Optimal Hardware Module Planning and Switch Backplane Configuration for Layer-One Traffic Grooming

Gangxiang (Steven) Shen, Yunfeng Shen, and Harshad P. Sardesai
Ciena Corporation, 920 Elkridge Landing Road, Linthicum, MD 21090
Email: gshen@ciena.com

Abstract: We present an automatic hardware module planning approach for optical transport networks. An optimal switch backplane size is identified that can achieve the most cost-effective hardware module planning.

OCIS codes: (060.4250) Networks; (060.4256) Networks, network optimization

1. Introduction

Most research in the literature on layer-one or subwave traffic grooming has focused on (link) capacity planning [1][2][3], and there is sparse material dedicated to planning hardware modules (cards) and configuring switch backplanes subsequent to link capacity planning. Optimal hardware module planning is important for network planning and deployment, which can minimize the total number of hardware modules and chasses resulting in space and power consumption saving, and cost reduction.

In this paper, we focus on layer-one hardware module planning for optical transport networks. Subject to several system constraints, including a limited size of switch backplane and limited number of network and client service ports on network and client service modules, we develop a general automatic and optimal hardware module planning approach called *one-neck-to-grab* (ONTG) algorithm. Through minimizing hardware module groups that commonly connect to switch backplanes, the algorithm is efficient to minimize the total numbers of hardware modules and chasses. Simulation results indicate that given a certain combination of traffic demand granularity and hardware module sizes, a limited-size switch backplane is sufficient to achieve a good performance close to that of a larger switch backplane. An optimal switch backplane size is identified that can achieve the most cost-effective design.

2. Module layout at a switch node

A typical layer-one switch consists of a switch backplane, several *network modules* and *client service modules*. The switch backplane is generally non-blocking, which can switch traffic demand from one port to any other port. Network modules provide high-speed network (line) ports connected to network trunk capacity, e.g., a 10G network port connects a 10G network trunk capacity unit. The network ports are ingress or egress points for client services to transit (bypass) a node. The client service modules provide ports for low-speed client service add/drop. They aggregate client services onto high-speed network trunk capacity over network ports. Fig. 1 shows an example of hardware modules that are organized and connect to a common non-blocking switch backplane. Specifically, there are two network modules, and two client service modules.

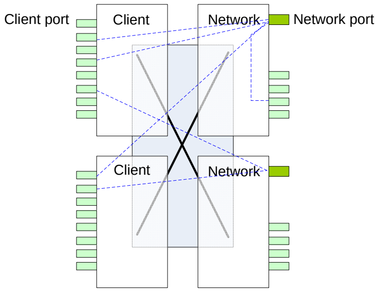

Figure 1: Modules connecting to a switch backplane.

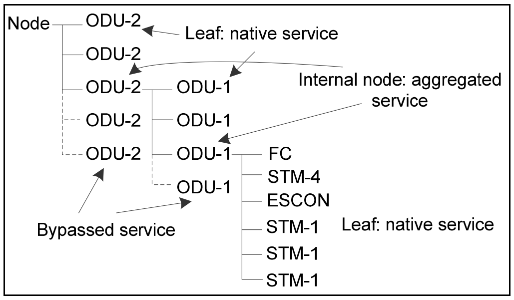

Figure 2: Tree representation of traffic demands at a node.

Each network trunk capacity unit accommodates a set of client services. As an example, an ODU-2 network port provides a 10-Gb/s line rate and can accommodate, for example, 16 STM-4s or eight GEs. A traffic grooming process is to find the link-capacity serving/served relationship between network capacity and client services. A hardware module planning process is to subsequently map this relationship to hardware connectivity between client service ports and network ports. Within a switch node, the connectivity between a client service port and a network port is realized through proper configuration of switch backplane. In Fig. 1, the blue dotted lines show the connectivity and relationship between client service ports and network ports.

In addition, the total number of hardware modules that are allowed to commonly connect to a switch backplane is limited. We term all the modules that connect to a common switch backplane "module group." For example, in Fig. 1, there are maximally four modules connecting to a common switch backplane. On the same switch backplane, all the client services or capacity slots can be switched between any ports. However, it is not allowed to switch a service between ports on different switch backplanes.

3. Tree representation of traffic demand at a node

To assist hardware module planning, we represent the traffic demand at each node (either add/drop or bypass) by using a tree structure as shown in Fig. 2. For example, we assume that there are different types of client services and these client services are divided into three hierarchies, including ODU-2, ODU-1, and sub-wavelength layers (lower than ODU-1). In each layer, all the leaf nodes with solid lines correspond to *native services* that are connected to customer client ports such as router ports. Other leaf nodes with dotted lines correspond to bypassing services. In Fig. 2, there are two bypassing ODU-2s and one bypassing ODU-1. Finally, all the internal nodes in the tree carry aggregated or groomed services. In the ODU-2 layer, there is one aggregated ODU-2 in Fig. 2, which grooms four ODU-1s. In the ODU-1 layer, there is an aggregated ODU-1, which grooms FC, STM-4, ESCON and STM-1 services that occupy different numbers of time slots in ODU-1. Here ODU-2 and ODU-1 respectively correspond to 10 Gb/s and 2.5 Gb/s capacity under the optical transport network (OTN) standards [4].

The tree architecture provides good mapping to real hardware. Each leaf node for native service corresponds to a client service port on a hardware module. All the leaf nodes for bypass services correspond to bypass connections over the switch backplane of the node, not requiring any client service ports. An internal ODU-2 node corresponds to a network port on a network card (if the wavelength data rate is 10 Gb/s), which aggregates all the sub-wavelength and ODU-1 client services over a switch backplane.

4. Hardware module planning algorithm

Based on the tree representation as shown in Fig. 2, we plan the required number of modules such that all the *leaf* client services are supported. Without losing generality, we assume that there are a single type of client service module and a single type of network module. Each client service module provides M client service ports, of which each can support any type of client service with a data rate up to ODU-1. Each network module is assumed to provide X network ports. In addition, for better port density, the network module may support additional Y client service ports. Generally $X+Y\leq M$. Finally, we assume that up to Φ network or client service modules can be connected to a common switch backplane.

When planning hardware modules, it is efficient to arrange all the modules, which carry client services under a common network trunk capacity unit, around a common switch backplane. For example, in Fig. 2, all the client services under an aggregated ODU-2 should be accommodated by the ports provisioned by the modules from the same module group. By doing this, we can avoid bringing in extra connecting fibers, of which each would waste two network or client service ports, between different module groups. Based on this key rule, we develop an algorithm called one-neck-to-grab (ONTG) to plan required hardware modules for *each network node*. We use the example in Fig. 2 to assist the description of the algorithm. The key principle of the algorithm is to "grab" an *aggregated* ODU-2 as a *neck* and to make mapping between an aggregated ODU-2 and its client services and the ports in a module group. A brief flowchart of this algorithm is shown in Fig. 3.

In Step 1, we create an empty list C of Φ-module groups, which is used to store planned Φ-module groups. Also, we sort aggregated ODU-2s based on their carried added/dropped client services. An ODU-2 carrying more added/dropped client services is ranked before an ODU-2 carrying fewer client services. This ranking step helps to reduce the required number of client service modules and module groups.

In Step 2, for each aggregated ODU-2, we examine each of the existing module groups in list C to see if there is a module group eligible for supporting the ODU-2 and its services. For this verification, there can be three scenarios. The first scenario is that there are one free network port and multiple free client service ports on the existing modules in the group. In this case, we use the existing free ports to support the current aggregated ODU-2 and its services. The second scenario is that the existing free ports are not sufficient and therefore we need to add more modules and meanwhile the new added modules can be accommodated within an existing module group (i.e., the total number of existing modules and new added modules in the group is not greater than Φ). In this case, we just add the required modules and serve the ODU-2 and its services. Step 3 in Fig. 3 supports these two scenarios. The last scenario is that both of the first two scenarios cannot pass. In this case, we say that there is no module group in the current group list C that is eligible for supporting the current aggregated ODU-2. Then we create a new module group to accommodate the current ODU-2 and add the new module group to list *C*. Step 4 implements this scenario.

After finishing serving the current ODU-2, we check whether the current aggregated ODU-2 is the last one. If so, we terminate the planning process and output the hardware modules; otherwise, we consider the next ODU-2 in the sorted ODU-2 list and repeat the same process.

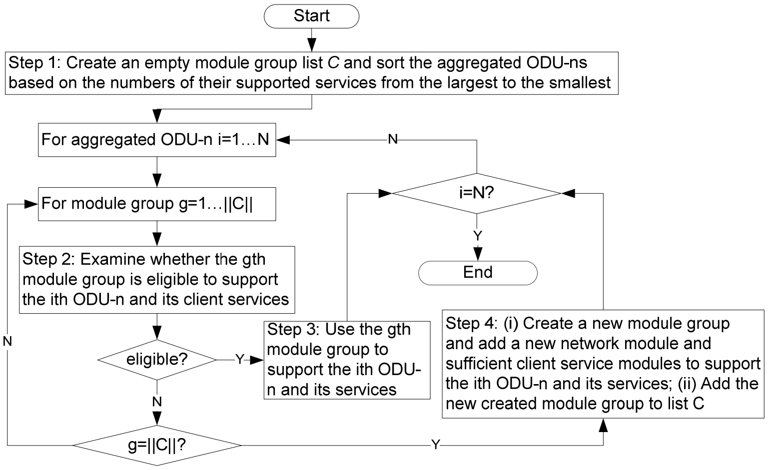

Figure 3: Flowchart of ONTG hardware module planning.

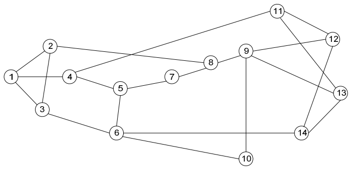

Figure 4: 14-node 21-link NSFNET.

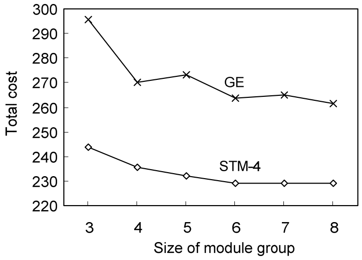

Figure 5: Results of hardware module planning.

5. Simulation and Performance Analyses

To evaluate the performance of the proposed module planning algorithm, we conducted simulations based on a 14-node and 21-link NSFNET network as shown in Fig. 4. The demand assumption is that each node pair has randomly 1~10 STM-4 or GE services. The wavelength data rate is ODU-2, i.e., 10 Gb/s. Thus, each wavelength supports 16 STM-4s or 8 GEs. The multi-hop grooming strategy [1] was applied to find a grooming result. The module planning algorithm input the grooming result and carried out module planning. We assumed that there is one type of network module that has one network port and four client service ports (i.e., X=1 and Y=4), and one type of client service module that supports eight client service ports (i.e., M=8). Each switch backplane can support different numbers of network or client service modules ranging from 2 to 8, (i.e., Φ=2 to 8). All the simulated services are unprotected. We also assumed that the nominal cost of a network module is 1.0, of a client service module is 0.5, and of a chassis for a module group with Φ=4 is 0.2. For other Φs, we scale their costs based on the cost of Φ=4. For example, if Φ=5, we scale its cost to $0.2\times5/4=0.25$.

Fig. 5 shows the simulation results. It can be found that for both of the traffic demand cases, the total cost decreases with an increasing module group size Φ. There are 6% and 11% cost reductions respectively for the STM-4 and GE traffic demands when Φ increases from 3 to 6. Such results are expected as the increase of module group size provides more flexibility and consequently better efficiency of organizing or arranging aggregated service groups in the module groups. Both of them approach a minimum when Φ is equal to 6. Meanwhile, it is also interesting to observe a saturation phenomenon. When Φ reaches 6 and if it is further increased, we do not see decrease of total costs for both of the cases. This phenomenon tells us that a large switch backplane is not necessary and also not a cost-effective solution; rather, depending on the ratio between the wavelength capacity and traffic demand granularity and hardware module sizes, there exists a most cost-effective switch backplane size. We have also considered other combinations of X, Y, M, and Φ and observed similar saturation behavior. Also, simulations where network and client service module costs were changed by +/-20% have shown similar saturation behavior.

6. Conclusion

We used multi-level tree architecture to represent the traffic demands and their layered relationship at a node. We developed an automatic hardware module planning approach that can support any type of layer-one switch backplane and module architecture. The performance evaluation indicated that a large switch plane is not necessary, and an optimal switch backplane size exists for each combination of traffic demand and hardware module size.

[1] K. Zhu and B. Mukherjee, “Traffic grooming in an optical WDM mesh network,” *IEEE JASC*, vol. 20, no. 1, pp. 122-133, January 2002.
[2] R. Dutta and G. N. Rouskas, “Traffic grooming in WDM networks: past and future,” *IEEE Network*, vol. 16, no. 6, pp. 46-56, 2002.
[3] R. Dutta, A.E. Kamal, G. N. Rouskas, *Traffic grooming for optical networks: foundations and techniques*, Springer, 2008.
[4] ITU-T Recommendation G.709, “Interfaces for the optical transport network (OTN)).”