-dedicated hardware modules, and the last sum calculates the cost of general hardware modules. Constrain (1) ensures that each network capacity unit and the aggregated client services under the unit are accommodated on a common switch backplane. Constraint (2) ensures that within each of the switch backplane, there are sufficient network ports at data rate r to accommodate all the network capacity units incident to the node. Constraint (3) counts the maximal number of un-served client services of protocol p on the backplane after we employ protocol-dedicated modules to serve them. These un-served client services are all served by general modules as guaranteed by Constraint (4). Constraint (5) ensures that on each switch backplane, the total number of modules (including protocol-dedicated and general modules) never exceeds the maximal number of hardware modules supported by each type of switch backplane. Constraint (6) indicates that if a module group is not empty, we need to deploy a switch backplane to accommodate all the modules in the group. Otherwise, we do not need to deploy any switch backplane.

C. Heuristic

The hardware module planning problem is essentially a variation of the well-known bin-packing problem, which is NP-hard [5]. Thus, when the problem size grows large, it is necessary to develop an efficient heuristic. In this section, we extend the heuristic in our previous work [6] to support a more generic hardware module planning scenario, which allows hardware modules to be either protocol-dedicated or general. The flowchart of the extended heuristic is presented in Fig. 2.

The optimization effort focuses on the following two sequential steps. Step 1 minimizes the total number of module groups and hardware modules, as the numbers of these components most significantly affect the total cost of a node. Specifically, we first sort all the capacity units (e.g., 10-Gb/s capacity trunks) locally incident to the node based on the numbers of their contained add-drop client services at the node from the largest to the smallest and store them in a capacity unit list *CL*. Such a sorting process is enlightened by the bin-packing process [5], in which it is always more efficient to pack larger items first. Next, we initialize an empty list *MG* of module group, of which each corresponds to a switch backplane that accommodates all the included hardware modules. The subsequent *for*-loop is a bin-packing-like step to put items (i.e., a network capacity unit and its carrying client services) into bins (i.e., module groups). For each of the capacity units, we try all the module groups in list *MG* to see if there is any group that can accommodate the current capacity unit and its associated client services. If so, we add the capacity unit and all the associated client services to the module group and perform mapping between add-drop client services and client service ports on hardware modules. Otherwise, we create a new module group with a *maximal* backplane size and use the new module group to accommodate the current capacity unit and all its associated client services. The new module group is meanwhile added to the module group list *MG*. The *for*-loop is terminated when all the capacity units in list *CL* are accommodated. Note in the current step (i.e., Step 1), we assume that all the placed hardware modules are general, of which each client service port can support a service of any protocol type. Such an assumption can minimize the total number of required hardware modules and switch backplanes.

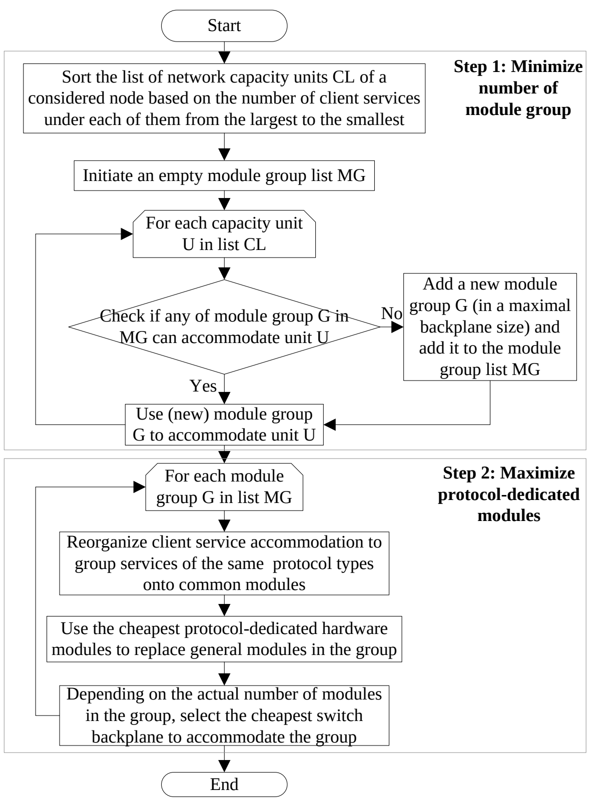

Fig. 2. Flowchart of hardware module planning heuristic.

After determining the total number of modules and grouping them, Step 2 performs per-backplane service re-organization, which groups client services that belong to the same protocol onto common hardware modules. After the re-organizing effort, we examine the client services on each of the modules. If all the client services on a module belong to the same protocol, we replace a cheaper protocol-dedicated module for the initial placed general module, such that the cost can be tuned down. Also, depending on the total number of hardware modules in each of the module groups, we deploy the cheapest switch backplanes that can provide sufficient module slots for the module groups.

IV. SIMULATIONS AND RESULT ANALYSES

To evaluate the performance of the proposed hardware module planning approaches, we ran simulations based on the following conditions and assumptions:

We employ the 14-node 21-link NSFNET network in [6], a benchmark topology that is widely used for network performance evaluation, as our test network. We apply the single-hop grooming approach [2] to groom all the client services between each pair of nodes onto network capacity units. The grooming result is input to the hardware module planning process. Without losing generality, we assume that the data rate of network capacity unit is 10 Gb/s that corresponds to an ODU-2. A uniform traffic demand distribution is assumed for all the node pairs in the network. The number of client services between each pair of nodes is randomly generated following a specific traffic demand density parameter. The protocol type of each service is randomly selected from a set of protocols. Again without losing generality, we consider two widely used protocols, namely SDH/SONET and Carrier Ethernet. Under the SDH/SONET protocol, three types of services including STM-1/OC-3, STM-4/OC-12, and STM-16/OC-48, are simulated. For Carrier Ethernet, Gigabit Ethernet services are assumed and each of them is mapped to 7 STM-1 slots.

We assume that there are three types of client service modules and network modules, respectively. They include general (client service and network) modules, SDH/SONET-dedicated modules, and GE-dedicated modules. Each of the client service modules is assumed to have 12 client service ports and each of the network modules is assumed to have two network ports and four client service ports. Each of the client ports on an SDH/SONET-dedicated module can support all types of SDH/SONET client services, a GE-dedicated module only supports GE client services, and general modules can support any type of client services. The data rate of each network port is 10 Gb/s, which corresponds to an ODU-2. Also, we assume that there are two types of switch backplanes with one accommodating six hardware modules and the other accommodating four modules. The basic time slot switched on the backplanes is STM-1/OC-3. The costs of the hardware modules and switch backplanes are normalized as in Table I.

TABLE I: COSTS OF HARDWARE MODULES AND SWITCH BACKPLANES

Hardware	Cost
SDH/SONET client service module	1.0
GE client service module	0.9
General client service module	1.2
SDH/SONET network module	2.0
GE network module	1.95
General network module	2.1
6-module switch backplane	0.55
4-module switch backplane	0.4

The ILP model was solved by commercial software AMPL/Gurobi [17], running on a desktop with 2.93-GHz Intel (R) Core (TM)2 Duo CPU and 2.00 Gb memory and the

MIPGAP of each optimal solution is within 2%. The heuristic was implemented in Java and executed on the same PC.

Node Hardware Cost: We assume that each node pair in the network has a random number of client services, of which each can be any one of the service types introduced in Section IV-A. Based on the traffic demand, we run single-hop grooming algorithm to generate grooming results, which are input to our optimization model and heuristic for hardware module planning. Fig. 3 shows the total node hardware cost (*network cost*, for simplicity) of the NSFNET network under different traffic demand densities as shown on the *x*-axis, on which each number *s* means that between each pair of nodes there is maximally up to *s* client services. The *y*-axis shows the total network costs. Four design cases are considered: (i) ILP model with general module, (ii) heuristic with general module, (iii) ILP model with all module types, and (iv) heuristic with all module types. The four cases correspond to "ILP (General)," "Heuristic (General)," "ILP (Protocol-dedicated)," and "Heuristic (Protocol-dedicated)" respectively in the legend.

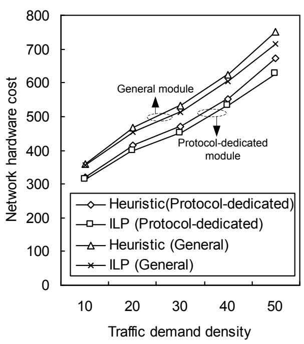

Fig. 3. Network hardware cost versus traffic demand density.

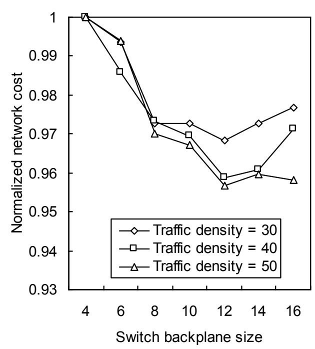

Fig. 4. Impact of switch backplane size on network hardware cost.

As expected, with the increase of traffic demand density, the total network costs of the four cases all increase. We compare the solutions of the ILP model and the heuristic. It is found that the proposed heuristic is quite efficient to find solutions close to those of the ILP model. The *maximal* difference between cases (i) and (ii) is about 5% and between cases (iii) and (iv) is about 6.5%. Moreover, when the traffic density between node pairs is smaller, the difference becomes smaller too. In addition, comparing the cases of "general modules" and "protocol-dedicated modules," we find that by replacing protocol-dedicated modules for general modules, we can significantly reduce up to 12% total hardware cost.

Effect of Switch Backplane Size: As in [6], we also evaluate the effect of switch backplane size on hardware cost when protocol-dedicated modules are employed. Fig. 4 shows the simulation results obtained by the heuristic under different traffic densities, in which the *x*-axis shows an increasing size of switch backplane and the *y*-axis shows a normalized network cost, which is normalized by the cost of a planning scenario whose switch backplane size is the smallest, four. According to Table I, we set the cost of 4-module switch backplane to be 0.4, and for other sizes of switch backplanes, an incremental cost 0.15 is set when the size of switch backplane increases by 2. With the increase of switch backplane size, the normalized network costs become smaller as a large switch backplane size provides more flexibility in accommodating client services on switch backplanes. In addition, it is interesting to find that a backplane size threshold exists, namely, when the backplane size reaches 12, a further increase of backplane size cannot bring a cost reduction. On the contrary, the cost can even become larger as a larger backplane is more expensive. This result is in line with the one under a general module configuration in [6].

V. CONCLUSION

In this paper, we investigated hardware module planning for layer-one Optical Transport Networks (OTN) given a link-based traffic grooming result. To minimize hardware cost, we developed an ILP optimization model and an efficient heuristic. Simulation studies showed that the heuristic is efficient to achieve performance close to that of the ILP model. Using protocol-dedicated modules can significantly reduce network cost over a single type of general module. Also, given a specific set of hardware module types, an optimal switch backplane size exists, which achieves the lowest hardware cost.

REFERENCES

[1] O. Gerstel, P. Lin, and G. Sasaki, "Combined WDM and SONET network design," in *Proc. INFOCOM'98*, vol. 2, San Francisco, CA, Mar. 1998.

[2] K. Zhu and B. Mukherjee, "Traffic grooming in …WDM mesh network," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 1, Jan. 2002, pp. 122–133.

[3] X. Niu, W. Zhong, G. Shen, and T. H. Cheng, "Connection establishment of label switched paths in IP/MPLS over optical networks," *Photonic Network Communications*, vol. 6, no. 1, pp. 33-41, January 2003.

[4] ITU Recommendation G.709/Y.1331, Interfaces for the Optical Transport Network (OTN), March 2003 (Amendment1 December 2003).

[5] Bin packing problem [Online]. Available: http://en.wikipedia.org/wiki/Bin_packing_problem.

[6] G. Shen, Y. Shen and H. P. Sardesai, "Optimal hardware module planning …," *OFC/NFOEC 2010*, San Diego, US, Mar. 2010.

[7] C. Xin, C. Qiao, and S. Dixit, "Traffic grooming in the mesh WDM optical network-performance analysis," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 9, 2004, pp. 1658 – 1669.

[8] W. Yao, M. Li, and B. Ramamurthy, "Performance analysis of sparse traffic grooming …," in *Proc. ICC'05*, 2005, pp. 1766–1770.

[9] B. Chen, G. N. Rouskas, and R. Dutta, "On hierarchical traffic grooming in WDM networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, Oct. 2008, pp. 1226–1238.

[10] G. Shen and R. S. Tucker, "Sparse traffic grooming …," *J. Lightwave Technol.*, vol. 27, no. 20, Oct. 2009, pp. 4471-4479.

[11] M. Batayneh, *et. al.*, "Optical network design for a multiline-rate …," *J. Lightwave Technol.*, vol. 26, no. 1, Jan. 2008. pp. 121-130.

[12] G. Shen, Y. Shen and H. P. Sardesai, "Cross-layer traffic grooming for … signal regeneration," *OFC/NFOEC 2010*, San Diego, US, Mar, 2010.

[13] G. Shen and R. S. Tucker, "Energy-Minimized Design for IP over WDM Networks," *J. of Optical Comm. and Net.*, vol. 1, no. 1, June 2009, pp. 176-186.

[14] M. Hasan, F. Farahmand, A. Patel, and J. Jue, "Traffic grooming in green optical networks," in *Proc. ICC'10*, South Africa, May 2010.

[15] Virtual concatenation concept [Online]. Available: http://en.wikipedia.org/wiki/Virtual_concatenation.

[16] Ciena's CN4200 product introduction: http://www.ciena.com/products/products_cn4200_overview.htm.

[17] AMPL+Gurobi, Linear programming optimization software package [Online]. Available: http://www.gurobi.com/doc/30/ampl/.