Multicast traffic grooming in optical WDM networks

Lin, Rongping

2013

Lin, R. (2013). Multicast traffic grooming in optical WDM networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54911>

<https://doi.org/10.32657/10356/54911>

Multicast Traffic Grooming in Optical WDM Networks

Lin Rongping

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Doctor of Philosophy

2013

Acknowledgement

First, I would like to give my deepest thanks to my supervisor, Professor ZHONG Wen-De for his outstanding guidance and support during my Ph.D study. He opens the research door to me and illuminates the ways. He teaches me how to do a better research with his brilliant knowledge and profound insight in optical networking. I feel privileged to have the opportunity to do research under his guidance. Without his guidance, this dissertation could not have been possibly completed.

I would also like to thank Professor Moshe Zukerman in City University of Hong Kong and Professor Sanjay Kumar Bose in Indian Institute of Technology. We have cooperated during my Ph.D study. Without their guidance, insightful discussions and continuous encouragements, I cannot achieve today's results.

I thank all the members of OPTIMUS - Photonics Centre of Excellence lab for being great colleagues to work with, and in particular thanks to Professor Shum Ping, Mr. Chong Lui Tat, Mr. Poh Khoo Yong, Dr. Fu Songnian, Dr. Zhang Feng, Dr. Huang Qirui, Dr. Zhou Junqiang, Dr. Zhang Han, Dr. Wang Dawei, Dr. Wu Kan, Mr. Yu Xiaojun, Mr. Li Xiang, Mr. Li Yihui, Mr. Wang Zixiong, and Ms. Xiong Fei for their kind support.

Last but not the least, I would like to express my love and gratitude to my parents, my wife, and my sister, for their encouragements and support to overcome all the difficulties in my Ph.D. study and life.

Table of Contents

Summary.....	vi
List of Figures.....	ix
List of Tables	xiv
List of Acronyms	xv
Chapter 1 Introduction.....	1
1.1 Background and Motivation	1
1.2 Objectives	4
1.3 Contributions	5
1.4 Organization of the Thesis.....	7
Chapter 2 Literature Review	9
2.1 Introduction	9
2.2 Enabling Technology in Optical WDM Networks	9
2.2.1 One-to-One Lightpath	17
2.2.2 One-to-Many Light-tree	18
2.3 Multicast Traffic Grooming.....	24
2.3.1 Multicast Traffic Grooming with Dynamic Traffic	27
2.3.2 Multicast Traffic Grooming with Static Traffic	28
2.4 Summary.....	31
Chapter 3 Light-tree Configuration for Multicast Traffic Grooming with Dynamic Traffic	32
3.1 Introduction	32
3.2 Light-tree Configuration.....	33
3.2.1 Light-tree Division at Destination Branch Nodes (LTD-DBN).....	34

3.2.2	Light-tree Division - Adjacent Node Component (LTD-ANC).....	36
3.3	Multicast Traffic Grooming Algorithms	40
3.3.1	The LTD-DBN based Grooming Scheme (LTD-DBNG).....	40
3.3.2	The LTD-ANC based Grooming Scheme (LTD-ANCG).....	44
3.3.3	Link Cost in Routing.....	45
3.4	Performance Evaluation	46
3.4.1	Test Networks and Traffic Model	46
3.4.2	Blocking Probability Comparison.....	48
3.4.3	Comparison of Average Number of LOHTs and of BLOHTs Among Different Grooming Algorithms	51
3.4.4	Effect of Add/drop Ratios on BP Performance	56
3.5	Summary.....	59
Chapter 4	Multicast Traffic Grooming with Leaking Strategy with Dynamic Traffic...	60
4.1	Introduction	60
4.2	Routing with Add/drop Port Limitation	63
4.2.1	Constrained Light-tree Multicast Routing (CLMR) Algorithm.....	65
4.2.2	Optimal Routing.....	67
4.2.3	Simulation Results.....	68
4.3	Multicast Traffic Grooming Algorithms with Leaking Strategy.....	70
4.4	Simulation Results.....	73
4.5	Summary.....	83
Chapter 5	Design of Multicast Traffic Grooming with Static Traffic.....	84
5.1	Introduction	84
5.2	Mathematical (ILP) Formulations	86
5.2.1	Light-tree based Multicast Traffic Grooming Framework.....	86
5.2.2	Number of Variables and Constraints	90
5.3	Optimal Solution of Light-tree and Lightpath based Formulations	91

5.4	Heuristic Algorithms	96
5.5	Numerical Results.....	99
5.5.1	Results for the Six-node Test Network	100
5.5.2	Results for USnet Topology	104
5.6	Summary.....	107
Chapter 6	Constrained Multicast Traffic Grooming with Static Traffic	108
6.1	Introduction	108
6.2	Problem Description.....	110
6.3	Problem Formulation.....	115
6.3.1	Mathematical (ILP) Formulation	115
6.3.2	Number of Variables and Constraints	119
6.4	Performance Evaluations of the Constrained Light-trees.....	121
6.4.1	Running Time Comparison	121
6.4.2	Efficiency of the Proposed Method.....	122
6.4.3	Numerical Results of the Proposed Formulation	125
6.5	Heuristic Approaches	128
6.6	Numerical Results.....	131
6.6.1	Results for NSFNET Topology.....	131
6.6.2	Results for Usnet Topology.....	133
6.7	Summary.....	136
Chapter 7	Multicast Traffic Grooming in Tap-and-Continue WDM Mesh Networks	138
7.1	Introduction	138
7.2	Node Architectures	141
7.3	Mathematical (ILP) Formulation.....	146
7.3.1	The ILP Formulation.....	147
7.3.2	Number of Variables and Constraints	153

7.4	An Example of the ILP Formulation	153
7.5	Trail Routing.....	160
7.5.1	Node Adding Trail Routing (NATR) Algorithm	160
7.5.2	Optimal Trail Routing	163
7.5.3	Simulation Results.....	164
7.6	Heuristic Approach.....	166
7.7	Numerical Results.....	168
7.7.1	Results of the Six-node Network Topology	169
7.7.2	Results of NSFNET Topology	174
7.8	Summary.....	175
Chapter 8	Conclusions and Future Research.....	177
8.1	Summary.....	177
8.2	Recommendations for Future Research.....	180
Author's Publications		183
Appendixes		185
References		198

Summary

In optical wavelength-division multiplexing (WDM) networks, multiple wavelengths with gigabit transmission rates may simultaneously transmit in a terabit/second capable fiber. WDM optical networks appear to be the most promising transport technology to facilitate and sustain the bandwidth intensive multicast applications, such as IPTV, video conferencing and streamed video broadcasts, which are expected to be the major drivers for Internet traffic growth. As the bandwidth requirements of the most multicast connections are largely less than the bandwidth of a wavelength, multicast traffic grooming which grooms (or multiplexes) multicast connections with low bandwidth requirements into high-capacity wavelength channels makes resources, like wavelengths, electronic switches, routers and multiplexers, highly utilized. Special considerations are required in multicast traffic grooming as a tree topology would be an optimal basic structure to follow for their implementation. This thesis investigates the algorithm design, optimal network design and performance analysis of optical WDM networks with multicast traffic under both static and dynamic traffic scenarios.

Light-trees instead of lightpaths are used to support multicast connections because a light-tree offers one-to-many connection in one (electronic) logical hop. The sharing of light-trees is crucial to improve the utilization, but this may be considerably limited by the size of light-trees, because the probability of a large light-tree with many destinations to be shared by multiple multicast connections that have the same destinations of that light-tree is very low. To improve the light-tree utilization, we propose two multicast traffic grooming algorithms whereby light-trees are divided into small ones so that they can be efficiently utilized. When add/drop port resource is inadequate, the light-tree sharing becomes crucial

for the performance of networks. To further improve the light-tree utilization, we propose multicast traffic grooming algorithms with leaking strategy, which allow a light-tree to deliver the traffic of a multicast connection to nodes which are not in the destination set of the connection. The results show that significant improvement can be achieved with the leaking strategy.

The theoretical analysis of the multicast traffic grooming problem could guide the design of efficient and practical algorithms. In this study, we investigate the optimal design of network with multicast traffic grooming. A light-tree based Integer Linear Programming (ILP) formulation is proposed to minimize the network cost, and a heuristic algorithm is also proposed to achieve scalability and its performance is compared with the optimal solution.

Due to the large numbers of variables and constraints caused by combinations of light-tree destinations, the light-tree based ILP model can only be used in small size networks. To improve the scalability of the light-tree based ILP analytical model, the hop constrained light-trees in multicast traffic grooming is introduced. An ILP formulation is proposed for optimal assignments of hop constrained light-trees for multicast connections, so that the network throughput can be maximized. A heuristic algorithm with a polynomial complexity is also proposed and its performance is compared with the ILP solution.

Since using light-trees may lead to some serious negative side-effects because of light splitting, like non-uniform gain over the operating waveband, gain saturation and additional noise arising from intensive use of optical amplifiers, multicast traffic grooming in Tap-and-Continue (TaC) WDM networks is investigated. A network node with TaC devices can tap a small amount of incoming optical power for the local station while forwarding the remainder to an output port. We propose a simple and efficient node architecture with TaC mechanism. An ILP formulation with the objective of minimizing the network cost is

developed and a heuristic algorithm of polynomial complexity is proposed for use in large networks.

List of Figures

Figure 1-1: The low-loss regions of an optical fiber.....	2
Figure 1-2: Global IPTV subscriber forecast.....	3
Figure 1-3: Global IPTV service revenue	3
Figure 2-1: The architecture of an OXC [17]	10
Figure 2-2: One-to-one lightpaths in the physical layer and the logical layer	18
Figure 2-3: Multicast (s; d_1, d_2) on IP over WDM network. (a) Multicast in IP layer only. (b) IP multicast via WDM unicast. (c) WDM multicast with light splitting.....	19
Figure 2-4: A multicast node architecture with splitters	20
Figure 2-5: A multicast node architecture with two stages of splitters.....	21
Figure 2-6: Splitter-and-delivery (SAD) switch architecture	22
Figure 2-7: One-to-many light-tree in the physical layer and the logical layer	24
Figure 2-8: The architecture of a grooming capable MC-OXC	26
Figure 3-1: A light-tree from source 1 to destinations 3, 4, 5 and 6.....	34
Figure 3-2: Flow chart of Light-tree Division at Destination Branch Nodes (LTD-DBN) algorithm.....	34
Figure 3-3: (a) Original light-tree for multicast (1; 3, 4, 5, 6; $0.5C$). (b) Two divided trees. (c) New request (1; 3, 4; $0.5C$) groomed with multicast (1; 3, 4, 5, 6; $0.5C$). (d) New request (2; 5, 6; $0.5C$) is groomed with multicast (1; 3, 4, 5, 6; $0.5C$).....	35
Figure 3-4: Four categories of adjacent node components (sub-light-trees)	36
Figure 3-5: Flow chart of Light-tree Division - Adjacent Node Component (LTD-ANC) algorithm.....	38

Figure 3-6: Light-tree division - adjacent node component approach	39
Figure 3-7: Flow chart of LTD-DBN based grooming (LTD-DBNG) algorithm.....	40
Figure 3-8: Flow chart of Logical Layer Grooming algorithm	41
Figure 3-9: Flow chart of Build New Light-trees algorithm.....	42
Figure 3-10: An example for illustrating the operation of LTD-DBNG.....	43
Figure 3-11: NSFNET topology	错误！未定义书签。
Figure 3-12: COST239 topology	47
Figure 3-13: Blocking probability comparison in NSFNET, $E(X)=5$	49
Figure 3-14: Blocking probability comparison in COST239, $E(X)=4$	49
Figure 3-15: Blocking probability comparisons with different mean values of the multicast destination size (traffic load fixed at 140 erlangs in NSFNET).....	51
Figure 3-16: Comparison of the Avg. no. of LOHTs per multicast request in NSFNET, $E(X)=5$	52
Figure 3-17: Comparison of the Avg. no. of LOHTs per multicast request in COST239, $E(X)=4$	53
Figure 3-18: Comparison of the Avg. no. of BLOHTs in NSFNET, $E(X)=5$	53
Figure 3-19: Comparison of the Avg. no. of BLOHTs in COST239, $E(X)=4$	54
Figure 3-20: Comparison of the single objective function values in NSFNET, $E(X)=5$..	56
Figure 3-21: BP Comparison by varying ADR in NSFNET, when the traffic load is fixed and $E(X)=5$	58
Figure 3-22: BP Comparison by varying ADR in COST239, when the traffic load is fixed and $E(X)=4$	58
Figure 4-1: Multicast traffic grooming with leaking	61
Figure 4-2: Adjacent node component (ANC). (a) 1-hop light-trees (b) 2-hop branches	64

Figure 4-3: Flow chart of Constrained Light-tree Multicast Routing (CLMR) algorithm	66
Figure 4-4: Comparison of number of electronic ports required by a multicast connection for different routing algorithms	69
Figure 4-5: Comparison of number of wavelinks required by a multicast connection for different routing algorithms.....	70
Figure 4-6: Flow chart of Multicast Traffic Leaky Grooming (MTLG) algorithm	71
Figure 4-7: Flow chart of Multicast Traffic Hybrid Grooming (MTHG) algorithm	72
Figure 4-8: Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.3 (The error bars indicate the 95% confidence interval).....	75
Figure 4-9: Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.5 (The error bars indicate the 95% confidence interval).....	76
Figure 4-10: Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.9 (The error bars indicate the 95% confidence interval).....	76
Figure 4-11: Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.3.....	78
Figure 4-12: Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.5.....	79
Figure 4-13: Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.9.....	79
Figure 4-14: Blocking ratio vs. the network load when the add/drop ratio is 0.5 and leaking ratio threshold is 0.2	80
Figure 4-15: Blocking ratio vs. the add/drop ratio when network load is 50 erlangs and the leaking ratio threshold is 0.2.....	81

Figure 4-16: Blocking ratio vs. the leaking ratio threshold when network load is 80 erlangs and the add/drop ratio is 0.3.....	82
Figure 4-17: Light-tree sharing degree vs. the leaking ratio threshold when network load is 80 erlangs and the add/drop ratio is 0.3.....	82
Figure 5-1: (a) Lightpath based traffic grooming. (b) Light-tree based traffic grooming	85
Figure 5-2: USnet topology	92
Figure 5-3: The physical routes of light-trees (1→2) and (1→3).....	94
Figure 5-4: Flow chart of Sub-Light-Tree Saturated Grooming (SLTSG) algorithm	97
Figure 5-5: Cost comparison of different algorithms	100
Figure 5-6: Number of transmitters comparison of different algorithms	101
Figure 5-7: Number of receivers comparison of different algorithms	102
Figure 5-8: Number of wavelengths comparison of different algorithms	103
Figure 5-9: Cost comparison of SLTSG and LTD-ANCG.....	105
Figure 5-10: Comparison of SLTSG and LTD-ANCG in terms of number of electronic ports required.....	105
Figure 5-11: Comparison of SLTSG and LTD-ANCG in terms of number of wavelengths required.....	106
Figure 6-1: NSFNET topology	111
Figure 6-2: (a) 1-hop light-trees. (b) 2-hop light-trees. (c) 2-hop branches.....	112
Figure 6-3: Network throughput of formulations with different constraints in random networks.....	124
Figure 6-4: Network throughput of formulations with different constraints in NSFNET	125
Figure 6-5: Flow chart of Dividable Light-Tree Grooming (DLTG) algorithm	129

Figure 6-6: Network throughput comparison	132
Figure 6-7: Comparison of light-tree sharing degrees of LTD-ANCG and DLTG algorithms	133
Figure 6-8: Network throughput vs. number of transceivers per node	134
Figure 6-9: Light-tree sharing degree vs. number of transceivers per node	135
Figure 7-1: A trail $s \rightarrow d_1 \rightarrow s \rightarrow d_2 \rightarrow d_3 \rightarrow d_2 \rightarrow n \rightarrow d_4$ supporting a multicast connection request ($s; d_1, d_2, d_3, d_4$)	139
Figure 7-2: A TaC node architecture with multiple TCMs	142
Figure 7-3: A 1×8 TCM consisting of $\lceil \log_2 8 \rceil = 3$ stages of 1×2 switches	143
Figure 7-4: The proposed TaC node architecture with the full tap capability	144
Figure 7-5: A six-node network topology	153
Figure 7-6: Three node-adding scenarios	161
Figure 7-7: Flow chart of Node Adding Trail Routing (NATR) algorithm	162
Figure 7-8: Comparison of numbers of edges per connection request used	165
Figure 7-9: Flow chart of Multicast Trail Grooming (MTG) algorithm	167
Figure 7-10: Cost comparison	169
Figure 7-11: Comparison of numbers of higher layer electronic ports used	170
Figure 7-12: Comparison of numbers of transmitting electronic ports used	171
Figure 7-13: Comparison of numbers of receiving electronic ports used	172
Figure 7-14: Comparison of numbers of wavelengths used	173

List of Tables

Table 2-1: Variables and notations used in this thesis	12
Table 5-1: Ten multicast requests for the six-node network.....	93
Table 5-2: The cost and resource (wavelength, transmitter, and receiver) used	94
Table 5-3: Light-tree based logical routes of selected requests	94
Table 5-4: Lightpath based logical routes of selected requests.....	94
Table 5-5: Running times of four schemes	103
Table 6-1: Number of LOHTs with different constraints	113
Table 6-2: Number of constraints and variables for four different cases.....	120
Table 6-3: Running times of ILP formulations with different constraints.....	122
Table 6-4: Network throughput (total traffic is OC-186).....	126
Table 6-5: Average wavelength link utilization ($TR=RR=3$).....	126
Table 6-6: Average resource utilization ($W=3$)	126
Table 7-1: Ten multicast connection requests	154
Table 7-2: The cost and resource (wavelength, transmitter, receiver) usage.....	155
Table 7-3: Traffic routings of connection requests based on trail.....	157
Table 7-4: Physical routings of trails	158
Table 7-5: Simulation results in NSFNET	174

List of Acronyms

Abbreviations	Full Expressions
ANC	Adjacent Node Component
ADM	Add/drop Multiplexer
ADR	Add/drop Ratio
AG	Auxiliary Graph
BLOHT	Bridging node of Logical One Hop Trees
BP	Blocking Probability
CAPEX	Capital Expenditure
CLMR	Constrained Light-tree Multicast Routing
CN	Current Network
COST239	Ultra-High Capacity Optical Transmission Networks
CPU	Central Processing Unit
DWDM	Dense Wavelength Division Multiplexing
DLTG	Dividable Light-Tree Grooming
EDFA	Erbium-Doped Fiber Amplifier
HDTV	High Definition TV
IP	Internet Protocol
IPTV	Internet Protocol Television
ILP	Integer Linear Programming
LTD-DBN	Light-Tree Division - Destination Branch Node

LTD-DBNG	Light-Tree Division - Destination Branch Node based Grooming
LTD-ANC	Light-Tree Division - Adjacent Node Component
LTD-ANCG	Light-Tree Division - Adjacent Node Component based Grooming
LOHT	Logical One Hop Tree
LTPG	Logical Tree Path Grooming
MC-OXC	Multicast Capable Optical Cross Connect
MDMCT	Multiple Destination Minimum Cost Trail
MDT	Multiple Destination Trail
MH	Multi-hop
MPEG	Moving Picture Experts Group
MPH	Minimum-cost Path Heuristic
MST	Minimum Spanning Tree
MTD	Multicast Tree Decompose
MTLG	Multicast Traffic Leaky Grooming
MTHG	Multicast Traffic Hybrid Grooming
MTG	Multicast Trail Grooming
NATR	Node Adding Trail Routing
NP	Nondeterministic Polynomial
NSFNET	National Science foundation Network
OXC	Optical Cross Connect
O-E-O	Optical-Electronic-Optical
OC	Optical Carrier
PPH	Pruned Prim's Heuristic

PC	Personal Computer
RAM	Random Access Memory
RWA	Routing and Wavelength Assignment
SONET	Synchronous Optical Network
SDTV	Standard Definition TV
SLTSG	Sub-light-tree Saturated Grooming
SAD	Splitter-and-Delivery
SH	Single Hop
TaC	Tap-and-Continue
TAP	Tapping device
TCM	TaC Module
VoIP	Voice-over IP
WDM	Wavelength Division Multiplexing
WRS	Wavelength-routing Switch

Chapter 1

Introduction

1.1 Background and Motivation

Wavelength-division multiplexing (WDM) is an efficient technology to divide the bandwidth of a fiber strand into multiple wavelengths where these wavelengths can be simultaneously transmitted in the fiber. A single fiber strand can offer a bandwidth of more than Tbit/s [1-4] as shown in Figure 1-1. With traditional standard single mode fibers, there are two low-loss bands located at 1300 nm and 1550 nm, of approximate widths of 100 nm and 150 nm, respectively. With low-water-peak fibers, the attenuation peak (also called OH Peak) around 1400 nm is removed. The total bandwidth of such an optical fiber is approximately 50 THz, which provides an even wider low-loss band than a traditional fiber. These optical fibers potentially provide a tremendous bandwidth.

With the WDM technology, multiple wavelengths each with a data rate of 10 Gbit/s, or 40 Gbit/s, or even higher can be simultaneously transmitted in a fiber in a non-overlapping way. The number of wavelengths transmitted in a fiber is determined by the spacing between two adjacent wavelength channels. Dense WDM (DWDM) supplies more wavelengths than usual WDM technology on a single fiber with denser channel spacing, this provides larger capacity

to network. Optical WDM networks would become the dominant transport infrastructure not only in backbone but also metropolitan networks [5-9].

Figure 1-1: The low-loss regions of an optical fiber

Over the past two decades, there has been an explosive growth of the Internet traffic. Among these traffic, one-to-many multicast applications, such as Internet protocol television (IPTV), video conferencing, video-on-demand, streamed video broadcasts, multi-party online games are bandwidth intensive and expected to become the major drivers for Internet traffic growth. As shown in Figure 1-2, according to the report of the Multimedia Research Group [10], the number of global IPTV subscribers in 2013 will increase by more than two times, reaching 81 million, as compared to 2009, with an average growth rate of 32% per annum. This massive growth is also observed in the service revenue of the global IPTV market, as in Figure 1-3, where the revenue will grow from \$6.7 billion in 2009 to \$19.9 billion in 2013, with an average growth rate of 31% per annum. However, the current deployed networks are mainly designed for unicast connections. Without efficient multicast routing techniques, these network may not be able to support multicast traffic efficiently. New node architectures and methods should be developed to facilitate the multicast applications. These applications

require special considerations in routing because trees instead of paths would be an optimal basic structure to follow for their implementation.

Figure 1-2: Global IPTV subscriber forecast

Figure 1-3: Global IPTV service revenue

Generally, multicast applications tend to have much lower bandwidth demands than the capacity of a wavelength channel. For example, Standard Definition TV (SDTV) and High Definition TV (HDTV) only require about 6Mbit/s and 25Mbit/s transmission rates when

using Moving Picture Experts Group (MPEG)-2 technology [11], and live auction without video needs at most several Kbit/s transmission bandwidth. The disparity between the bandwidth offered by a wavelength and the bandwidth requirement of a connection request can be very large. It is very inefficient to use one wavelength to transmit only one multicast connection. For example, the bandwidth efficiency is about 0.25% when one HDTV channel is carried by one wavelength with 10 Gbit/s transmission rate.

The potential inefficiency that may be caused by this disparity has motivated the development of traffic grooming techniques that enable multiplexing or grooming of multiple connections or flows of low bandwidth to share a wavelength channel so that the overall bandwidth can be more efficiently utilized. Traffic grooming can also improve the utilization of other network resources, like electronic switches, routers, and multiplexers in the network. Comparing to unicast traffic, a multicast connection would occupy even more resources, and there would be more expectations to apply proper traffic grooming schemes to enhance the network performance.

1.2 Objectives

As discussed above, significant research efforts are needed to develop novel and efficient multicast traffic grooming techniques. This thesis is focused on multicast traffic grooming in optical WDM networks. A multicast traffic grooming policy is mainly concerned about how to accommodate multicast connection requests with an objective of maximizing the network throughput or minimizing the network blocking ratio, or minimizing the network cost. With a specific grooming method, network resource sharing, optical-electronic-optical (O-E-O) conversion or network equipment spending is manipulated to achieve the objective.

Light-trees are considered as a natural extension of lightpaths to support multicast traffic efficiently.

Thus, the objectives of this thesis are to study and investigate the light-tree based multicast traffic grooming approaches in terms of traffic grooming mechanism, optimized integer linear programming (ILP) algorithms and heuristic algorithms. The effectiveness in terms of network resource utilization, network cost spending or blocking probability, of various multicast traffic grooming methods are investigated in the thesis.

Specifically, in the context of dynamic multicast traffic, thesis firstly investigates and evaluates the sub-light-tree based multicast traffic grooming approaches, whereby light-trees are divided into small ones, and O-E-O conversion and network resource sharing are evaluated.

Secondly, in the context of static multicast traffic, the thesis develops ILP formulations for multicast traffic grooming problems, that can obtain the optimal solutions for the problems and guide the design of efficient heuristic algorithms.

1.3 Contributions

The original contributions of this thesis are listed as follows:

- In dynamic multicast traffic scenario, light-tree dividing methods are proposed, whereby a light-tree is divided into smaller sub-light-trees which can significantly improve the resource utilization. To achieve the tradeoff between the O-E-O conversion overhead and the network performance, two new grooming schemes, namely Light-Tree Division - Destination Branch Node based Grooming scheme (LTD-DBNG) and Light-Tree Division - Adjacent Node Component based Grooming scheme (LTD-ANCG), are developed and their performances are evaluated. Compared with existing algorithms,

these schemes not only can significantly reduce the request blocking probability, but they can also be implemented with reasonable electronic processing overhead, with LTD-ANCG performing better than LTD-DBNG but with greater complexity.

- In dynamic multicast traffic scenario, light-tree based multicast traffic grooming algorithms with leaking strategy are proposed, whereby multicast traffic may be groomed into light-trees in which some nodes are not in the destination set of a multicast connection. This leaking strategy improves the sharing of light-trees and add/drop ports, leading to lower blocking ratios. Two multicast traffic grooming algorithms with leaking strategy, namely Multicast Traffic Leaky Grooming (MTLG) and Multicast Traffic Hybrid Grooming (MTHG), are proposed. MTHG is an improvement over MTLG as it can attain higher light-tree sharing with less traffic leaked. Simulations have shown that the two proposed algorithms perform better than other existing algorithms at low add/drop port ratios.
- In the context of static multicast traffic, a light-tree based ILP formulation is proposed to minimize the network cost associated with the number of higher layer electronic ports and the number of wavelengths used. A heuristic algorithm, called Sub-light-tree Saturated Grooming (SLTSG), is proposed to achieve scalability. Simulations have demonstrated significant benefits of using a light-tree based design over a design that only uses lightpaths. Furthermore, an ILP formulation is proposed based on hop constrained light-trees. The approach of hop constrained light-trees improves the scalability by reducing the search space of the ILP formulation. A heuristic algorithm with hop constrained light-trees, called Dividable Light-Tree Grooming (DLTG) algorithm, is proposed with a polynomial complexity. Simulations have shown that the proposed DLTG heuristic performs better than other algorithms. It achieves the network throughput very close to that of the ILP formulation, but with far less running times.

- In the context of static multicast traffic, the multicast traffic grooming problem in Tap-and-Continue (TaC) networks is investigated, where a node can tap a small amount of incoming optical power for the local station while forwarding the remainder to an output port. A simple and efficient node architecture is proposed for the TaC mechanism. An ILP formulation is developed to minimize the network cost in terms of the number of higher layer electronic ports and the number of wavelengths used. A heuristic algorithm of polynomial complexity, called Multicast Trail Grooming (MTG), is proposed for use in large networks. Simulation results have shown that although the ILP formulation proposed here does not require multicast capable nodes, its performance is still very close to that of a light-tree based ILP formulation where network nodes have multicast capability. The solution obtained by the MTG algorithm is close to the ILP optimal solution and is shown to work efficiently.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 reviews the enabling technologies and multicast traffic grooming researches in optical WDM networks. According to the traffic pattern, multicast traffic grooming algorithms are divided into two categories as with static traffic and with dynamic traffic.

Chapter 3 proposes two light-tree dividing grooming methods, where a light-tree is divided into small ones to increase the resource utilization. The performances of the proposed algorithms are compared with other algorithms by simulations.

Chapter 4 proposes a new multicast traffic grooming scheme with leaking strategy, whereby traffic may leak to nodes which are not the destinations of connection. Two

proposed algorithms are compared with other algorithms without traffic leaking by simulations.

Chapter 5 presents a light-tree based ILP formulation to optimally design network with multicast traffic, a heuristic algorithm is proposed.

Chapter 6 presents an ILP formulation with hop constrained light-trees to scale the model of Chapter 5 to large networks, and a heuristic algorithm with hop constrained light-tree is also proposed.

Chapter 7 presents the research about multicast traffic grooming in TaC networks, which includes a proposal of new node architectures, an ILP formulation of the multicast traffic grooming on TaC networks, a heuristic algorithm for the problem and the routing investigation.

The thesis is concluded in Chapter 8, and recommendations for future research are given in this chapter.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a review of multicast traffic grooming in optical WDM networks. Firstly, an overview of enabling technology is discussed, which contains one-to-one lightpath [12-14] and one-to-many light-tree [15, 16]. Secondly, a review of the existing research on multicast traffic grooming in optical WDM networks is presented, which includes dynamic traffic scenario and static traffic scenario.

2.2 Enabling Technology in Optical WDM Networks

The initial deployment of WDM networks is in point-to-point long haul transmission networks, known as the first generation optical networks, which mainly provide large bandwidth to transmission links. With the continuous growth in the traffic, the network nodes get overloaded with electronic processing as all traffic in fiber transmission links has to be processed and switched in the electronic domain for the next hop transmission. More specifically, optical signals arriving at a network node are first converted to electrical signals,

which are then switched by an electronic cross connect; after being electronically processed and switched, the electrical signals at the outputs of the electronic cross connect are converted back to respective optical signals before being transmitted to the next hop. So the first generation optical network suffers the problem of the electronic processing and switching bottleneck at network nodes. Meanwhile, a lot of energy is consumed at these very high data rates.

Figure 2-1: The architecture of an OXC [17]

With the rapid development of optical network technologies, new optical elements have emerged to enable switching traffic optically, which is referred to as the second generation optical network. Some significant elements, such as Erbium doped fiber amplifiers (EDFAs) [18], which can significantly increase the transmission distance of fiber links, and optical cross connects (OXCs) [17, 19-21], which can bypass some optical signals from electronic

processing, i.e., some optical signals do not need to undergo optical to electrical and then electrical to optical conversion (O-E-O conversion). Figure 2-1 shows an OXC architecture. The use of OXCs in optical networks makes optical networking practical. Specifically, the OXC provides wavelength routing capability, making the optical signals bypass the OXC without undergoing O-E-O conversion if they do not terminate at the node. An add port is for an optical signal to be sent into the optical layer as the source and a drop port is for an optical signal to be extracted from the optical layer as the terminal of the optical signal. An optical network consisting of OXCs and optical links can improve network capacity and is an extension of the point-to-point first generation optical network.

If a connection traverses multiple optical links, the links have to be assigned with the same wavelength. Thus, two different wavelengths should be assigned to two connections that share a common fiber link to avoid interference. This is called the wavelength continuity constraint, and it can be relaxed by deploying wavelength converters [22-26] at nodes. The wavelength converter is another key component in WDM wavelength-routed networks. It can shift a wavelength to a different one. If wavelength conversion is employed in an OXC, an incoming optical signal can be switched to a different free wavelength if the wavelength of the incoming signal is busy, leading to lower blocking probability.

In our work, we consider a undirected network graph $G(V, E)$, where V is the set of network nodes and E is the set of undirected optical fiber links. We assume that each fiber link consists of two optical fibers transmitting in opposite directions and each fiber has W wavelengths each of a bandwidth capacity of C . We also assume that the same wavelength has to be assigned to all links of a lightpath (or light-tree) so that the wavelength continuity constraint is met. For the clarity of presentation, Table 2-1 lists all the variables and notations used in this thesis.

Table 2-1: Variables and notations used in this thesis

Variable & Notation	Description
$G(V, E)$	A undirected network graph denoting a network
V	The set of nodes of G
E	The set of edges of G
$ V $	The number of nodes of G
$ E $	The number of edges of G
W	The number of wavelengths per fiber
C	The capacity of a wavelength
$(s; D)$:	a 2-tuple of the elements s and D representing a multicast connection request with a full wavelength capacity requirement, where s is the source of the request, $D = \{d_1, d_2, d_3, \dots\}$ is the destination set of the request
$(s; D; f)$	A multicast request where s denotes the source, $D = \{d_1, d_2, d_3, \dots\}$ denotes the destination set, f is the bandwidth requirement
AG	An auxiliary graph
CN	A current network, including all resources information
$LOHT_r$	The list of logical one hop trees occupied by request r
NLT_r	The list of new light-trees needed by request r
$NSLT_r$	The list of new sub-light-trees needed by request r
p_{ij}	The cost of link (i, j) of auxiliary graph constructed in optical layer
l_{ij}	The cost of link (i, j) of auxiliary graph constructed in logical layer
$out_degree(i)$	The number of edges leaving node i

m and n	two endpoints of a physical fiber link
g	The maximal node degree in a network
$(i \rightarrow J)$	A LOHT (light-tree) sourcing at node i terminating at the destination set J in the logical layer, where $J = \{d_1, d_2, \dots, d_m\}$. A light-tree in the optical layer maps to a LOHT in the logical layer
J_i	LOHT destination sets from node i
λ_{iJ}	a Boolean variable. It is 1 if the connection request traverses LOHT $(i \rightarrow J)$, otherwise 0
Q_j^{iJ}	a Boolean variable. It is 1 if the connection request traverses LOHT $(i \rightarrow J)$ and node j is included in the set J . i.e. $j \in J$, otherwise 0
H_n	an integer variable used to break loops, which could be the lower bound of logical hops from source to node n
F_{mn}^{iJ}	an integer variable denoting the number of streams traverses link (m, n) in LOHT $(i \rightarrow J)$. Each node in J needs one stream, therefore, there are $ J $ streams out of source node i
w	wavelength index, starting from 1 and ending at W
Z	a sufficiently large integer number
α	relative cost of a higher layer electronic port
β	relative cost of a wavelength deployed in the network
P_{mn}	number of fibers interconnecting node m and node n in the physical layer. It is 0 if m and n are not physically adjacent to each other. In this study, we assume $P_{mn} = P_{nm} = 1$ if there is a physical fiber link between nodes m and n
TR_i	number of tunable transmitters at node i
RR_i	number of tunable receivers at node i

φ	number of wavelengths used in the network
y_w	a Boolean variable. It is 1 when wavelength w is used in the network, otherwise 0
L_{iJ}	number of light-trees for LOHT ($i \rightarrow J$) from node i to the destination node set J , $J \in J_i$
L_{iJ}^w	number of light-trees on wavelength w , for LOHT ($i \rightarrow J$) from node i to the destination node set J . $L_{iJ}^w > 1$ means that there are multiple link disjoint light-trees from node i to the set J using the same wavelength w
$P_{d,mn}^{iJ,w}$	a Boolean variable. It is 1 when the path from the root i to a destination d traverses link (m, n) and uses the wavelength w of the link, where $J \in J_i$, $d \in J$, otherwise, it is 0
$M_{mn}^{iJ,w}$	a Boolean variable. It is 1 when the light-tree on the wavelength w for LOHT ($i \rightarrow J$) traverses the link (m, n) , where $J \in J_i$, otherwise 0
λ_{iJ}^t	a Boolean variable. It is 1 if multicast request t traverses LOHT ($i \rightarrow J$) in the logical layer, otherwise 0
$Q_{iJ,n}^t$	a Boolean variable. It is 1 if multicast request t traverses LOHT ($i \rightarrow J$) in the logical layer and node n belongs to J , i.e. $n \in J$, otherwise 0
H_n^t	an integer variable. It is used to break loops of the traffic routing in the logical layer, which can be the lower bound of logical hops from the source s_t of multicast request t to node n
R	set of all multicast connection requests
$(s_t; D_t; f_t)$	denote a multicast connection request t ($1 \leq t \leq R $), where s_t denotes the source, $D_t = \{d_1, d_2, d_3, \dots\}$ denotes the set of destinations, f_t is the bandwidth requirement

R'	set of connection requests with sub-wavelength requirements in R
R''	set of connection requests with a full wavelength capacity requirement in R , so $R = R' \cup R''$
J'_i	all LOHT (light-tree) destination sets with the constraint, whose root is node i . Here the constraint can be any one of the four cases discussed
J''_i	all LOHT (light-tree) destination sets, which may be used by connection requests with a full wavelength capacity requirement, whose source is node i ; this set can be obtained from the destinations of requests in R''
U_t	a Boolean variable. It is 1 if multicast connection request t is successfully accommodated; otherwise it is 0
$Deg(i)$	nodal degree of node i
A_q^w	number of add ports of WRS for wavelength w at node q
B_q^w	number of drop ports of WRS for wavelength w at node q
A_q	number of transmitting electronic ports at node q
B_q	number of receiving electronic ports at node q
$U_{i,w,k}$	a Boolean variable. It is 1 when trail (i, w, k) is built, otherwise 0
$S_{i,w,k}^{m,q,n}$	a Boolean variable. It is 1 when the physical routing of trail (i, w, k) has links $m \rightarrow q \rightarrow n$ to forward the incoming stream on link (m, q) to link (q, n) in the trail, where $m \in adj(q) \cup \{q\}, n \in adj(q)$. At the source node i , $S_{i,w,k}^{i,i,n}$ indicates the first physical hop $i \rightarrow n$ is traversed or not
$F_{i,w,k}^{mn}$	an integer commodity-flow variable. Each destination node of a trail

	needs one unit of commodity. So a total of $\sum_q T_{i,w,k}^q$ units of commodity flow out of the source i for trail (i, w, k) . $F_{i,w,k}^{mn}$ is the number of units of commodity flowing on link (m, n) for trail (i, w, k) , which is equal to the number of destinations in the downstream of link (m, n) of the trail
$M_{i,w,k}^{mn}$	a Boolean variable. It is 1 when trail (i, w, k) traverses link (m, n) , otherwise 0
$T_{i,w,k}^q$	a Boolean variable. It is 1 if node q uses a receiver to receive the tapped/dropped signal on trail (i, w, k) , otherwise 0
$\lambda_{i,w,k}^t$	a Boolean variable. It is 1 if multicast connection request t traverses trail (i, w, k) , otherwise 0
$Q_{i,w,k}^{t,q}$	a Boolean variable. It is 1 if node q is the end node or a tap-receiving node of trail (i, w, k) occupied by connection request t , otherwise 0. This variable should be set to $\lambda_{i,w,k}^t$ times $T_{i,w,k}^q$
F_{mn}	an integer commodity-flow variable. It denotes the number of units of commodity flowing on link (m, n) , and is also the number of destinations downstream of link (m, n)
M_{mn}	a Boolean variable. It is 1 if the trail traverses link (m, n) , otherwise 0
$S_{q,n}^m$	a Boolean variable. It is 1 when the physical routing includes links (m, q) and (q, n) to forward an incoming stream where $m \in adj(q) \cup \{q\}, n \in adj(q)$, otherwise 0. At the source node s , $S_{s,n}^s$ indicates the first physical hop $s \rightarrow n$ is traversed or not

2.2.1 One-to-One Lightpath

In a wavelength-routed WDM network, data is transported in an all-optical channel, i.e. a lightpath, for an end-to-end connection from a source node to a destination. This may span multiple fiber links, and the intermediate nodes (OXC) of a lightpath would route the incoming wavelength to an output port without O-E-O conversion. These lightpaths act as the conduits for upper layer traffic, and a lightpath connects two end nodes without O-E-O conversion intermediately. The end nodes of a lightpath that may be geographical far away each other are connected by one logical hop. In the logical layer, each link denotes a lightpath from the source to the destination in the optical layer. The set of lightpaths in the optical layer of a WDM network forms the virtual (logical) topology of the network. In other words, the lightpaths are managed and operated based on the logical topology over the optical fiber topology and the logical topology can be reconfigured dynamically according to the traffic demands. For example, as shown in Figure 2-2, there are two lightpaths in the physical layer: one is from node 1 to node 5, where nodes 2 and 3 are intermediate nodes; another is from node 4 to node 6, where node 5 is an intermediate node. No matter how many links a lightpath traverses in the physical layer, only one link is generated in the logical layer for this lightpath to denote the connection of the two end nodes. It is noted that intermediate nodes would forward the incoming light signal directly to the next hop in the optical domain without O-E-O conversion. For example, the lightpath from node 4 to node 6, traverses node 5 as intermediate node, and node 5 would forward the optical signal to node 6 directly in the optical domain.

The wavelength continuity constraint is imposed if the light signal is transmitted on the same wavelength on all links of the lightpath without using wavelength converters. E.g., in Figure 2-2, the lightpath from node 1 to node 5 traverses link 1-2, 2-3 and 3-5, three links are assigned with the same wavelength λ_1 to obey the wavelength continuity constraint.

Figure 2-2: One-to-one lightpaths in the physical layer and the logical layer

2.2.2 One-to-Many Light-tree

Although lightpath is well suited for one-to-one unicast traffic, it may not be the best solution for multicast traffic as that needs one-to-many connection between the source node and a set of destination nodes. There are two ways to route a multicast request in a WDM network deployed for unicast connection [27, 28]. In Figure 2-3, we use the notation $(s; d_1, d_2, \dots, d_n)$ to represent a multicast of a full wavelength bandwidth from source s to destinations $d_1, d_2, \dots, d_n$. Multicast session $(s; d_1, d_2)$ is built in the IP layer in the Figure 2-3(a), and we can see that the multicast session is supported by IP routers and lightpaths (or logical links in the logical layer). IP routers are in charge of copying every multicast packet and forward them to downstream routers. So every multicast packet undergoes O-E-O conversions at IP routers on the multicast tree, which will burden the IP router's electronic processing capability, inducing a long latency of information transmission and decreasing the network throughput. Even this scheme can accommodate multicast sessions with fewer link resources than the method in Figure 2-3(b), but the electronic processing overhead is a fatal problem for deployment, because the electronic processing rate is the bottleneck for the huge wavelength bandwidth

and high electronic processing rate would consume a large amount of electrical power and cooling the device would become a problem [29-31].

Figure 2-3: Multicast (s; d_1 , d_2) on IP over WDM network. (a) Multicast in IP layer only. (b) IP multicast via WDM unicast. (c) WDM multicast with light splitting

In Figure 2-3(b), IP multicast via WDM unicast can avoid the O-E-O conversion by constructing multiple unicast lightpaths. A multicast request is treated as multiple unicasts from the source to each destination. As shown in Figure 2-3(b), the request (s; d_1 , d_2) is divided into two unicast connections (s; d_1) and (s; d_2) and hence two separate lightpaths are derived from the source s to destinations d_1 and d_2 . With this method, conversions between optics and electronics only happen at the source and the destinations of the multicast request, which would not incur extra electronic processing at the intermediate nodes of a lightpath. However, this method consumes large bandwidth resources as multiple lightpaths have to be built [32] for one multicast connection, leading to higher consumption of bandwidth resource and higher blocking probability. Compared with the method of multicast in the IP layer of Figure 2-3(a), this method can avoid extra O-E-O conversion at the expense of consuming more bandwidth resources to accommodate a multicast request.

The concept of light-tree was introduced in [15], which is an extension of the lightpath concept. A light-tree is a one-hop one-to-many connection between a source and a set of

destinations in the optical layer. Using light-tree to support multicast requests can significantly improve efficiency and reduce cost. If the wavelength-continuity constraint is applied then all the optical links of a light-tree must use the same wavelength. Figure 2-3(c) shows the multicast routing in a WDM network with nodes capable of light splitting (copying). There is a tree topology (light-tree) built in the optical domain from the source to the destinations. This also means that the multicast information stays in the optical domain from the source which injects the light into the destinations which extract the light out. This is called one hop routing. As per the case of IP multicast via WDM unicast in Figure 2-3(b), electronic and optical conversion happens at the source and destinations, and no extra overhead incurs at the intermediate nodes of the light-tree, which can improve the network throughput and guarantee the end-to-end delay of multicast transmission. With splitters being installed in nodes, multicast in WDM networks becomes easier and more efficient as the light splitting is much more efficient and cheaper than copying packets electronically.

Figure 2-4: A multicast node architecture with splitters

Figure 2-5: A multicast node architecture with two stages of splitters

Several multicast capable optical cross connects (MC-OXCs) have been reported [15, 27, 28, 33-35]. References [15, 33] proposed a MC-OXC with splitters that can do light splitting optically as shown in Figure 2-4. A splitter can split an incoming optical signal to multiple outgoing optical links for downstream destinations achieving information replication optically and reducing costs without using O-E-O conversion. As shown in Figure 2-4, the signal arriving on wavelength λ_b from input link 1 bypasses the node. The signal arriving on wavelength λ_a from input link 2 is switched to splitter bank X where it is split into three identical copies. Two replicas are switched to outgoing links 1 and D by the small optical switch separately, the other one is dropped locally. This MC-OXC architecture is straightforward, but the multicast capability is constrained by the size of the splitter bank. In [27, 28], another MC-OXC architecture with splitters was proposed, as shown in Figure 2-5, which includes two stages of splitters. Reference [34] reported a splitter-and-delivery (SAD) switch, as shown in Figure 2-6, having both multicasting and point-to-point switching capability. A SAD switch consists of $1 \times Q$ splitters, optical gates (G) and 1×2 switches (SW).

As shown in Figure 2-6, a splitter splits an input optical signal into Q identical copies, each of which is subsequently switched to an associated output port by a 1×2 switch. The SAD switch is strictly non-blocking and has multicasting capability. Using the SAD, several MC-OXCs were derived [34]. Some improvements of this SAD architecture were reported in [35], where the fixed $1 \times Q$ splitters were replaced by configurable splitters which can split signals into a variable number of outputs.

Figure 2-6: Splitter-and-delivery (SAD) switch architecture

Multicast transmission in an optical network would typically require a light-tree rooted at the source node and traversing all destination nodes. To find a light-tree with the minimal edge cost in a graph is the Steiner tree problem [36]. Given edge costs, the problem of finding a Steiner tree with the minimum total edge cost is known to be NP-complete [37, 38]. It is therefore not scalable, so heuristic algorithms are needed. Two efficient heuristic algorithms have been reported in [39, 40]. These are:

- Pruned Prim's Heuristic (PPH) [39]: It derives a minimum spanning tree (MST) using Prim's MST algorithm, then prunes unwanted arcs in the MST.

- Minimum-cost Path Heuristic (MPH) [40]: It constructs the tree by adding destinations one by one that is the nearest to the partially built tree and adding the shortest path to the partially built tree.

In this work, whenever required, the MPH is used to find the minimum cost Steiner tree and the Dijkstra algorithm is used to find the shortest path between nodes. The tree topology calculated by multicast algorithms is a light-tree in optical layer, and branch nodes of the light-tree can split the income light into several identical copies for downstream destinations.

The problem of establishing light-trees for multicast connection requests is multicast routing and wavelength assignment problem [41], which are two tight coupling sub-problems. The same wavelength has to be assigned to every link of a light-tree for the wavelength continuity purposed [42]. If wavelength conversion is applied, the constraint is relaxed. Some work on the multicast routing and wavelength assignment problem without wavelength continuity can be found in [43-45].

We give an example to show the detail of light-tree in physical layer and logical layer as Figure 2-7. Two light-trees are built in physical layer: the first one has node 1 as its source node, node 3 and 4 are destination nodes, and node 2 is an intermediate node which also do the light splitting in the optical domain to duplicate the information for node 3 and 4; the second light-tree has node 4 as source node, and node 3 and 6 as destinations, where node 5 is an intermediate node. The transmission from the source to all destinations takes only one hop and the signal stays in the optical domain from the source to the destinations. In the logical layer, the specific routing information of the light-trees are concealed, only show the connectivity between source node and destinations, where the first light-tree is denoted as an one hop tree from node 1 to node 3 and 4, and the second light-tree is denoted as another one hop tree from node 4 to node 3 and 6, and intermediate nodes of the light-trees are not shown

in the logical layer. We call this logical connection in logical layer for light-tree as Logical One Hop Tree (LOHT) which is an extension of the logical link for lightpath.

Figure 2-7: One-to-many light-tree in the physical layer and the logical layer

2.3 Multicast Traffic Grooming

Since the bandwidth requirement of a connection request is usually sub-wavelength, traffic grooming technology is used to groom multiple connections into a wavelength channel to increase the utilization of the resources. Traffic grooming in unicast has been extensively studied [46-57]. In order to aggregate several low transmission rate connections into one wavelength, the architecture of OXC needs to be improved with the grooming function that can switch the traffic at finer granularities than the wavelength granularity. In the literature [49], two general architectures are devised. They are based on the OXC architecture of Figure 2-1 with a grooming fabric (grooming function) adding into. Achieving efficient traffic grooming in WDM networks has been a challenging research topic in recent years. Based on

whether or not the connection requests are known a priori, traffic grooming is categorized as dynamic or static traffic grooming.

In dynamic traffic grooming, connection requests arrive dynamically (typically assumed to follow a Poisson process) and both routing and wavelength assignment are dynamically decided for new connection requests. The target is to accommodate the requests efficiently and to maintain acceptable blocking probability [48, 55-60]. Chen et al. [61] proposed a hierarchical framework for dynamic traffic grooming that can scale to large networks. The authors of [62-66] developed analytical models to evaluate the blocking performance of unicast traffic grooming with dynamic traffic.

In the static traffic case, the matrix of fixed end-to-end traffic demands is known a priori. This model is applicable to situations where traffic demands between any two nodes do not change for a relatively long time. Studies on static traffic grooming are useful for network design and planning purposes. In this case, an optimal solution for traffic grooming may be obtained [49, 67-72], usually using an Integer Linear Programming (ILP) method, with the objective of minimizing the resources used, e.g., wavelengths, wavelinks (a wavelink is a wavelength channel in a particular link it traverses) or add/drop ports, or of maximizing the network throughput. As the goal of traffic grooming is to efficiently groom multiple independent low bandwidth streams into high bandwidth wavelength channels subject to some resource constraints, it may be viewed as a combination of the following three optimization sub-problems [67]:

- Virtual topology design [73-79]: find the optimal set of lightpaths to form the logical layer.
- Traffic routing: route traffic on the virtual topology based on the given traffic demand matrix.

- Lightpath routing and wavelength assignment (RWA) [22, 80-86]: solve the lightpath RWA problem in the physical layer.

However, extending traffic grooming design to an environment that includes multicast traffic is important because multicast applications, such as IPTV, video conferencing and streamed video broadcasts, are likely to become a very significant Internet traffic component in the near future. These multicast applications tend to require only sub-wavelength capacities, so efficient multicast traffic grooming techniques are needed to support them. The node architecture with multicast capability and traffic grooming capability is actually a multicast grooming function adding into the MC-OXC architecture [15, 87], shown in Figure 2-8. The MC-OXC supplies the multicasting capability and wavelength switching same as the node function of Figure 2-4, 2-5 or 2-6. We will give reviews of research on multicast traffic grooming with dynamic and static traffic next.

Figure 2-8: The architecture of a grooming capable MC-OXC

2.3.1 Multicast Traffic Grooming with Dynamic Traffic

In the dynamic traffic case, multicast connection requests arrive dynamically and both routing and wavelength assignment are dynamically decided for new requests [87-101]. The objective is to maximize the resource utilization while maintaining an acceptable level of blocking probability. Furthermore, multicast traffic from a single source to many destinations normally follows a tree topology. Therefore, for multicast applications, it is natural that the light-tree can significantly improve efficiency and reduce cost.

In [87], both single hop (SH) and multi-hop (MH) algorithms were proposed. In SH, connections with the same source and destinations can be groomed together. Here, the optical signal transits from the source to the destinations in one logical hop without undergoing any electronic processing and queuing delay. This approach is simple to implement but it is not efficient, since the probability of multiple requests having exactly the same source and destinations is very low. The MH algorithm grooms traffic same as SH algorithm first. If it cannot groom the traffic, MH algorithm would searches for a light-tree with the same destination nodes as the new request and with sufficient available bandwidth. This light-tree in conjunction with a lightpath (an existing one or a new one) from the source of the new request to the root of this light-tree is then used to support the new request. This provides routing to the new request in two hops in the logical layer and would be more likely to succeed than SH. However, since the probability of multiple connection requests having exactly the same destinations is still very low, the light-tree sharing is quite poor. In [91], a Multicast Tree Decompose (MTD) algorithm was proposed to further improve the blocking probability by grooming traffic partially to several light-trees. However, as the sharing limitation, achieving high utilization with low blocking still remains a challenge in this system. The work in [92] studied multicast traffic grooming using a novel light-tree

architecture, where a light-tree can be dropped, branched, and extended when necessary and proposed a new auxiliary graph model for multicast traffic grooming.

A multicast request can also be accommodated by using a set of lightpaths that forms a source-destinations tree at the logical layer (i.e. a logical tree) [102]. When a multicast request is admitted, bandwidths of these lightpaths are allocated to meet its required traffic demand. As grooming would occur on each of these lightpaths of the logical tree, we call this method the Logical Tree Path Grooming scheme (LTPG). A branch node of the logical tree duplicates and forwards traffic in the electronic domain. Though conceptually simple, it requires O-E-O conversion and routing for each data packet at each node of the logical tree. This conversion and routing overhead lead to long latency and router overload [27, 28, 103-105] and significant energy consumption implications [29-31, 106-110].

2.3.2 Multicast Traffic Grooming with Static Traffic

With static traffic scenarios, multicast traffic grooming is similar to unicast traffic grooming, which is to efficiently groom multiple low bandwidth traffic streams into higher bandwidth wavelength channels. Multicast traffic grooming is to groom one-to-many multicast traffic streams into lightpaths and light-trees, while unicast traffic grooming is to groom one-to-one unicast traffic streams into lightpaths. The multicast traffic grooming problem may be viewed as a combination of the following three sub-problems

- virtual topology design: find a set of lightpaths/light-trees to form the logical layer
- traffic routing: route multicast traffic demands on the virtual topology
- lightpath/light-tree routing and wavelength assignment (RWA): solve the RWA problem of the lightpaths/light-trees in the physical layer

A virtual topology (i.e. the logical layer of a network) can be either formed by a set of lightpaths [111-113] or a set of light-trees [114, 115]. In a lightpath-based logical layer, each link of the logical layer represents a lightpath between two end nodes that may be geographically far away from each other. These lightpaths act as the conduits for upper layer multicast applications. Low bandwidth multicast connection requests find their routing paths (a tree session) in the logical layer, using logical links that have sufficient free bandwidth. Electronic packet switching is used at the connecting nodes of logical paths (end-points of lightpaths), implementing traffic forwarding from an upstream lightpath to downstream lightpaths with O-E-O conversion. In this paradigm, traffic grooming occurs on each of the lightpaths. With the lightpath-based virtual topology, the authors of [111] and [112] proposed an ILP formulation of multicast traffic grooming, with the objective of minimizing the cost of the network in terms of the number of Add/Drop Multiplexers (ADM) and the number of wavelengths used.

Light-tree is an extension of the lightpath concept, which is natural for supporting multicast applications with the one-to-many connection in one hop without interruption at intermediate nodes, and with potentially higher efficiency [15]. In the logical topology, a light-tree is presented as a logical one hop tree (LOHT) which is from a node (the source of the light-tree) to a set of nodes (the destinations of the light-tree). In this way, the traffic from the source to all the destinations is transported all-optically in the physical topology. The authors of [114] presented a tri-partite graph model to solve the multicast flow aggregation problem under a static traffic scenario. In order to improve light-tree utilization, multicast traffic aggregated to a light-tree is allowed to leak to unrelated nodes, which is called “tail waste”. However in [114], a multicast connection request is accommodated by a single light-tree, which may result in high tail wastage (i.e., low utilization of light-trees). In [115], an ILP optimization problem is formulated to design a light-tree based logical topology with

delay bounds. Here, the light-trees are somehow randomly chosen which reduces the complexity of the problem, but the optimal result may be not achieved.

In [116], the routing information of all multicast connection requests is known a priori and wavelength conversion is used. Therefore, the multicast traffic grooming problem was reduced to a bin packing problem on each link. A multicast tree with full wavelength conversion is constructed in [33] by grooming the multicast traffic in a link-by-link fashion. While the assumption of wavelength conversion would lead to high resource utilizations, it would require extensive O-E-O conversion at every node of the network. The authors in [117] modeled the network design and the session provisioning problem as an integer nonlinear programming formulation with the objective function of minimizing the cost of the electronic equipment. The formulation also determined whether or not an add/drop port is required at a node.

Because of the inherent reliability, the ring network architecture has been widely deployed in the current network infrastructure. Multicast traffic grooming researches are also carried out with ring network topologies [118-121], to efficiently deal with the increasing multicast traffic in ring networks.

Unicast traffic grooming is NP-complete problem [49, 50, 69, 122]. Since the unicast is a special case of the multicast with only one destination, and lightpath is also a special case of light-tree with only one leaf node, then by generalization, the multicast traffic grooming based on light-trees is also NP-complete. Using the ILP method to optimize the multicast grooming is not scalable to large networks, so it is needed to propose some practical heuristic algorithms that are scalable.

2.4 Summary

This chapter first reviewed the enabling technology in optical WDM networks. Both lightpath and light-tree are given brief descriptions in physical layer and logical layer, which are the transmission conduits for the upper layer applications. The research works of multicast traffic grooming with dynamic and static traffic are reviewed, where in dynamic case the connection requests arrive dynamically, and in static case all connection requests are known a priori. Furthermore, multicast traffic grooming based on lightpath and light-tree is reviewed accordingly, where lightpath-based one can use the existing network facility, but light-tree based one needs to install optical splitter to duplicate traffic.

From the literature review, it is clear that in both dynamic and static traffic scenarios, multicast traffic grooming would benefit much from light-tree transmission scheme since a light-tree provides one-to-many connection in one hop with potentially higher efficiency. This is the reason why we focus our research on light-tree based traffic grooming in both dynamic and static traffic scenarios.

Chapter 3

Light-tree Configuration for Multicast Traffic Grooming with Dynamic Traffic

3.1 Introduction

As reviewed in Chapter 2, single hop (SH) and multi-hop (MH) grooming schemes [87] are not efficient since the probability of multiple requests having the same destinations is low; while the logical tree path grooming (LTPG) algorithm [102] is expected to have a much higher efficiency, it requires O-E-O conversion at each node of the logical tree and may incur queuing delay at each of those nodes. It is important to strike a good tradeoff between the blocking performance and the electronic processing overheads. To address the above problems, this chapter proposes two new grooming schemes, called Light-Tree Division - Destination Branch Node based Grooming scheme (LTD-DBNG) and Light-Tree Division - Adjacent Node Component based Grooming scheme (LTD-ANCG). The idea behind these schemes is to divide a light-tree into smaller light-trees to improve the resource utilization. With the new schemes, a multicast request may occupy several LOHTs (logical one hop trees) to serve all its destinations, where the individual LOHTs are interconnected by bridging nodes (referred to as BLOHTs) which receive optical signals from the upstream LOHT,

duplicates them in electronic domain and sends copies to downstream LOHTs. The number of BLOHTs traversed by a multicast request represents the degree of O-E-O processing required to serve it and may also cause additional latency (queuing delay). Apart from improving efficiency, two new schemes also have low O-E-O conversion overheads and provide better overall resource utilization. We assume a dynamic operating scenario where multicast connection requests arrive following a Poisson process. (The main content of this chapter has been published in the author's own papers [123] and [124].)

3.2 Light-tree Configuration

From the resource sharing point of view, small light-trees are better as they can be shared by more multicast connections. However, provisioning for multicast requests will require more O-E-O conversions owing to inter-tree linkages and hence result in more queuing delay if small trees are used. In practice, a tradeoff between resource sharing and O-E-O conversion would be needed. Considering a light-tree that has been configured for servicing a multicast request, the nodes of the light-tree may be categorized as follows.

- *Root Node*: Source of a multicast request. E.g., in Figure 3-1, node 1 is the source of the light-tree.
- *Destination Node*: Destination(s) of a multicast request. E.g., in Figure 3-1, nodes 3, 4, 5 and 6 are destination nodes.
- *Intermediate Node*: a node that forwards optical signal from an input port to one or more output ports (but is not a destination node itself). E.g., in Figure 3-1, node 2 is an intermediate node.
- *Destination Branch Node*: a node that forwards optical signal from an input port to one or more output ports and is also a destination node. Splitter is required to duplicate data

for this node itself. E.g., in Figure 3-1, node 4 is a destination branch node which is also a destination node.

Figure 3-1: A light-tree from source 1 to destinations 3, 4, 5 and 6.

3.2.1 Light-tree Division at Destination Branch Nodes (LTD-DBN)

Figure 3-2: Flow chart of Light-tree Division at Destination Branch Nodes (LTD-DBN) algorithm

Dividing light-trees at destination branch nodes would save on drop ports as optical to electrical conversion is anyway needed at these nodes. Since the traffic grooming function will also be included in destination branch nodes, traffic originating at the nodes can also be groomed and added to the optical signals of the small light-trees. The flow chart of the LTD-DBN algorithm is given in the Figure 3-2 (The pseudocode of the LTD-DBN algorithm is provided in Appendix A).

Figure 3-3: (a) Original light-tree for multicast (1; 3, 4, 5, 6; 0.5C). (b) Two divided trees. (c) New request (1; 3, 4; 0.5C) groomed with multicast (1; 3, 4, 5, 6; 0.5C). (d) New request (2; 5, 6; 0.5C) is groomed with multicast (1; 3, 4, 5, 6; 0.5C)

The benefit of the LTD-DBN approach is illustrated by the example presented in Figure 3-3. We use the notation $(s; d_1, d_2, \dots, d_n; b)$ to represent a multicast of bandwidth b from source s to destinations $d_1, \dots, d_n$. Here the original light-tree of Figure 3-3(a) is for the multicast request (1; 3, 4, 5, 6; 0.5C), where C is the capacity of each of the wavelength channels. In Figure 3-3(b), this is divided into two sub-light-trees at the destination branch node 3. This allows a new request (1; 3, 4; 0.5C) to be accommodated as in Figure 3-3(c). Moreover, by adding a lightpath from node 2 to node 3, a new request (2; 5, 6; 0.5C) is also

simultaneously accommodated as in Figure 3-3(d). This combination of two sub-light-trees and one lightpath supports three multicast requests and allows an overall wavelength utilization of 83.33%. Using LTD-DBN, the size of the light-tree becomes smaller, with a finer grooming granularity and higher resource utilization.

When the original light-tree is divided into sub-light-trees, it becomes possible to use different wavelengths in different sub-light-trees. This makes wavelength assignments also easier. For example, the sub-light-trees of Figure 3-3 may use different wavelengths with the destination branch node 3 connecting the two trees electronically.

3.2.2 Light-tree Division - Adjacent Node Component (LTD-ANC)

For high resource utilization, the wavelength channels, i.e. lightpaths and light-trees, should be intensely shared. However, there is a trade-off between such resource sharing and the amount of O-E-O conversion that would be required in the network. Assigning wavelengths to lightpaths between individual nodes would lead to the most intense wavelength channel sharing in multicast situations but would require O-E-O conversion at a very large number of nodes. This is the approach taken by the LTPG scheme used for comparison later. This has a drawback related to the high degree of O-E-O conversion required.

Figure 3-4: Four categories of adjacent node components (sub-light-trees)

Resource utilization in multicasting can be increased by the improvement of light-tree sharing. This is achieved by dividing a light-tree into smaller sub-light-trees in a way such that O-E-O conversion and electronic routing is kept low. At the logical layer of a network, LOHTs would be selected to groom new requests. The success of grooming is highly related to the number of types of LOHTs. A type of LOHTs is identified by the root and leaves of that LOHT. LOHTs with different roots or different set of leaves belong to different types of LOHTs. With the same traffic load, a logical layer with less number of types would have higher successful grooming probability than other with more types. The reason is that it is more likely that one can successfully select the LOHTs from a small number of types to sustain the new request. So the LTPG scheme has the highest probability of successful grooming as the number of types of LOHT (actually logical link) is the smallest. We achieve this purpose in the proposed LTD-ANC approach by dividing a light-tree into four categories of sub-light-trees called Adjacent Node Components (ANCs). These four categories correspond to four different basic topologies as illustrated in Figure 3-4 and have at most two optical hops. Note that these sub-light-trees are actually the light-trees between adjacent nodes. We can see that the category-3 and category-4 are special cases of category-1 and category-2 with only one leave node separately. When dividing a light-tree into these components, we give category-1 the highest priority while category-4 (which is merely a lightpath between two adjacent nodes) has the lowest priority. The flow chart of the LTD-ANC algorithm is given in Figure 3-5 (The pseudocode of the LTD-ANC algorithm is provided in Appendix B).

Figure 3-6 gives an example where the original light-tree of Figure 3-6(a) is divided into the four categories of ANCs. Following the LTD-ANC algorithm, these ANCs (thick links) are identified step-by-step from Figure 3-6(b) to (f). The final collection of these ANCs is given in Figure 3-6(f). In Figure 3-6(b), start from the source node 1, the algorithm identifies

an ANC from node 1 to nodes 4, 5 and 6, which belongs to the highest priority category (category-1). After this ANC is divided out, three light-trees are left. In Figure 3-6(c), the LTD-ANC algorithm continues to divide the rest. A root of a light-tree is randomly selected, in the figure, node 6 is selected, and the highest priority category of ANC from node 6 is category-2, so a category-2 ANC from node 6 to nodes 8 and 9 is divided out. The algorithm continues to work in Figure 3-6(d), source node 1 is selected at this time, and a category-4 ANC from node 1 to node 3 is divided out. In Figure 3-6(e), a category-3 ANC from node 4 to node 10 is divided out. At last, in Figure 3-6(f), the last ANC from node 8 to node 11 is identified. Totally, there are 5 ANCs, from category-1 to category-4, each has 1, 1, 1 and 2 ANCs, respectively.

Figure 3-5: Flow chart of Light-tree Division - Adjacent Node Component (LTD-ANC) algorithm

Figure 3-6: Light-tree division - adjacent node component approach

Suppose that a network has $|V|$ nodes, and every node has the same node degree M . In theory, for each node, the number of possible LOHTs belonging to four different categories are $M(C_{M-1}^2 + C_{M-1}^3 + \dots + C_{M-1}^{M-1})$, $C_M^2 + C_M^3 + \dots + C_M^M$, $M \cdot C_{M-1}^1$ and C_M^1 , respectively. The total number of possible LOHTs is $|V|(M(2^{M-1} - 1) + 2^M - 1)$, which is much less than $|V|(2^{N-1} - 1)$ of possible LOHTs without splitting the light-trees (every node has $(2^{N-1} - 1)$ LOHTs).

3.3 Multicast Traffic Grooming Algorithms

In this section, we provide the traffic grooming procedures using LTD-DBN and LTD-ANC. We subsequently compare their performance with the SH and MH schemes and the LTPG scheme mentioned earlier.

3.3.1 The LTD-DBN based Grooming Scheme (LTD-DBNG)

Figure 3-7: Flow chart of LTD-DBN based grooming (LTD-DBNG) algorithm

We provide the flow chart of LTD-DBN based grooming (LTD-DBNG) algorithm in Figure 3-7, and flow charts of two related algorithms, Logical Layer Grooming algorithm and Build New Light-trees algorithm, in Figure 3-8 and 3-9, respectively (we also provide the pseudocodes of the above three algorithms in Appendixes C, D and E, respectively). Given the current network which contains resources in the logical and optical layer and the new multicast request, this algorithm finds the route and the resources required if the request is admitted.

Figure 3-8: Flow chart of Logical Layer Grooming algorithm

In LTD-DBNG algorithm, the Logical Layer Grooming Algorithm in Figure 3-8 tries to admit the new request into the existing light-trees by selecting multiple LOHTs, each of which has sufficient free bandwidth and whose set of leaves is a subset of the destination set to be reached. Some new nodes may be added into destinations because this scheme allows a

node which is neither the source nor a destination to be a bridge of two or more LOHTs (BLOHT). The selected LOHTs are saved as the output of the algorithm.

Figure 3-9: Flow chart of Build New Light-trees algorithm

The Build New Light-trees Algorithm in Figure 3-9 is called to construct new light-trees to reach destinations that are not covered by Logical Layer Grooming Algorithm, i.e. those that cannot be reached by the selected existing LOHTs. These new light-trees can root at any node connected to the source of the request by the LOHTs obtained in Logical Layer Grooming Algorithm. In Build New Light-trees Algorithm, it forms a source set that contains nodes connecting to the source by selected LOHTs; then it constructs new light-trees to

accommodate the destination set to be reached. After Build New Light-trees Algorithm, selected LOHTs and the new light-trees connect each other to form a tree topology to accommodate the multicast request.

Then, the new light-trees are divided using the LTD-DBN Algorithm. All light-trees are divided from destination branch nodes to get smaller light-trees. The LTD-DBNG algorithm further improves the resource utilization with SH traffic grooming, because some LOHTs of the logical layer may have the same end nodes as the smaller light-trees. Hence grooming the traffic into existing LOHTs increases resource sharing and saves wavelength resources. Next, wavelength and add/drop port resources to the new light-tree are allocated. After resources are successfully allocated in the optical layer, a corresponding LOHT is added to the logical layer with available bandwidth C , and this LOHT is occupied by connection request. Finally, it allocates the bandwidth for every LOHT of occupy LOHTs in the logical layer which concludes the successful routing of the multicast request in the current network.

Figure 3-10: An example for illustrating the operation of LTD-DBNG

To illustrate the operation of LTD-DBNG, an example is given in Figure 3-10. All existing LOHTs of current network (*CN*) are shown in Figure 3-10(a), and the new request is (6; 1, 2, 8, 9; *b*). Suppose the available bandwidths of all LOHTs in Figure 3-10(a) are larger than *b*. At first, the Logical Layer Grooming Algorithm is called. LOHT (3→8, 9) is selected to groom traffic to destinations 8 and 9 in Figure 3-10(b). LOHT (3→4, 8, 9) will not be selected as node 4 does not belong to the destinations set. Note that, since the root (node 3) of the selected LOHT (3→8, 9) is not the source of the request and is not reached before, it will be added to the destination set of the request, and leaves 8 and 9 will be deleted from the destination set. As shown in Figure 3-10(c), the Logical Layer Grooming Algorithm will also select the other two LOHTs (6→1) and (2→3) (actually two logical links). There is a LOHT from node 5 to node 3, which can reach one destination node, the same as the LOHT from node 2 to 3, but the algorithm selects the LOHT from node 2 to 3 as it would not introduce another node into the destination set. At Figure 3-10(d), a new light-tree (actually a light-path) is setup from node 1 to node 2 to fulfill the request. Note that, the new light-tree can be rooted at the source 6 or node 1, but node 1 is selected, since this requires less resource. Four LOHTs including a newly built are used to accommodate the request, and 3 BLOHT (nodes 1, 2 and 3) are traversed to merge these LOHTs. The request (6; 1, 2, 8, 9; *b*) is successfully accommodated by reserving a bandwidth *b* on each of the four LOHTs.

3.3.2 The LTD-ANC based Grooming Scheme (LTD-ANCG)

The algorithm for the LTD-ANCG scheme is the same as that given earlier for the LTD-DBNG except that the LTD-ANC Algorithm is called instead of the LTD-DBN Algorithm to divide light-trees. In the LTD-ANC Algorithm, adjacent node components are classified into four categories with Category-1 having the highest priority and Category-4 the

lowest priority. This ensures that optical splitting and forwarding are maintained after the light-tree division.

3.3.3 Link Cost in Routing

To derive a lightpath between two nodes, we use Dijkstra's algorithm to find the shortest path between them. (MPH is based on Dijkstra's algorithm.) To calculate the shortest path in the optical layer, we first generate an auxiliary graph and then run Dijkstra's algorithm on it. For this, we use the link cost p_{ij} of link (i, j) to be 1 if link (i, j) exists in the optical layer and has free wavelengths; otherwise, we use $p_{ij} = \infty$. With this setting, the shortest path derived is the path with the least optical hop count between two nodes.

To derive the shortest paths in the logical layer, which are used to construct a logical tree for a multicast request, we first generate an auxiliary graph with logical links with sufficient available bandwidth for the new request and then calculate shortest paths using Dijkstra's algorithm. Let l_{ij} be the cost of link (i, j) of the auxiliary graph. For simplicity, we set $l_{ij} = 1$ if a logical link with sufficient available bandwidth exists between i and j ; otherwise set $l_{ij} = \infty$ (i.e., i and j are not directly connected by a logical link) in the auxiliary graph. With this setting, the shortest path derived is the path with the least logical hop count between two nodes.

When a multicast request terminates, the resources allocated to the request will be released. The bandwidth would be returned to every LOHT occupied by the request. Once this is done, if the free bandwidth of a LOHT equals C (i.e. capacity of a wavelength), then we tear down the light-tree in the optical layer which maps to the LOHT in the logical layer and release the wavelength of optical links of the light-tree. We then delete the LOHT in the logical layer,

and the light-tree in the optical layer, because there is no traffic being transmitted in the light-tree.

3.4 Performance Evaluation

Extensive simulations were conducted to study the performance of the proposed schemes, LTD-DBNG and LTD-ANCG, and compare them with SH, MH and LTPG. The comparison was carried out in terms of the blocking probability (BP), the average number of LOHTs per admitted request (Avg. no. of LOHTs) and the average number of bridges of LOHTs per admitted request (Avg. no. of BLOHTs). These are defined as:

$$BP = \frac{\text{No. of blocked requests}}{\text{No. of total requests}} \quad (3.1)$$

$$\text{Avg. no. of LOHTs} = \frac{\sum_{i=1}^{TR-BR} LOHT_i}{\text{No. of total requests} - \text{No. of blocked requests}} \quad (3.2)$$

$$\text{Avg. no. of BLOHTs} = \frac{\sum_{i=1}^{TR-BR} BLOHT_i}{\text{No. of total requests} - \text{No. of blocked requests}} \quad (3.3)$$

3.4.1 Test Networks and Traffic Model

We evaluate the performance of the new schemes using two network topologies, NSFNET and COST239, shown in Figure 3-11 and 3-12. The NSFNET has 14 nodes and 21 links, with an average node degree of 3. Note that the node degree of a node is defined as the number of edges entering/leaving that node. The average node degree of a network is equal to the number of edges in the network divided by the number of nodes in the same network. The COST239 has 11 nodes and 26 links, with an average node degree of 4.727. The number of

wavelengths per fiber link, W , is set to be 64 in NSFNET, and 32 in COST239. The following assumptions are relevant to either the NSFNET or the COST239 networks. There are no wavelength converters, so all light-trees and lightpaths must satisfy the wavelength continuity constraint.

Figure 3-11: NSFNET topology

Figure 3-12: COST239 topology

We assume that every node is able to generate multicast requests. Multicast request arrivals are assumed to follow a Poisson process with rate λ and their holding times are assumed to have a negative exponential distribution with a mean $1/\mu$. For simplicity the bandwidth required by the requests is assumed to be uniformly distributed between $(0, C]$, where C is the bandwidth of a wavelength channel. A request is rejected immediately if the network cannot support it. The source of a multicast request is randomly chosen from the network nodes. The

number of multicast destinations, denoted by the random variable X , is assumed to have a truncated geometric distribution [125], with parameter q ($0 < q < 1$). The probability of a multicast request having k destinations and the mean number of the destinations are given as:

$$P(X = k) = \frac{(1-q)q^{k-1}}{q - q^N}, \quad 2 \leq k \leq N \quad (3.4)$$

$$E(X) = \sum_{k=2}^N k \cdot P(X = k) = \frac{2q - q^2 - (N+1)q^N + Nq^{N+1}}{(1-q)(q - q^N)} \quad (3.5)$$

where N is the maximal number of the destinations, $N = |V| - 1$.

3.4.2 Blocking Probability Comparison

Figures 3-13 and 3-14 compare the blocking probability (BP) of five schemes, SH, MH, LTPG, LTD-DBNG and LTD-ANCG in NSFNET and COST239, where the mean values of the multicast destination size are 5 and 4, respectively. As seen from the curves, BP increases with the traffic load, as expected. It is demonstrated that the LTPG gives the lowest BP, followed by LTD-ANCG, LTD-DBNG, MH and SH in an ascending order. As shown in Figures 3-13 and 3-14, the BPs of SH and MH are almost identical (overlapped) and are much higher than that of other schemes. The reason for this is that the probability of finding requests with the same destinations and that of finding a request with the same destinations and the same source are both very low. SH and MH perform the worst because a large portion of bandwidth resources are wasted since very few requests are able to share light-trees; this also leads to the blocking of requests that arrive later. We give an example to show why SH and MH has low resource utilization and exhibit very similar performance. Consider a network with $|V|$ nodes, and the destinations size of multicast requests is T . If we randomly generate two multicast requests, then the probability of two requests having the same source

and destinations is $1/(|V| \cdot C_{|V|-1}^T)^2$ (for SH), and the probability of two requests having the same destinations is $1/(C_{|V|}^T)^2$ (for MH). If $|V|=14$ and $T=5$, the probability of two requests that can share a light-tree is 3.08×10^{-9} for SH, and 6.04×10^{-7} for MH. Both of them are negligible.

Figure 3-13: Blocking probability comparison in NSFNET, $E(X)=5$

Figure 3-14: Blocking probability comparison in COST239, $E(X)=4$

Compared with SH and MH, LTD-DBNG and LTD-ANCG significantly improve the BP performance. This agrees with our expectation: by dividing light-trees into small ones, resource utilization can be improved and BP can be reduced. Specifically, as shown in Figure 3-13, with LTD-DBNG, the BP is reduced by a range of about 2.8 orders of magnitude (at traffic load of 100 erlangs) to about 0.8 orders of magnitude (at traffic load of 150 erlangs), while with LTD-ANCG, the BP is reduced by a range of about 5.6 orders of magnitude to about 1.9 orders of magnitude. LTD-ANCG performs better than LTD-DBNG, which confirms our prediction that decreasing the number of LOHT types can improve resource utilization and hence reduce BP. LTD-ANCG performs close to LTD-DBNG in the COST239 network, as shown in Figure 3-14. The reason may be that the COST239 network has a higher node degree than the NSFNET, and a high node degree leads to a larger number of LOHT types, resulting in lower resource utilization.

In both Figure 3-13 and Figure 3-14, LTPG has the lowest BP. This can be explained as follows: Traffic grooming among lightpaths can achieve higher resource utilization than that among light-trees, even with small sub-light-trees. A lightpath is an extreme case of a light-tree with only one destination. The probability of finding a lightpath to groom requests of the same destination is much higher than that of finding a light-tree to groom requests of the same set of destinations.

We also compare the BP of the five schemes with different mean values of the destination size in Figure 3-15. The test network is NSFNET, and the traffic load is set to 140 erlangs. As shown in Figure 3-15, LTPG and LTD-ANCG have a cross-over point at a low mean value of the destination size (about 4). Below this point, the BP of LTD-ANCG is lower than that of LTPG. This is because with fewer destinations, a light-tree without many branches is more like a light-path, and LTD-ANCG would divide the light-path into short light-paths, which will increase the resource utilization.

Figure 3-15: Blocking probability comparisons with different mean values of the multicast destination size (traffic load fixed at 140 erlangs in NSFNET)

It is noted that, although LTPG has a lower BP than LTD-ANCG and LTD-DBNG when the multicast destination size increases from 4 to 7, as shown in Figure 3-13, 3-14 and 3-15, LTPG involves a large number of O-E-O conversions and electronic processing, which would make LTPG method impractical in applying to large networks. However, LTD-ANCG achieves a good tradeoff between the BP and the degree of O-E-O conversion and electronic processing as compared to SH, MH and LTPG schemes. LTD-ANCG has a much lower BP than SH or MH, and similar blocking performance of LTPG, meanwhile LTD-ANCG requires significant less O-E-O conversion and electronic processing than LTPG. We next compare the average number of LOHTs and BLOHTs among different grooming algorithm.

3.4.3 Comparison of Average Number of LOHTs and of BLOHTs Among Different Grooming Algorithms

Though the BP of LTPG is the lowest, it may not be a practical grooming scheme. In LTPG, logical trees are constructed only with lightpaths, instead of light-trees, and the number of

lightpaths needed for a multicast request is larger than that of light-trees as a lightpath can reach fewer nodes than a light-tree does. Figures 3-16 and 3-17 show the average number of LOHTs occupied by a multicast request for the schemes studied (except SH as exactly one LOHT is used in SH). In both test networks, LTPG has the largest Avg. no. of LOHTs per multicast request and MH has the lowest value. Both LTD-DBNG and LTD-ANCG have a smaller average number of LOHTs to support a request as sub-light-trees can reach multiple nodes simultaneously.

Figure 3-16: Comparison of the Avg. no. of LOHTs per multicast request in NSFNET, $E(X)=5$

As shown in Figure 3-16, in NSFNET, the average number of LOHTs of LTD-ANCG is slightly higher than that of LTD-DBNG since sub-light-trees of LTD-ANCG (i.e. light-trees among adjacent nodes) are smaller than that of LTD-DBNG. This is because in a network with a small node degree such as NSFNET, sub-light-trees obtained by LTD-ANC approach would reach fewer nodes and hence a request would occupy more sub-light-trees. For the case of COST239 presented in Figure 3-17, the Avg. no. of LOHTs of LTD-ANCG is almost

the same as that of LTD-DBNG. This is because the COST239 has a higher node degree and the sub-light-trees obtained by LTD-ANC approach become larger.

Figure 3-17: Comparison of the Avg. no. of LOHTs per multicast request in COST239, $E(X)=4$

Figure 3-18: Comparison of the Avg. no. of BLOHTs in NSFNET, $E(X)=5$

Figure 3-19: Comparison of the Avg. no. of BLOHTs in COST239, $E(X)=4$

Figures 3-18 and 3-19 show the average number of BLOHTs needed by a multicast request for different schemes. The average number of LOHTs for MH is about one, and hence its average number of BLOHTs is nearly zero, so it is not included in the figures. The average number of LOHTs for SH is exactly one and hence it is also not included in the figures. Since a bridge node connects multiple LOHTs by O-E-O conversion, so the average number of BLOHTs per multicast request represents the degree of O-E-O conversion and electronic forwarding in the network (i.e, electronic processing overhead).

As shown in Figures 3-18 and 3-19, LTPG has the largest average number of BLOHTs and hence requires more O-E-O conversion and electronic forwarding. Our new schemes, LTD-ANCG and LTD-DBNG, have smaller values in both test networks. Specifically, in the NSFNET (Figure 3-18), the average number of BLOHTs is about 1.93 for LTD-DBNG, and is about 2.03 for LTD-ANCG. However, LTPG has the largest average number of BLOHTs of about 2.3. So LTD-DBNG reduces about 16% for the average number of BLOHTs per multicast request, which also means a reduction of about 16% O-E-O conversion and electronic forwarding in the network, while LTD-ANCG reduces about 12% of O-E-O

conversion and electronic forwarding. In the COST239 (Figure 3-19), LTD-ANCG achieves the lowest average number of BLOHTs per multicast request. This is because the COST239 has a higher node degree, sub-light-trees obtained by LTD-ANC become larger and also a bridge of LOHTs in a denser network may connect more LOHTs. With a small average number of BLOHTs, more traffic would bypasses electronic routers, staying in optical domain, which can significantly lower the burden of electronic processing [15].

Usually, minimizing blocking probability and minimizing O-E-O processing conversion of a network are two conflicting objectives in the network design, and there does not exist a single solution that simultaneously optimizes both objectives. To achieve a low blocking probability, the O-E-O processing objective has to be compromised to some extent. On the other hand, to achieve a low O-E-O processing overhead, the blocking probability objective has to be impaired to some extent [126]. This problem is called a multi-objective optimization problem. One way to verify if a solution is better than another or not, is to convert the multi-objective optimization problem to a single objective optimization problem, where a weighted linear combination of two objectives is usually formed as the objective function, i.e. $a \times BP + b \times \text{avg. no. of BLOHTs}$ (we use the value of avg. no. of BLOHTs to denote the O-E-O processing), where a and b are the weights of two respective objectives and they are decided by the preference information given by decision makers, and they may also vary with different topologies, traffic loads, revenue expectations or marketing strategies. In the study, we assume $a=10$ and $b=1$ for NSFNET topology in the considered traffic loads to illustrate the benefit of our proposed algorithms over the LTPG algorithm. In Figure 3-20, the single objective function values of three algorithms are plotted. It is clear that the two proposed algorithms have lower values (i.e., better performances) than the LTPG algorithm. Specifically, LTD-ANCG and LTPG have stable values with respect to different traffic loads, while LTD-DBNG have very different objective values for different traffic loads because it

has a high blocking probability as traffic load increases (see Figure 3-13). It is also noted that at low traffic load (under 130 erlang), LTD-DBNG has a lower value than LTD-ANCG. This is because of the larger contribution of avg. no. of BLOHT of LTD-DBNG to the objective value, which makes the objective value lower than the others.

Figure 3-20: Comparison of the single objective function values in NSFNET, $E(X)=5$

From the simulations, we can observe that, our new schemes can achieve a good tradeoff between the BP and the degree of O-E-O conversion and electronic processing as compared to SH, MH and LTPG schemes. Especially, LTD-ANCG has the similar blocking performance of LTPG, and reduced O-E-O conversion and electronic processing in a moderate node degree network such as NSFNET.

3.4.4 Effect of Add/drop Ratios on BP Performance

For grooming low bandwidth connections into lightpaths or light-trees, optical signals have to be converted into electrical signals. After being groomed, they are converted back to optical

signals and sent out. Optical add/drop ports are needed at each network node, which consists of an OXC and an electronic router, to facilitate traffic grooming. An add port is used at the source node of a lightpath or a light-tree, and a drop port is needed at the destination node of a lightpath, or each destination node of a light-tree. Earlier, we assumed that add/drop ports are always available. However, if the number of add/drop ports is limited, it may not be possible to set a new lightpath or light-tree even if there are spare wavelengths. As in [127], it is important to investigate the impact of the add/drop ratio on the blocking performance. The add/drop ratio (ADR) is defined as the number of add/drop ports per fiber to the number of wavelengths per fiber [127].

Figures 3-21 and 3-22 show the BP versus the ADR for NSFNET and COST239, respectively. Here the traffic load is fixed at 130 erlangs in NSFNET and 140 erlangs in COST239. The ADR is varied from 10% to 100% (100% means the number of add/drop ports equals to that of input/output wavelength ports). As shown in the figures, when the traffic load is low, as the ADR increases, the blocking probability of each scheme decreases rapidly. However, once the ADR reaches a certain threshold value, the increase of the ADR does not reduce the blocking probability, remaining almost unchanged regardless of the increase in ADR. As shown, the threshold value is approximately 0.4 for all schemes in NSFNET. In COST239, the threshold value is about 0.5 for LTPG, and 0.4 for other schemes. Our simulation results also demonstrate that when the number of wavelengths per fiber or traffic load is changed, similar conclusions are observed, i.e., a threshold value exists for each scheme. The existence of a threshold value is also reported in the case of unicast grooming [127, 128], which can be explained by the same reason that only a fraction of traffic is terminated and dropped at each OXC and the rest of traffic bypasses the electronic processing. The threshold value for the ADR can be used to determine the optimal number of add/drop ports needed at an OXC to get almost the same blocking performance as the case with the

100% ADR without paying unnecessary network costs. A smaller ADR means that one can use fewer add/drop ports, leading to lower cost of OXCs and electronic routers.

Figure 3-21: BP Comparison by varying ADR in NSFNET, when the traffic load is fixed and $E(X)=5$

Figure 3-22: BP Comparison by varying ADR in COST239, when the traffic load is fixed and $E(X)=4$

Because of the similarity between SH and MH, these two algorithms have almost the same BP when varying the ADR. LTPG has the best blocking performance, followed by LTD-ANCG, LTD-DBNG, MH and SH in an ascending order. This result is consistent with the blocking performance shown in Figures 3-13 and 3-14 without the ADR constraint.

3.5 Summary

We have considered the problem of multicast traffic grooming in WDM mesh networks. Observing that the division of a light-tree into smaller sub-light-trees would improve resource utilization, we have proposed two new grooming schemes, namely Light-Tree Division - Destination Branch Node based Grooming scheme (LTD-DBNG) and Light-Tree Division - Adjacent Node Component based Grooming scheme (LTD-ANCG). Apart from being more efficient than other algorithms available in current literature, our proposed schemes also have low O-E-O conversion and electronic forwarding overheads and provide better overall resource utilization. Extensive simulation experiments have been carried out to study the blocking performance, the average number of logical one hop trees and the average number of bridge nodes occupied by a multicast connection. The results have revealed that compared to the existing scheme, the proposed new schemes can strike a good tradeoff between the blocking performance and the electronic processing overheads. LTD-ANCG performs better than LTD-DBNG. We have also investigated the blocking performance of these schemes with variations in the add/drop ratio and shown that a proper choice of this ratio will provide optimal blocking performance with low network costs.

Chapter 4

Multicast Traffic Grooming with Leaking Strategy with Dynamic Traffic

4.1 Introduction

Since the ultimate goal of optical network design is to minimize cost, it is important to realize that the more significant CAPEX cost is that of the higher layer electronic components (e.g., Add/Drop Multiplexers (ADMs), or IP routers) rather than the cost of the wavelengths [46, 67-69, 111, 129]. For example, an OC-48 SONET ADM costs about \$25,000; for optical devices which enable wavelength transmission, a 2.5 GHz laser costs about \$175, and a detector costs about \$75, so a wavelength would cost about \$250/fiber [111], which is much lower than that of an OC-48 SONET ADM. Moreover, the ever-increasing number of wavelengths that can be simultaneously transmitted in a fiber tends to decrease the cost per wavelength over time. It should also be noted that any higher layer electronic component will have only a limited number of electronic ports. Therefore significant CAPEX savings on electronic components can be made by minimizing the number of ports used. This is especially important in the case of light-tree based WDM networks because every light-tree uses multiple ports, i.e. an add port and several drop ports. For general WDM networks,

considerable research [15, 46, 67-69] has been reported on reducing the usage of add/drop ports or increasing their utilizations so that more connection requests can be efficiently accommodated in the network. These general principles also apply here, i.e., increasing the utilization of light-trees reduces the usage of electronic ports and lowers the overall cost.

Figure 4-1: Multicast traffic grooming with leaking

In Chapter 3, we proposed a multicast traffic grooming algorithm *Light-Tree Division - Adjacent Node Component based Grooming algorithm* (LTD-ANCG), which can increase the light-tree sharing (add/drop ports utilization) by dividing light-trees into adjacent node components (ANC). These are small light-trees within two optical hops, and traffic is groomed to those ANCs whose destination sets are subsets of the destinations to be reached. Although this strategy increases the sharing of light-trees, applying the strict condition that the destinations of the ANC must be a subset of the destination nodes to be reached limits the extent of sharing when the add/drop port resource is limited. Similarly as above, a multicast connection request is denoted as $r(s; d_1, d_2, \dots, d_n; b)$, where s is the source, d_1 to d_n are the destinations, and b is the required bandwidth. A light-tree is denoted as a LOHT $t(h \rightarrow d_1, d_2, \dots, d_m)$ where h is the root, and d_1 to d_m are the destinations. As an example, consider the scenario of Figure 4-1 where every node has only one transceiver (one add port and one drop port) and where two multicast connection requests, $r_1(1; 3, 4; 0.3)$ and $r_2(3; 5, 6; 0.3)$, with bandwidth requirements of 0.3 each are already in the network. Two light-trees $t_1(1 \rightarrow 3, 4)$

and $t_2(3 \rightarrow 5,6)$ are used to accommodate the two connection requests r_1 and r_2 , respectively. Assuming the (normalized) bandwidth of a wavelength to be 1, this leaves a bandwidth of 0.7 on each tree to accommodate other connection requests, e.g., a new request r_3 with a bandwidth requirement of 0.2. According to the LTD-ANCG algorithm, the existing light-trees $t_1(1 \rightarrow 3,4)$ is selected to groom the traffic as the destination set $\{3,4\}$ of the light-trees is a subset of the connection request destinations $\{3,4,5\}$. The light-trees $t_2(3 \rightarrow 5,6)$ is not selected to groom traffic r_3 as node 6 is not in the destinations of r_3 . The LTD-ANCG algorithm then tries to build a new light-tree (which in this case, is actually a lightpath) from node 4 or node 3 to node 5 since this is the only remaining destination. However, this cannot be done as there are no drop ports left at node 5, since the only one available has already been used by $t_2(3 \rightarrow 5,6)$. This blocks the new connection request r_3 in the LTD-ANCG algorithm. However, if traffic grooming with leaking is allowed, after $t_1(1 \rightarrow 3,4)$ is selected to reach node 3 and node 4, $t_2(3 \rightarrow 5,6)$ is selected to transmit the traffic of r_3 to node 5 even through this will cause traffic to leak to node 6, i.e., node 6 then discards the traffic of r_3 . This example illustrates that multicast traffic grooming with leaking may increase the sharing of light-trees at the cost of leaking traffic to unintended nodes, i.e., sharing $t_2(3 \rightarrow 5,6)$ with leaking enables to accommodate the connection request r_3 .

Apart from improving the utilization of light-trees, traffic grooming with leaking may also increase the probability that a connection request is successfully accommodated. Since a multicast session is formed by multiple small light-trees, if there is traffic leaking to unrelated nodes, then starting from these nodes, we may be able to find optimal light-trees to reach other destinations. This results in more light-trees being sourced from the leaves of the grooming trees than when traffic grooming without leaking is implemented.

In this chapter, we investigate multicast traffic grooming with this leaking strategy in dynamic traffic scenarios. When the add/drop port resource is limited, this leaky grooming

strategy has the potential to improve the sharing of light-trees. We propose two multicast traffic grooming algorithms with the leaking strategy: *Multicast Traffic Leaky Grooming* algorithm (MTLG) and *Multicast Traffic Hybrid Grooming* algorithm (MTHG). MTLG grooms traffic to light-trees only if the leaked traffic is less than a pre-specified threshold. MTHG first grooms traffic to light-trees without leaking; if this fails to reach some destinations, then it further grooms the traffic to light-trees with leaking. We also consider the routing problem of multicast connections, where multiple light-trees form a larger tree session to accommodate a multicast connection request. A new routing algorithm, *Constrained Light-tree Multicast Routing* (CLMR), is also proposed which consumes fewer add/drop ports and exhibits better performance. (The main content of this chapter has been published in the author's own paper [130].)

4.2 Routing with Add/drop Port Limitation

Since the light-tree based LTD-ANCG algorithm in Chapter 3 has been demonstrated to perform the best among the existing light-tree based multicast traffic grooming algorithms, we adopt similar ideas in the new traffic grooming algorithm with leaking. In particular, we use ANCs to construct larger multicast sessions and encourage sharing of a small light-tree between several multicast connections.

Before describing our new grooming algorithms, we investigate first the routing problem of small light-trees (ANCs) as that directly affects the performance of our grooming algorithms. The *Light-Tree Division - Adjacent Node Component* (LTD-ANC) algorithm was proposed to divide a light-tree into adjacent node components which are small light-trees within two optical hops. The grooming algorithm LTD-ANCG uses the LTD-ANC algorithm to divide new light-trees built for the remaining nodes which could not be reached by that

grooming procedure. Four kinds of adjacent node components in Figure 3-2 are categorized into two kinds as 1-hop light-trees and 2-hop branches in Figure 4-2. Here optical splitting is done at the roots of the 1-hop light-trees and the intermediate nodes of the 2-hop branches. With these two kinds of light-trees, a larger tree session can be formed by O-E-O conversion at the connection nodes so that the smaller light-trees can be interconnected.

Figure 4-2: Adjacent node component (ANC). (a) 1-hop light-trees (b) 2-hop branches

Existing multicast routing algorithms (including the MPH used for LTD-ANC) usually try to build light-trees with the minimal total link cost (or minimal number of links) to accommodate a multicast connection request. However, when dividing a light-tree into small components, add/drop ports need to be introduced at the dividing nodes to connect the components. The routing algorithm builds a light-tree with a minimal number of wavelinks (a wavelink is a wavelength channel in a particular link it traverses, here the number of wavelinks of a light-tree is equal to the number of physical links the light-tree traverses) then divides it into smaller ones, which may substantially increase the usage of add/drop ports. For networks where add/drop ports are a scarce resource, new algorithms need to be developed to reduce the usage of add/drop ports rather than reduce the usage of wavelinks. We propose next a new *Constrained Light-tree Multicast Routing* (CLMR) algorithm to efficiently perform multicast routing in networks where the add/drop port resource is limited.

4.2.1 Constrained Light-tree Multicast Routing (CLMR) Algorithm

Using multiple small light-trees to construct a large tree session for a multicast connection can improve the sharing of light-trees in multicast grooming scenarios, as small light-trees can be easily shared by multiple connections. We have observed that connecting multiple small light-trees at a node to form a bigger tree consumes one drop port and multiple (at least one) add ports. The idea of the CLMR algorithm is to build small light-trees (ANCs) between the source and the destinations in a way that reduces the number of connecting nodes to reduce the using of add/drop ports, i.e., nodes other than the source and destination nodes. It may also be noted that connecting light-trees at a destination node of a light-tree can reduce one drop port usage since such a port is already used there anyway. The flow chart of the CLMR algorithm is given in the Figure 4-3 (The pseudocode of the CLMR algorithm is given in the Appendix F).

The algorithm consists of three parts: 2-hop branch building, 1-hop light-tree building and extension path building. These three parts are included in the loop, so every time after an ANC has been built, the flow starts the loop once again. The 2-hop branch building is positioned at the first, so that only when no further 2-hop branches are available, does the algorithm go to the 1-hop light-tree building to derive 1-hop light-trees. When neither the 2-hop branch nor 1-hop light-tree are available, the algorithm goes to the extension path building. The reason for introducing the extension path building in the algorithm is that using 2-hop branches and 1-hop light-trees to construct a large tree session to accommodate connections may fail as some destination nodes may be very far from all the other nodes (i.e. more than 2 hops away). In the extension path building, a shortest path to the closest destination node is built. Then from the destination of the path, a new ANC may be derived. The CLMR algorithm tries to build ANCs between the source node and the destinations in such a way that it can take advantage of the drop ports already at the destination nodes.

Figure 4-3: Flow chart of Constrained Light-tree Multicast Routing (CLMR) algorithm

In the 2-hop branch building, a 2-hop branch is selected after $O(|V||D|^2)$ computations, where D is the destination set. This is because starting from a source node of source set, it checks every node of D as the first hop and then check all nodes in D as the second hop (actually, only adjacent nodes are checked) to select the branch with the maximal number of level nodes (subset of destination set to be reached). In the 1-hop light-tree building, the complexity is $O(|V||D|)$ as every node in D is checked once to select the light-tree with the maximal number of level nodes when a source node is fixed. The process of constructing the extension path has complexity of $O(|V|^2|D|)$. This shows that the complexity of the CLMR algorithm is $O(|V|^2|D|^2)$.

4.2.2 Optimal Routing

We use ILP formulation to model the optimal routing for a multicast connection request. A multicast connection request $(s; D)$ is given, and the ILP formulation is to find the optimal routing of ANC's for the request. It is noted that the LOHT destination set J_i in this ILP formulation contains all ANC's from node i and also every unicast destination with more than two hops from node i . Please refer to Table 2-1 for the meanings of notations and variables used in this section.

$$\text{Minimize:} \quad \sum_i \sum_{J \in J_i} \lambda_{iJ} + \sum_i \sum_{J \in J_i} \sum_{j \in J} Q_j^{iJ} \quad (4.1)$$

The objective function is to minimize the number of higher layer electronic ports (add ports and drop ports) to accommodate the multicast connection request $(s; D)$. The first part of the objective function is the number of add ports used, and the second part is the number of drop ports used. The set of constraints is given below.

$$Q_j^{iJ} = \lambda_{iJ} \quad \forall i, J \in J_i, j \in J \quad (4.2)$$

$$\sum_{J \in J_s} \lambda_{sJ} \geq 1 \quad (4.3)$$

$$\sum_i \sum_{J \in J_i} Q_j^{iJ} = 1 \quad \forall j \in D, j \in J \quad (4.4)$$

$$\sum_{J \in J_q} \lambda_{qJ} \leq |J_q| \cdot \sum_i \sum_{J \in J_i} Q_q^{iJ} \quad \forall q \neq s \quad (4.5)$$

$$H_n \geq H_m + 1 - (1 - \sum_{J \in J_m} Q_n^{mJ}) \cdot |V| \quad \forall n, \forall m \neq n \quad (4.6)$$

$$\sum_m F_{mi}^{iJ} = 0 \quad \forall i, J \in J_i \quad (4.7)$$

$$\sum_n F_{in}^{iJ} = |J| \cdot \lambda_{iJ} \quad \forall i, J \in J_i \quad (4.8)$$

$$\sum_m F_{mj}^{iJ} - \sum_n F_{jn}^{iJ} = Q_j^{iJ} \quad \forall i, J \in J_i, j \in J \quad (4.9)$$

Equation (4.2) is to set the value of Q_j^{ij} to be 1 if LOHT ($i \rightarrow J$) is used by the connection request and node j belongs to the destination set of the LOHT. Equation (4.3) ensures that the number of outgoing streams from the source node s is no less than one. Equation (4.4) constrains the incoming stream of each destination to be 1. Equation (4.5) ensures that except the source node, the root of a LOHT that supports the connection request must be a destination of another LOHT which also supports the request. Equation (4.6) ensures that, if node m and n are in the same LOHT that supports the connection request, the delay from the source s to node n is larger than that to node m , where m is the root of the LOHT and n belongs to the destinations of the LOHT.

Equations (4.7)-(4.9) are to constrain the routing of LOHTs used by the connection request. Equation (4.7) ensures that there is no stream flowing into the source of LOHT. Equation (4.8) ensures that if a LOHT is used by the connection request, the number of streams flowing out of the source equals the number of the LOHT destinations. Equation (4.9) ensures that the outgoing stream is one less than the incoming stream for the destination node of LOHT, and that for other intermediate nodes, which are neither a source nor one of the destinations, the incoming and outgoing streams are the same.

4.2.3 Simulation Results

We compare the performances of CLMR, LTD-ANC and the optimal results derived by the ILP for the NSFNET network with 14 nodes. We use a commercial ILP solver, “CPLEX” [131] to solve the mathematical formulations. The source of the connection request is randomly chosen from the network nodes, and the destination nodes are randomly selected from among the network nodes (excluding the source node). In each simulation experiment, one multicast connection request is generated as the input. We here compare the performance

of the algorithms in terms the numbers of electronic ports (add ports and drop ports) and wavelinks required by a multicast connection. The results are shown in Figures 4-4 and 4-5 where the values at each destination size are the average value of 100 simulation runs.

As shown in Figure 4-4 for all algorithms, when the destination size increases, more electronic ports are consumed as more destinations make multicast sessions bigger. We can see that the optimal ILP approach consumes the least number of electronic ports. Our proposed algorithm CLMR uses fewer electronic ports than LTD-ANC. It may be noted that CLMR uses almost the same number of electronic ports as the optimal ILP approach and that is about 9% less than what is needed for LTD-ANC averagely. The numbers of wavelinks used by these schemes are compared in Figure 4-5. As shown, more wavelinks are used when destination size increases as the larger multicast sessions. The optimal approach has the lowest value, followed by LTD-ANC and then CLMR. In summary, we find that our proposed CLMR algorithm uses almost as many electronic ports as the optimal ILP approach but performs better than LTD-ANC.

Figure 4-4: Comparison of number of electronic ports required by a multicast connection for different routing algorithms

Figure 4-5: Comparison of number of wavelinks required by a multicast connection for different routing algorithms

4.3 Multicast Traffic Grooming Algorithms with Leaking Strategy

Traffic grooming with leaking can improve the utilization of add/drop ports at the expense of traffic leaking to unrelated nodes. This may cause some other undesirable side effects. For example, if too much traffic is leaked then the utilization of add/drop ports may degrade as resources are wasted. This implies that traffic leaking and resource sharing should be balanced and its impact on blocking performance should be examined. Consider a multicast connection request $(s; D)$, where s is the source and D is the set of destinations. If a LOHT $(r \rightarrow L)$, where r is the root and L is the destination set of the LOHT, is selected to groom the traffic, then the leaking ratio is defined as

$$\text{leaking ratio} = \frac{|L| - |D'|}{|L|} \quad (4.10)$$

where $D' = D \cap L$. If this value is 0, then there is no leaking and the grooming strategy is the same as that for LTD-ANCG. For our proposed grooming algorithms, if the traffic leaking of a selected LOHT is below a given leaking ratio threshold then the grooming operation for

that LOHT is considered acceptable. We here propose two multicast traffic grooming algorithms with the leaking strategy, i.e., *Multicast Traffic Leaky Grooming* (MTLG) and *Multicast Traffic Hybrid Grooming* (MTHG) algorithms. The flow chart of the MTLG algorithm is given in Figure 4-6 (The pseudocode of the MTLG algorithm is given in the Appendix G).

Figure 4-6: Flow chart of Multicast Traffic Leaky Grooming (MTLG) algorithm

In the MTLG algorithm, the loop is to groom the traffic to those existing LOHTs which have maximal intersection to the destination set to be reached while ensuring that the traffic leaking ratio is lower than the leaking ratio threshold. After the leaky grooming, if there are still destinations unreached, new light-trees are derived to support them. If required

bandwidth of the request is smaller than the capacity of a wavelength, the CLMR algorithm is called to build light-trees, then physical resources are allocated, and finally the bandwidth of routing is allocated. Note that the connection requests that require a full wavelength bandwidth are not groomed. Instead, a new light-tree with the minimal total link cost is derived by the MPH algorithm, as this consumes the least add/drop ports and wavelinks.

Figure 4-7: Flow chart of Multicast Traffic Hybrid Grooming (MTHG) algorithm

The MTHG algorithm is similar to the MTLG algorithm, except that MTHG grooms traffic to LOHTs without leaking first. If some destinations remain to be reached, it then grooms traffic to LOHTs with leaking in the same manner as in the MTLG algorithm. Therefore, MTHG is an improvement over MTLG as it is able to support more resource sharing with less leaking traffic. The flow chart of MTHG is given in Figure 4-7 (The pseudocode of MTHG is given in the Appendix H).

In the first loop of the MTHG algorithm, a LOHT with the maximal subset of the unreached destination set is selected. Since there is no traffic leaking at this stage, the destinations of the selected LOHT must be a subset of the destination set to be reached. If some destinations remain to be reached, MTHG grooms traffic with leaking in the same manner as the MTLG algorithm. The first loop of MTHG is to select existing LOHTs to groom traffic without leaking, which checks all existing LOHTs in the current network. This has complexity of $O(|V|2^{lg^2})$, where g is the maximal node degree of the network (the complexity is from the checking of every node and each node with 2^{lg^2} LOHTs). Therefore, the first loop has complexity of $O(|V|^2 2^{lg^2})$. It is clear that the grooming with leaking in the second loop is also with the complexity $O(|V|^2 2^{lg^2})$. Building a new light-tree has complexity $O(|V|^2 \log |V|)$, while using CLMR to build light-trees has complexity $O(|V|^2 |D|^2)$. Therefore the MTHG algorithm has overall complexity of $O(|V|^2 2^{lg^2} + \max(|V|^2 \log |V|, |V|^2 |D|^2))$. The MTLG algorithm has the same complexity.

4.4 Simulation Results

In this section, we present simulation results for the MTLG, MTHG and LTD-ANCG algorithms for the NSFNET topology with the number of wavelengths, W , set to 32.

Multicast connection request arrivals are assumed to follow a Poisson process with rate λ and holding times are negatively exponentially distributed with a mean $1/\mu$. Unless otherwise specified, the bandwidth required by connection requests is assumed to be uniformly distributed between $(0, C]$, where C is the bandwidth of a wavelength channel. The source of a multicast connection request is randomly chosen from the network nodes. The multicast destination size is also randomly chosen. In the present case the choice for the destination size is between 1 and 13, and destination nodes are randomly selected from network nodes (excluding the source node). A connection request is immediately rejected if the network cannot accommodate it. A transmitter and a receiver are usually built together as a transceiver. Since a transmitter is connected to an add port for transmitting optical signals and a receiver is connected to a drop port for receiving optical signals, we assume that the number of add ports and drop ports are equal at a node. From cost considerations, add/drop ports may be sparse in a network, and the add/drop ratio is defined as the ratio of the number of add/drop ports per fiber to the number of wavelengths per fiber [127].

In Figure 4-8, blocking ratios are compared when network load is 50 erlangs and the add/drop ratio is set to 0.3. The error bars in the figure indicate the 95% confidence interval of the blocking ratios. When the leaking ratio threshold is varied from 0 to 0.9, the MTHG algorithm achieves the best performance, followed by the MTLG algorithm. The LTD-ANCG has the highest blocking ratio. The MTHG performs better than the MTLG, as MTHG grooms traffic without leaking in the first phase and then grooms traffic in the same way as MTLG in the second phase, resulting in less leakage (i.e. less wastage of resources). If leaking is not allowed (the leaking ratio threshold is 0), MTLG and MTHG have the same blocking ratio, as then MTHG becomes effectively the same as MTLG. In this case, even though all the algorithms operate without leaking, the blocking ratio of MTLG and MTHG is still smaller than that of LTD-ANCG. The reason for this is that the CLMR algorithm used by

MTLG and MTHG can build light-trees with less add/drop ports than the LTD-ANC algorithm used by LTD-ANCG, and can therefore accommodate more connections. When the leaking ratio threshold increases from 0 to 0.2, the blocking ratios of MTLG and MTHG decrease significantly. This is because of the gain provided by the leaking strategy. (When the leaking ratio threshold is 0.2, the blocking ratios reach the lowest value for this simulation scenario.) If the leaking ratio threshold is greater than 0.2, it is observed that the blocking ratios of both MTLG and MTHG increase with increasing in the leaking ratio threshold. This means that in this case, traffic leaking of about 20% is preferred to attain the lowest blocking ratio. If the leaking ratio threshold is greater than 0.2, traffic leaking causes more resource wastage which consequently degrades the gains obtained from the leaky grooming strategy.

Figure 4-8: Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.3 (The error bars indicate the 95% confidence interval)

As shown in Figure 4-9, if the add/drop ratio is changed to 0.5, MTHG still has the best performance and the lowest blocking ratio is at the leaking ratio threshold of 0.2. However, MTLG has a higher blocking ratio than LTD-ANCG when the leaking ratio threshold is

larger than 0.8. The reason for this is that, when leaking ratio threshold increases to 0.8, too much traffic is leaked, which leads to too much resource wastage in MTLG.

Figure 4-9: Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.5 (The error bars indicate the 95% confidence interval)

Figure 4-10: Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.9 (The error bars indicate the 95% confidence interval)

In Figure 4-10, we provide (for the purpose of completion) an extreme and probably unrealistic example, with add/drop ratio of 0.9, where our proposed solution is not superior. In this scenario, there are a fairly large number of add/drop ports and more light-trees are built. This means that the number of add/drop ports is sufficient, and therefore the wavelengths available now becomes the crucial resource in the network. As expected, LTD-ANCG is observed to perform the best to groom traffic without leaking as it builds light-trees with fewer wavelinks and grooming without leaking does not waste wavelinks. In contrast, in this extreme case, our solution suffers from two weaknesses that lead to wastage of wavelinks and higher blocking ratios. Firstly, the CLMR algorithm, used in grooming algorithms with leaking strategy, requires more wavelinks when it builds light-trees, and secondly, the leaky grooming wastes a large number of wavelinks. We also observe in Figure 4-10 that when leaking is not allowed (i.e. the leaking ratio threshold is 0), MTLG and MTHG have their lowest blocking ratios. This is because if the leaking ratio is zero, the second weakness becomes irrelevant. However, since the first weakness is still relevant, LTD-ANCG performs better than either MTLG or MTHG. When traffic leaking is allowed (leaking ratio threshold has value), the second weakness is valid again, so the blocking ratios of MTLG and MTHG are larger than that without leaking.

Next, we compare between the three algorithms in terms of a measure that we call *light-tree sharing degree* defined as

$$\text{light-tree sharing degree} = \frac{\sum_{i=1}^T \text{number of connections on tree } i}{T} \quad (4.11)$$

where T is the number of light-trees set up in the network. Figures 4-11 to 4-13 show the average light-tree sharing degree vs. the leaking ratio threshold for three different values of the add/drop ratio. In these figures, each point is the average value of twenty sampling results over the simulation running time of an experiment. Generally, the MTHG algorithm achieves

the highest light-tree sharing degree, followed by MTLG and LTD-ANCG. We observe that the two proposed grooming algorithms with leaking strategy have higher light-tree sharing degrees than the LTD-ANCG algorithm. The reason is that the leaking strategies of MTLG and MTHG can groom traffic to light-trees that contain the potential light-trees of LTD-ANCG, so they have higher probabilities to groom traffic than LTD-ANCG. It is clear that in the three figures, MTHG always has a higher light-tree sharing degree than MTLG for different leaking ratio thresholds. This is because the traffic grooming without leaking incorporated in the MTHG algorithm can more efficiently use bandwidth resources, and leave more bandwidth resources for future connections, hence more connections would be groomed into light-trees. The fluctuation of MTLG and MTHG is due to the fact that the light-tree sharing degree is dependent on both the number of light-trees that can groom traffic (leaking ratio threshold decides) and the degree of resources wasted.

Figure 4-11: Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.3

Figure 4-12: Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.5

Figure 4-13: Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.9

Figure 4-14 shows the blocking ratio vs. the network load for the three algorithms when the add/drop ratio is 0.5 and the leaking ratio threshold is 0.2. In this scenario, the MTHG algorithm always performs the best and LTD-ANCG has the highest blocking ratio. Figure 4-15 shows the relationship between the blocking ratio and the add/drop ratio for the three

algorithms when network load is 50 erlangs and the leaking ratio threshold is 0.2. It is noted that, for this scenario, when add/drop ratio is less than 0.6, MTHG outperforms the others, but when add/drop ratio is 0.6, LTD-ANCG performs the best. This is because when add/drop ratio is 0.6, the add/drop port resource is sufficient so that LTD-ANCG uses wavelengths more efficiently than MTHG, while traffic grooming with leaking wastes wavelengths which may lead to higher blocking ratios.

Figure 4-14: Blocking ratio vs. the network load when the add/drop ratio is 0.5 and leaking ratio threshold is 0.2

We next consider a scenario, where connections with lower bandwidth have a larger portion, implying that the average connection rate requirement is further lower than wavelength capacity. Specifically, we assume that there are three different ranges of bandwidths $(0, 0.2C]$, $(0.2C, 0.7C]$ and $(0.7C, C]$ in proportion of 10:2:1, respectively, and the bandwidths required in each range are uniformly distributed. We provide the simulation results under this bandwidth requirement model in Figures 4-16 and 4-17, where the network load is set to 80 erlangs and the add/drop ratio is 0.3. Figure 4-16 shows the blocking ratios for the three algorithms. It is clear that the two proposed leaking algorithms perform better

than LTD-ANCG, and MTHG performs the best. Figure 4-17 compares the light-tree sharing degree for the three algorithms. MTHG has the highest light-tree sharing degree, followed by MTLG and LTD-ANCG. We also note that the light-tree sharing degrees of MTLG and MTHG in Figure 4-17 are much higher than those in Figures 4-11 to 4-13 where the required bandwidth is uniformly distributed between $(0, C]$. This is because there are more lower bandwidth connections in the new traffic scenario, which allows more connections to share a light-tree whose capacity is of a full wavelength capacity C . We also observe that the difference between the newly proposed leaking algorithms and LTD-ANCG increases, this is because with smaller bandwidth requirements, leaking strategy can make more connections share one light-tree than LTD-ANCG, and hence achieve a higher grooming gain.

Figure 4-15: Blocking ratio vs. the add/drop ratio when network load is 50 erlangs and the leaking ratio threshold is 0.2

Since our proposed multicast traffic grooming algorithms with leaking strategy use add/drop ports more efficiently, they can perform significantly better than other algorithms at low add/drop ratios, which is the practical situation in WDM networks. By adjusting the

leaking ratio threshold, proper tradeoff between the add/drop port utilization and the blocking ratio can be achieved. We have tested other network topologies, like USnet and COST239. These give similar results which have not been shown here for the sake of brevity.

Figure 4-16: Blocking ratio vs. the leaking ratio threshold when network load is 80 erlangs and the add/drop ratio is 0.3

Figure 4-17: Light-tree sharing degree vs. the leaking ratio threshold when network load is 80 erlangs and the add/drop ratio is 0.3

4.5 Summary

As add/drop ports are more expensive than wavelengths in WDM networks, we have investigated the problem of multicast traffic grooming with leaking strategy to increase network resource utilization. Two multicast traffic grooming algorithms with leaking strategy, namely, multicast traffic leaky grooming (MTLG) and multicast traffic hybrid grooming (MTHG), have been proposed. The MTLG algorithm grooms traffic to light-trees if the traffic leaked is below a given threshold value. The MTHG algorithm grooms traffic to light-trees without leaking first; then if some destinations are left, it grooms traffic to light-trees with leaking. The MTHG algorithm is an improvement over MTLG as it attains higher light-tree sharing with less traffic leaked. Simulation results have demonstrated that both proposed algorithms perform better than LTD-ANCG for low add/drop ratios and MTHG performs better than MTLG. The tradeoff between the add/drop port utilization and the blocking ratio can be suitably adjusted by adjusting the leaking ratio threshold.

Chapter 5

Design of Multicast Traffic Grooming with Static Traffic

5.1 Introduction

Both lightpaths and light-trees can be used to support multicast connections; but multicast traffic grooming would be very different with these two schemes. Figure 5-1 shows an example of multicast traffic grooming that illustrates two different methods: lightpath based and light-tree based. The network has six nodes and each fiber has one wavelength of OC-12 capacity. Suppose that there are two multicast requests to be accommodated: R_1 from node 1 to nodes 5 and 6 with a bandwidth requirement of OC-3, and R_2 from node 3 to nodes 5 and 6 with a bandwidth requirement of OC-3. In Figure 5-1(a), lightpaths are used to accommodate the two multicast requests using three lightpaths (1→3), (3→5) and (3→6), working on the same wavelength λ_1 . Three lightpaths form a multicast session to transmit the traffic of R_1 . We note that, at node 3, the lightpath (1→3) is terminated at a receiver. Here traffic is processed at the higher layer and then forwarded to two different lightpaths (3→5) and (3→6) to reach the destinations of R_1 . Since the available bandwidth of lightpaths (3→5) and (3→6) are sufficient to accommodate the traffic of R_2 , these two lightpaths are used to groom the

traffic to reach the destinations of R_2 . In total, six higher layer electronic ports and one wavelength are required to accommodate these two requests. If light-tree based traffic grooming is used to accommodate these two requests, as shown in Figure 5-1(b), two light-trees are set up, $(1 \rightarrow 3)$ and $(3 \rightarrow 5, 6)$. Light-tree $(1 \rightarrow 3)$ is a lightpath which is the extreme case of a light-tree with only one destination. Light-trees $(1 \rightarrow 3)$ and $(3 \rightarrow 5, 6)$ form a multicast session to support R_1 , while light-tree $(3 \rightarrow 5, 6)$ can also groom the traffic of R_2 with its spare bandwidth. Five higher layer electronic ports and one wavelength are required to accommodate the requests in this case. Note that this uses one less electronic port than the lightpath based algorithm of Figure 5-1(a) by saving one electronic port at node 3.

Figure 5-1: (a) Lightpath based traffic grooming. (b) Light-tree based traffic grooming

Due to the natural advantages of using a tree topology, a light-tree based logical topology is more beneficial than a lightpath based logical topology for multicast requests, where the logical topology consists of logical one hop trees (LOHTs). In this chapter, we investigate the optimal cost design and provisioning of WDM networks with multicast traffic grooming

based on a light-tree virtual topology in static traffic scenarios. We first propose an ILP formulation to solve this problem with the objective of minimizing the network cost associated with the number of higher layer electronic ports and the number of wavelengths used. Given the scalability problem of ILP, we further provide heuristic algorithms for configuring light-trees in larger networks and evaluate their performance. The solutions obtained from the proposed ILP formulation serve as a bench mark for the results obtained from the heuristic algorithms. (The main content of this chapter has been published in the author's own paper [132].)

5.2 Mathematical (ILP) Formulations

5.2.1 Light-tree based Multicast Traffic Grooming Framework

A light-tree in the physical layer is represented as a LOHT in the logical layer, where the source and destinations of a LOHT are connected within one logical hop, with no interruption to the optical signal. LOHT $(i \rightarrow J)$ is rooted at node i terminating at the set of destinations J in the logical layer. All possible destination sets of LOHTs starting from node i is denoted as the set J_i , the cardinality of J_i is $|J_i| = 2^{|V|-1} - 1$, $i = 1, \dots, |V|$, where $|V|$ is the number of nodes in the network. (LOHTs with one destination contribute $C_{|V|-1}^1$, LOHTs with two destinations contribute $C_{|V|-1}^2$, LOHTs with three destinations contribute $C_{|V|-1}^3, \dots, C_{|V|-1}^{|V|-1}$, so the total is $|J_i| = C_{|V|-1}^1 + C_{|V|-1}^2 + \dots + C_{|V|-1}^{|V|-1} = 2^{|V|-1} - 1$).

We use ILP formulation to model the optimal multicast traffic grooming problem of a set of multicast connection requests. It is noted that the LOHT destination set J_i in this ILP

formulation contains all possible destination sets from node i . Please refer to Table 2-1 for the meanings of notations and variables used in this section.

Minimize:
$$\alpha * \sum_n (TR_n + RR_n) + \beta * \varphi \quad (5.1)$$

This equation shows the optimization objective function of the network cost in terms of the number of transmitters, receivers and wavelengths used in the network. Here α and β represent the relative cost of a transmitter or a receiver and the cost of a wavelength channel, respectively.

Constraints:

— *Constraints on virtual topology variables*

The number of light-trees is constrained by the number of transmitters and receivers in the network. Equation (5.2) ensures that the number of light-trees rooted at node i is less than or equal to the number of transmitters of node i . Equation (5.3) ensures that the number of light-trees terminated at node d is less than or equal to the number of receivers of node d .

$$\sum_{J \in J_i} L_{iJ} \leq TR_i \quad \forall i \quad (5.2)$$

$$\sum_i \sum_{d \in J, J \in J_i} L_{iJ} \leq RR_d \quad \forall d \quad (5.3)$$

Constraint (5.4) ensures that the number of light-trees mapping to the same LOHT ($i \rightarrow J$) equals the sum of the light-trees on different wavelengths.

$$\sum_w L_{iJ}^w = L_{iJ} \quad \forall i, J \in J_i \quad (5.4)$$

As the objective function is to minimize φ , equation (5.5) and the objective function together ensure that φ is the highest index of the used wavelengths. Equation (5.6) ensures that y_w is set to 1 if wavelength w is used by any light-tree.

$$\varphi \geq w * y_w \quad \forall w \quad (5.5)$$

$$y_w \geq \sum_i \sum_{J \in J_i} L_{iJ}^w / Z \quad \forall w \quad (5.6)$$

—Constraints on physical route variables

Equations (5.7) to (5.12) adopted from [133] are to find a path from the source node i to each destination node d of J , then merge all the paths to form a light-tree, which is referred to as a LOHT ($i \rightarrow J$) in the logical layer.

$$\sum_m P_{d,mi}^{iJ,w} = \sum_n P_{d,dn}^{iJ,w} = 0 \quad \forall i, J \in J_i, d \in J, \forall w \quad (5.7)$$

$$\sum_n P_{d,in}^{iJ,w} = \sum_m P_{d,md}^{iJ,w} = L_{iJ}^w \quad \forall i, J \in J_i, d \in J, \forall w \quad (5.8)$$

$$\sum_m P_{d,mq}^{iJ,w} = \sum_n P_{d,qn}^{iJ,w} \quad \forall i, J \in J_i, d \in J, \forall q, q \neq i, d, \forall w \quad (5.9)$$

Equation (5.7) ensures that, the source node i and the destination node d of a light-tree have no incoming and outgoing lightpath stream. Equation (5.8) ensures that for the source node i , and the destination node d , the number of outgoing and incoming lightpath streams are both equal to L_{iJ}^w (the number of the LOHTs ($i \rightarrow J$) on wavelength w). Equation (5.9) ensures that, for an intermediate node of a lightpath from source node i to destination node d , the number of incoming lightpath streams is equal to the number of outgoing streams.

$$\sum_{d \in J} P_{d,mn}^{iJ,w} \geq M_{mn}^{iJ,w} \quad \forall i, J \in J_i, \forall m, n, w \quad (5.10)$$

$$\sum_{d \in J} P_{d,mn}^{iJ,w} \leq |J| \cdot M_{mn}^{iJ,w} \quad \forall i, J \in J_i, \forall m, n, w \quad (5.11)$$

Equations (5.10) and (5.11) are to merge the lightpaths from the source node to each destination node to be a light-tree, denoted as LOHT ($i \rightarrow J$) in the logical layer. By equation (5.10), if the link (m, n) is not traversed by any path from the source to destinations, then this link is not used in the routing of this light-tree. In equation (5.11), the physical routing of light-tree would traverse the link (m, n) if it is used by at least one path pair from the source to the destinations.

$$\sum_i \sum_{J \in J_i} M_{mn}^{iJ,w} \leq P_{mn} \quad \forall m, n, w \quad (5.12)$$

Equation (5.12) ensures that wavelength w of a fiber link can only be occupied by at most one light-tree.

—*Constraints on traffic routing variables*

Each multicast session is to be supported by multiple sub-light-trees (each of which is a LOHT). Equation (5.14) ensures that, at least one sub-light-tree starts from the source node of each multicast request. Equation (5.15) ensures that for every request, the sub-light-trees that are used to support that request are not terminated at the source node of that request.

$$Q_{iJ,n}^t = \lambda_{iJ}^t \quad \forall t, \forall i, J \in J_i, n \in J \quad (5.13)$$

$$\sum_{J \in J_{s_t}} \lambda_{s_t J}^t \geq 1 \quad \forall t \quad (5.14)$$

$$\sum_i \sum_{J \in J_i} Q_{iJ,s_t}^t = 0 \quad \forall t \quad (5.15)$$

Equation (5.16) ensures that each destination of a request has only one incoming stream.

$$\sum_i \sum_{J \in J_i} Q_{iJ,n}^t = 1 \quad \forall t, \forall n \in D_t, n \in J \quad (5.16)$$

Equation (5.17) ensures that each intermediate node of a multicast request (session) has at most one incoming stream.

$$\sum_i \sum_{J \in J_i} Q_{iJ,n}^t \leq 1 \quad \forall t, \forall n, n \notin D_t, n \neq s_t, n \in J \quad (5.17)$$

Equation (5.18) ensures that for every multicast request, except the source node, the root of a sub-light-tree which supports that request must be a destination of another sub-light-tree which also supports that request.

$$\sum_{J \in J_q} \lambda_{qJ}^t \leq |J_q| \cdot \sum_i \sum_{J \in J_i} Q_{iJ,q}^t \quad \forall t, \forall q, q \neq s_t \quad (5.18)$$

Equation (5.19) ensures that for every multicast request, except the source and destinations, each intermediate node of that multicast session must be the root of a sub-light-tree which

support that request.

$$\sum_{J \in J_q} \lambda_{qJ}^t \geq \sum_i \sum_{J \in J_i} Q_{iJ,q}^t \quad \forall t, \forall q, q \notin D_i, q \neq s_i, q \in J \quad (5.19)$$

Equation (5.20) ensures that the bandwidth used by all multicast requests on LOHT ($i \rightarrow J$), must be equal to or less than the total capacity offered by the LOHT ($i \rightarrow J$).

$$\sum_t f_t \cdot \lambda_{iJ}^t \leq L_{iJ} \cdot C \quad \forall i, J \in J_i \quad (5.20)$$

—Constraints on route loop-free variables

The following constraints ensure that the logical route of each request, derived by equations (5.13)-(5.20), is loop-free.

$$H_n^t \leq |V| \cdot \sum_i \sum_{J \in J_i} Q_{iJ,n}^t \quad \forall t, \forall n \quad (5.21)$$

$$H_n^t \geq H_m^t + 1 - (1 - \sum_{J \in J_m} Q_{mJ,n}^t) \cdot |V| \quad \forall t, \forall n, \forall m \neq n \quad (5.22)$$

Equation (5.21) ensures that, for each multicast request, all the nodes which are not traversed by the multicast session have a zero value, and the source node has a zero value as well. Equation (5.22) ensures that, for a multicast request, if node m and node n are in the same LOHT that supports the multicast request, the hop number from the source of the request to node n is larger than that from the source to node m , where the m is the root of that LOHT; if node m and node n are not in the same LOHT that supports the multicast request, this equation is always satisfied.

5.2.2 Number of Variables and Constraints

As the efficiency of an ILP model is decided by the number of variables and constraints, we count these to gain insight into the complexity of the framework. The number of variables in the light-tree based formulation is $O(W |E| |V|^2 2^{|V|} + |R| |V|^2 2^{|V|})$ which grows exponentially

with the number of nodes in the network. The number of constraints is $O(W|V|^3 2^{|V|} + |R||V|^2 2^{|V|})$, which also grows exponentially with the number of nodes. Because of this, the ILP based approach cannot scale to large networks (or even medium sized ones) and heuristic algorithms are needed.

The authors in [111] proposed an ILP formulation for lightpath based multicast traffic grooming, where a multicast session consists of multiple lightpaths. We do not provide this formulation here due to limited space - please refer to [111] for details. In the next section, we will compare the optimal cost of our proposed light-tree based traffic grooming with that of the lightpath based traffic grooming. The numbers of variables and constraints in the lightpath based formulation are $O(W|E||V|^2 + |R||V|^2)$ and $O(W|V|^3 + |R||V|^2)$, respectively, both of which grow more slowly than the light-tree based alternatives. Nevertheless, it is still time consuming to solve the formulation for large networks, and hence lightpath based heuristic grooming algorithms were proposed in [111].

5.3 Optimal Solution of Light-tree and Lightpath based Formulations

To compare the proposed light-tree based algorithm with the lightpath based algorithm of [111], we present an example of optimal cost design of a small network with multicast traffic grooming. The test network has six nodes and eight links, as shown in Figure 5-1. Ten randomly generated multicast requests, shown in Table 5-1, are given as input to the two ILP problems. We assume that the capacity C of a wavelength is OC-12, and the required bandwidth of a multicast demand can be OC-1, OC-3 or OC-12. Solving the ILP problem, we can obtain the optimal solution of the network with these ten multicast requests. The obtained results are given in Tables 5-2 to 5-4.

Our objective is to reduce the network cost in terms of the number of higher layer electronic components and wavelength channels. As explained in [111], the values of parameters α and β depend on the network topology as well actual equipment costs. In [111], a higher layer electronic line terminating system which contains two electronic ports is assumed to cost about \$25,000, and the cost per wavelength of a network graph is approximately equal to $\$250 \times \text{number of edges in the graph} \times 2$. So, we here assume that a higher layer electronic port is three times as expensive as a wavelength in this six-node, eight-edge (bidirectional) network of Figure 5-1, i.e. the values of α and β are set to be 3 and 1, respectively; whereas in the USnet of Figure 5-2, the values of α and β are set to be 3 and 5, respectively. The unit of cost is the normalized value to one unit of the irreducible proportion of one port cost to one wavelength cost. For example, for the six-node network with eight edges, the proportion of one port cost to one wavelength cost is $12500 : (250 \times 8 \times 2) \approx 3 : 1$, so a port cost is three normalized units and a wavelength cost is one normalized unit; for the USnet with 43 edges, the proportion is $12500 : (250 \times 43 \times 2) \approx 3 : 5$, and a port and a wavelength cost is three and five normalized units, respectively. Note that the normalized units are different in the above two networks as a normalized unit is a specific cost proportion of a network.

Figure 5-2: USnet topology

Table 5-2 shows the network cost, number of wavelengths, number of transmitters and number of receivers used. As expected, the optimal cost of the light-tree based formulation is less than that of the lightpath based one (the cost is calculated by $\alpha \cdot (TR + RR) + \beta \cdot \varphi$). This is due to the reduction in the number of transmitters used. The light-tree based formulation uses nine transmitters, while the lightpath based uses twelve transmitters. The reason is that the light-tree based formulation constructs light-trees to support one-to-many connection, where only one transmitter is used at the root of the tree. However, the lightpath based formulation requires several lightpaths to form a logical tree to reach multiple destinations and hence requires more transmitters. From Table 5-2, we can see that both light-tree and lightpath based formulations require the same number of receivers and wavelengths.

Table 5-1: Ten multicast requests for the six-node network

Index	Bandwidth requirement(OC)	Source	Destination
1	3	1	2, 3, 4, 6
2	3	4	1, 2, 3, 6
3	12	1	2
4	3	6	1, 3, 4, 5
5	3	6	1, 4, 5
6	3	1	2, 4, 5, 6
7	1	6	2, 3
8	3	5	2, 3, 6
9	3	6	1, 2, 4
10	3	2	1, 4, 5, 6

Table 5-2: The cost and resource (wavelength, transmitter, and receiver) used

Schemes	Cost	ϕ	TR	RR	No. of TR at each node TR_i ($i=1,2,\dots,6$)	No. of RR at each node RR_i ($i=1,2,\dots,6$)
Light-tree based	64	1	9	12	2, 1, 1, 2, 1, 2	2, 3, 2, 2, 1, 2
Lightpath based	73	1	12	12	2, 3, 2, 2, 1, 2	2, 3, 2, 2, 1, 2

Table 5-3: Light-tree based logical routes of selected requests

Index	Light-tree
4	(6→4, 5), (4→2), (2→1, 3)
5	(6→4, 5), (4→1)
6	(1→3), (3→6), (6→4, 5), (4→2)
10	(2→1, 3), (3→6), (6→4, 5)

Table 5-4: Lightpath based logical routes of selected requests

Index	Lightpath
4	(6→3), (3→2), (2→5), (2→4), (4→1)
5	(6→3), (3→2), (2→5), (2→4), (4→1)
6	(1→3), (3→6), (3→2), (2→4), (2→5)
10	(2→1), (2→5), (5→6), (6→4)

Figure 5-3: The physical routes of light-trees (1→2) and (1→3)

For simplicity, Tables 5-3 and 5-4 show the logical routes of four selected multicast requests (their indexes are 4, 5, 6 and 10, respectively) for the light-tree and lightpath based formulations, respectively. As shown in Table 5-3, each multicast session consists of several sub-light-trees. Sub-light-trees are shared by several multicast requests whose required bandwidth is smaller than a full wavelength bandwidth. For example, sub-light-tree (6→4, 5) is shared by four multicast sessions (requests), and its capacity is fully used. It may be noted that most of the sub-light-trees are rooted at the source or one destination node of a request. This can reduce one receiver since a receiver is needed anyway at each destination node. Similar observations can be found in Table 5-4 for the lightpath based formulation, which are described in [111]. Since a lightpath is just a special case of a light-tree with only one destination and a light-tree can cover multiple destinations in one hop, the light-tree based multicast traffic grooming is naturally more efficient.

The ILP results show that the physical routes of light-trees and lightpaths in the two approaches are distributed over the optical layer so as to effectively minimize the number of wavelengths used. (This arises naturally in the optimization process.) At times, this will lead to a physical route for a light-tree or lightpath which is not the shortest in the given topology, e.g., according to the ILP results, the physical routes of two light-trees, (1→2) and (1→3), are depicted in Figure 5-3. The light-tree (1→3) uses routing 1→4→5→3 instead of the shorter route 1→2→3 in order to use only one wavelength as the link 1→2 has been traversed by light-tree (1→2) already.

The optimal results obtained by solving the light-tree based formulation on the test network provide us with following useful observations that lead us to the development of the heuristic algorithm presented in next section.

-
- Most multicast sessions consist of multiple sub-light-trees, and these sub-light-trees are shared by multiple requests. The size of a sub-light-tree, (i.e., its number of destinations) should be as large as possible so as to save on transmitters (electronic ports).
 - Sub-light-trees which are fully utilized by one or more requests are preferred so as to maximize the optical link resource usage of the network.
 - The root of a sub-light-tree is usually either the source or a destination of a request.
 - The routes of sub-light-trees are not always the shortest optical paths. It may be desirable to distribute the optical link usage over the network so that the number of wavelengths used can be reduced.
 - A request with a full wavelength capacity requirement would be accommodated by only one light-tree.

5.4 Heuristic Algorithms

Motivated by the observations made above from the ILP based optimization approaches, we propose a heuristic algorithm for sub-light-tree based grooming for static multicast requests which would be scalable for handling large networks and will still perform well in terms of our cost parameters. This heuristic algorithm is called *Sub-Light-Tree Saturated Grooming* (SLTSG) algorithm. The flow chart of the SLTSG algorithm is given in Figure 5-4 (The pseudocode of the SLTSG algorithm is given in the Appendix I). It should be noted that, as in the case of the optimization procedures, we presume that the network has enough resources to handle the given set of traffic requests, and the final used resources will be deployed.

In Figure 5-4, the algorithm first constructs a light-tree for every request with a full wavelength capacity requirement as this is the optimal way to support these requests. Since

the complexity of constructing a light-tree is $O(|V|^2 \log |V|)$, the complexity of this step is $O(T|V|^2 \log |V|)$.

Figure 5-4: Flow chart of Sub-Light-Tree Saturated Grooming (SLTSG) algorithm

Then, SLTSG algorithm tries to find out a combination of requests, of which common destinations can be accommodated by a light-tree whose bandwidth can be fully utilized. However, there are 2^n (n is the number of requests each with a sub-wavelength bandwidth requirement) combinations to check, which is very time consuming. Instead, we develop an approximate method to find this combination, i.e., the requests are checked one by one. First,

the requests are sorted in descending order of destination size. The algorithm then tries to find the combination with as large a common destination size as possible to save transmitters (see Appendix I for details). If the selected requests can form a sub-light-tree and this light-tree occupies the whole wavelength channel, the light-tree is built to accommodate the common destinations, and the destinations of the requests are updated; then the algorithm starts over with this finding procedure once again. In this step, finding and constructing a light-tree has a complexity of $O(T^2|V|^3\log|V|)$. The number of light-trees to be constructed is determined by the number of common destinations sets found among the multicast requests, which is at most $T|V|/2$, so the complexity of this step is $O(T^3|V|^4\log|V|)$.

Next, the algorithm tries to accommodate the destinations that are not covered. Here it gives preference to smaller trees because smaller trees are more likely to be shared by the requests. It first calls the *Logical Layer Grooming Algorithm*, then the *Build New Light-trees Algorithm* and the *Light-tree Division-Adjacent Node Component (LTD-ANC) Algorithm* (please refer to Chapter 3 for details of these three algorithms), if necessary. Specifically, *Logical Layer Grooming Algorithm* is to find existing light-trees which can support some of the destinations of a request, and then the traffic is groomed into it. If some destinations still cannot be accommodated, then we use the *Build New Light-trees Algorithm* and the *LTD-ANC Algorithm* to construct light-trees to support these destinations. The *LTD-ANC Algorithm* divides the light-trees derived by the *Build New Light-trees Algorithm* into Adjacent Nodes Components (ANC).

The complexity of *Logical Layer Grooming Algorithm* is $O(|V|^2)$ as at most $|V|$ existing light-trees would be selected for a multicast request, and in the algorithm, to select an existing light-tree, at most $|V|$ nodes have to be checked to see if any light-trees rooted at the nodes can be selected or not. Note that with the constraint of ANC, the number of the light-trees rooted at a node is limited by a constant number, so it would take a constant time

to check all of the light-trees. Since *Build New Light-trees Algorithm* has a complexity of $O(|V|^2 \log |V|)$, and *LTD-ANC Algorithm* has a complexity of $O(|V|)$, the complexity of this step is $O(T|V|^2 \log |V|)$. The overall complexity of the SLTSG algorithm is $O(T^3|V|^4 \log |V|)$.

To reduce the total number of wavelengths used, the link cost would be changed during the operation. If a fiber link has been traversed, the cost of the link will be increased by 1. With this method, the routings of light-trees would be more evenly distributed in the optical layer and should help in reducing the number of wavelengths used.

We compare the performance of our proposed SLTSG scheme with that of LTD-ANCG in Chapter 3. It is noted that LTD-ANCG scheme is similar as SLTSG, where SLTSG can build light-tree fully utilized and partially utilized, and LTD-ANCG only builds light-tree partially utilized. The complexity of the LTD-ANCG algorithm is $O(T|V|^2 \log |V|)$.

A comparison on the running times of the four grooming methods, namely, light-tree optimal, lightpath optimal, SLTSG and LTD-ANCG is given in Table 5-5. This shows that light-tree and lightpath optimal designs require a much longer running time than the heuristic algorithms.

5.5 Numerical Results

We first present numerical results of the light-tree based optimal formulation, the lightpath based optimal formulation, and the SLTSG and LTD-ANCG heuristic algorithms for the test network with 6 nodes and 8 links shown in Figure 5-1. We then compare the performance of the two heuristic algorithms SLTSG and LTD-ANCG using the much larger USnet topology shown in Figure 5-2 which has 24 nodes and 43 links.

5.5.1 Results for the Six-node Test Network

Here ten multicast requests are generated in each simulation experiment. The source of a multicast request is randomly chosen from the network nodes. The multicast destination set size is randomly chosen between 1 and 5 and the destination nodes are randomly selected among the network nodes (excluding the source node). We still assume that the capacity C of a wavelength is OC-12, and the required bandwidth of a multicast demand is randomly chosen to be one of OC-1, OC-3 or OC-12. We set α to 3 and β to 1 as the relative cost parameters. The results are shown in Figures 5-5 to 5-8, where the value at each instance is the average over 15 simulation runs.

Figure 5-5: Cost comparison of different algorithms

Figure 5-5 gives a cost comparison of the four algorithms, light-tree optimal, lightpath optimal, SLTSG and LTD-ANCG. It is clear that the light-tree based optimal algorithm achieves the lowest cost followed by the SLTSG, LTD-ANCG and the lightpath based optimal algorithms. SLTSG has a lower cost than LTD-ANCG. This is because SLTSG

constructs sub-light-trees which can be fully and partially utilized. However, LTD-ANCG constructs light-trees one by one, and each constructed light-tree is subsequently divided into smaller ones which may not be shared by other requests. On average, the heuristic algorithms SLTSG and LTD-ANCG respectively have 16% and 23% higher costs than the light-tree based optimal design. Nevertheless, both of them perform better than the lightpath based optimal design which costs 28% higher than the light-tree based optimal design.

Figure 5-6: Number of transmitters comparison of different algorithms

Figure 5-6 compares the number of transmitters needed by the four methods. Here the optimal light-tree algorithm has the lowest value followed by SLTSG, LTD-ANCG and the lightpath optimal algorithm. It is noted that the optimal lightpath uses the largest number of transmitters. Since a lightpath can reach only one destination, this method would need more lightpaths to support requests. Every lightpath would use a transmitter, which increases the transmitter number. SLTSG uses fewer transmitters than LTD-ANCG. The reason may be that SLTSG uses saturated grooming to increase the utilization of sub-light-trees, which benefits more requests and allows a large portion of the destinations to be covered. Therefore,

fewer additional transmitters would be needed to construct light-trees to cover the remaining destinations. However, in LTD-ANCG, dividing light-tree into smaller ones according to Adjacent Nodes Components (ANC) regardless of light-tree sharing would lead to low utilization and high transmitter usage.

Figure 5-7: Number of receivers comparison of different algorithms

Figure 5-7 compares the number of receivers needed by the four algorithms. Both light-tree and lightpath based optimal algorithms require almost the same number of receivers. This is observed to be less than the number required by SLTSG and LTD-ANCG. It is because of the ILP formulations that nodes other than the source and destination nodes of the requests are not selected as the end points of lightpaths or light-trees, as that would increase the network cost. Therefore, the number of receivers required is almost equal to the sum of the receivers needed at all request destinations. SLTSG uses fewer receivers than LTD-ANCG because SLTSG uses saturated grooming to increase the utilization of sub-light-trees. This is the same reason why SLTSG uses fewer transmitters than LTD-ANCG as shown in Figure 5-6.

Figure 5-8: Number of wavelengths comparison of different algorithms

Figure 5-8 compares the number of wavelengths required by the four algorithms. The lightpath based optimal algorithm has the smallest value, followed by the light-tree based optimal algorithm, SLTSG and LTD-ANCG. The lightpath based optimal algorithm uses lightpaths to accommodate requests and will have a higher probability of finding several lightpaths to groom a multicast request than that of finding light-trees to do so. Therefore wavelength utilization will be higher than the other methods, leading to a lower number of wavelengths used. SLTSG uses fewer wavelengths than LTD-ANCG. Since SLTSG constructs sub-light-trees which can be fully utilized with saturated grooming, we expect it to use fewer wavelengths.

Table 5-5: Running times of four schemes

Scheme	Light-tree optimal	Lightpath optimal	SLTSG	LTD-ANCG
Running time (s)	168639	18720	0.033	0.017

Table 5-5 compares the running times of the four grooming methods: Light-tree optimal, lightpath optimal, SLTSG and LTD-ANCG. The values in Table 5-5 are the averages over 20 simulation runs. (The simulations were run on a PC with 2.8GHz CPU and 1024 MB RAM). It can be seen that the two ILP based optimal schemes have significantly longer running times than the two heuristic algorithms. The light-tree optimal scheme has the longest running time, followed by the lightpath optimal scheme as the latter has fewer constraints and variables. The two heuristic algorithms require much less running times than the optimal ones. LTD-ANCG has the shortest running time as its complexity is less than that of SLTSG.

5.5.2 Results for USnet Topology

These simulations are essentially similar to the simulations done for the small test network except that we consider a much larger network, i.e. USnet with 24 nodes. Multicast requests are once again randomly generated with the multicast destination size ranging from 1 to 23. We assume that the capacity C of a wavelength is OC-48, and that the bandwidth required by a multicast request is randomly chosen from OC-1, OC-3, OC-12 and OC-48. In each simulation experiment we generate 1000 multicast requests. As described before, we used $\alpha=3$ and $\beta=5$ for USnet as the relative cost parameters. It may be noted that even if α and β are set to different values, the dominant factor of the network cost is still the number of higher layer electronic ports, i.e. reducing the number of electronic ports should always be the primary objective for a low cost design. The results are shown in Figures 5-9 to 5-11, where the value at each instance is the average over 15 simulation runs.

Figure 5-9 shows that SLTSG achieves lower design cost than LTD-ANCG. The reason is that SLTSG guarantees the sharing of the sub-light-trees, and uses fewer higher layer

electronic ports while LTD-ANCG constructs a light-tree without knowledge of the other requests leading to lower light-tree utilizations.

Figure 5-9: Cost comparison of SLTSG and LTD-ANCG

Figure 5-10: Comparison of SLTSG and LTD-ANCG in terms of number of electronic ports required

Figure 5-11: Comparison of SLTSG and LTD-ANCG in terms of number of wavelengths required

The results presented in Figures 5-10 and 5-11 demonstrate that LTD-ANCG uses a larger number of higher layer electronic ports and wavelengths than SLTSG, where the number of higher layer electronic ports is equal to the sum of the transmitters and receivers used. It was noted that the cost of all the higher layer electronic ports is about 97% of the total network cost while the cost of all the wavelengths is only about 3%. As noted earlier, this implies that reducing the number of electronic ports is the most effective way to reduce the overall network cost. SLTSG performs better than LTD-ANCG. The reason is that SLTSG grooms multicast requests as much as possible leading to higher utilization of the sub-light-trees. On the other hand, LTD-ANCG grooms requests one by one, and each constructed light-tree is subsequently divided into small ones which may not be shared by other requests.

5.6 Summary

In this Chapter, we have considered the optimal design of WDM networks with multicast traffic grooming under the static traffic scenario. Aiming to minimize the cost associated with the number of higher layer electronic ports and the number of wavelengths used, we have proposed a light-tree based ILP formulation for multicast traffic grooming. The results have revealed that sub-light-tree sharing is more resource efficient than lightpath sharing as the sub-light-tree is an optimal structure to support one-to-many connections.

We have also proposed a heuristic algorithm, namely Sub-Light-Tree Saturated Grooming (SLTSG), which first tries to construct sub-light-trees fully utilized by several multicast requests and then uses the light-tree traffic grooming method of LTD-ANCG to accommodate the remaining destinations. We have compared the performance of the proposed light-tree based ILP formulation, lightpath based ILP formulation and the two heuristic algorithms, SLTSG and LTD-ANCG, in a small test network. Results have demonstrated that the light-tree based formulation incurs the lowest cost, followed by SLTSG, LTD-ANCG, and the lightpath based ILP formulation. Our newly proposed heuristic algorithm SLTSG performs close to the light-tree based ILP optimal design. The heuristic algorithms based on sub-light-tree grooming outperform the lightpath based ILP optimal design because of the natural benefit of light-trees for one-to-many connection. To demonstrate the scalability of our heuristic algorithm, we have studied its performance in a larger network, USnet. Our results have shown that the SLTSG algorithm can scale to a network of practical size with thousands of requests, and it performs better than LTD-ANCG with a higher complexity.

Chapter 6

Constrained Multicast Traffic Grooming with Static Traffic

6.1 Introduction

In Chapter 5, we studied an unconstrained version of the problem of optimizing cost for multicast traffic grooming. In particular, we proposed a light-tree based ILP formulation to minimize network cost in terms of the number of higher layer electronic ports and the number of wavelengths used. However, the number of variables and the number of constraints in the ILP formulation grow exponentially with the number of nodes in the network, which limits the use of this approach to small size networks. This is caused by the large search space arising from the prohibitively large number of node combinations selected as logical one hop tree (LOHT) destinations in the virtual topology design sub-problem, i.e. a network with N nodes will have $N(2^{N-1} - 1)$ possible LOHTs without any constraints. In [115] and [116], optimization problems are constrained to convert the complicated multicast grooming problem to a simpler one. In [115], an ILP optimization problem was formulated for multicast traffic grooming to design a light-tree based logical topology with delay bounds, incorporating the constraint that the light-trees are given a priori, which reduces the

complexity of the problem but optimal results may not be obtained. The model in [116] reduced the multicast traffic grooming problem to a bin packing problem on each link where the constraints were: 1) multicast routing trees were given, and 2) there was no wavelength continuity on light-trees (i.e. wavelength conversion was used). The authors of [134] also introduced a constraint in the problem of spare capacity assignment in survivable networks, where the restoration routes have hop limitations. With this constraint, the computational complexity of the problem was largely reduced and better results were obtained relative to other algorithms. In this chapter, we would like to introduce the hop constraint in multicast traffic grooming. If light-trees are with hop limitations, using hop constrained light-trees to support multicast traffic in WDM networks, the following advantages may be achieved:

- The limitation of splitting times of an optical signal is relaxed. An optical signal can have only a limited number of splits for it to be successfully recovered as every split would decrease the power of light at the output ports.
- Fewer repeaters are needed. The power of light decreases with distance in a long distance optical transmission. Repeaters are required to regenerate the light signal if the quality of signal becomes sufficiently degraded. With hop constraints, the length of transmission in optical domain is largely reduced and repeaters can be eliminated.
- The routing information is reduced since each network node is only associated with a small number of LOHTs. Especially in distributed networks, if every node needs to maintain the routing information, then connections status to be stored is largely reduced.
- The number of LOHTs is inherently limited, which reduces the complexity of the multicast traffic grooming problem.

Using hop constrained light-trees may imply restriction of opportunities to reduce energy consumptions by the use of long optical connections associated with large light-trees.

However, it is difficult to take full advantage of such opportunities because it is difficult to optimize networks with a large number of possible LOHTs.

We here investigate the light-tree design problem with the hop constraint that reduces the number of potential LOHTs and yields a scalable version of the ILP formulation that can simultaneously solve the three sub-problems mentioned earlier, and maximize network throughput. Solving the ILP formulation is time consuming because of the large number of variables and constraints involved. A new heuristic algorithm of polynomial complexity, called *Dividable Light-Tree Grooming* (DLTG), is proposed to efficiently deal with the multicast traffic grooming problem. The DLTG algorithm is based on hop constrained light-trees and can select constrained light-trees to groom traffic or divide light-trees then groom traffic onto these divided light-trees to increase resource utilizations. (The main content of this chapter has been published in the author's own paper [135], and some preliminary results have been presented in conference papers [136] and [137].)

6.2 Problem Description

As discussed in Chapter 5, without any hop constraint, there are $N(2^{N-1} - 1)$ possible LOHTs in an N -node network, which grow exponentially with N . The prohibitively large number of LOHTs makes the optimal design of multicast traffic grooming problem impractical due to the large search space of LOHTs. To reduce the number of possible LOHTs, a straight-forward way is to limit the hops in the light-trees, so that starting from a root node, only a partial set of nodes will be selected as the destinations of light-trees. Other nodes outside this set will not be reached directly but can be reached with multiple hop-constrained light-trees. Increasing the number of hops in the light-trees will increase the number of nodes that are directly reachable.

Figure 6-1: NSFNET topology

In the NSFNET topology shown in Figure 6-1 (we show it again here for the convenience), starting from node 4 as the root node, there are three adjacent nodes (1, 5 and 10) that can be reached in one hop. Considering all trees with one destination, two destinations and three destinations, there are a total of $(2^3-1=7)$ 1-hop light-trees rooted at node 4. (There are C_3^1 LOHTs with one destination, C_3^2 LOHTs with two destinations, and C_3^3 LOHTs with three destinations, so the total is $C_3^1 + C_3^2 + C_3^3 = 2^3 - 1$). These 1-hop light-trees can be represented as in Figure 6-2(a), where the optical information duplication (i.e. power splitting) occurs at the root of the light-trees. An n -hop light-tree is defined as a tree whose maximal number of hops from its root is n . The 2-hop light-trees can be represented as in Figure 6-2(b), where the maximal number of hops from the root is two, but there may be branches with one hop as shown in the figure. In the NSFNET topology, starting from node 4, there are six more nodes (2, 3, 6, 7, 11 and 14) that are reachable with 2-hop light-trees. Including the nodes (1, 5 and 10) reached by 1-hop light-trees, nine nodes are reachable by 2-hop light-trees starting from node 4. The numbers of LOHTs with 1, 2, ..., and 9 destinations are $C_9^1, C_9^2, \dots$, and C_9^9 , respectively, so the total number of LOHTs is $C_9^1 + C_9^2 + C_9^3 + \dots + C_9^9 = 2^9 - 1$. Since the number of 1-hop light-trees from node 4 is still $2^3 - 1$, the number of all the 2-hop light-trees is $(2^9 - 1) - (2^3 - 1) = 2^9 - 2^3 = 504$, which is much greater than that of the 1-hop LOHTs from a node. With reference to Figure 6-2(b), in a 2-hop light-tree, the optical information duplication may

occur at the root of the light-trees and/or adjacent nodes of the root. As we can see, using 2-hop light-trees can reach more nodes than using 1-hop light-trees, but the number of LOHTs is largely increased (i.e., in the above example, it is increased from 9 to 504).

Figure 6-2: (a) 1-hop light-trees. (b) 2-hop light-trees. (c) 2-hop branches

In order to reduce the number of 2-hop light-trees, we here introduce a constraint which limits the number of branches directly outgoing from the root node to be one. The resultant 2-hop light-trees are called 2-hop branches, as shown in Figure 6-2(c), where optical information duplication occurs at the intermediate node, and not at the root node. With reference to NSFNET in Figure 6-1, the number of 2-hop branches from node 4 is only 9, which is much smaller than that of 2-hop light-trees (504).

It can be shown that any 2-hop light-tree can always be constructed by a combination of 2-hop branches and 1-hop light-trees, but this may require multiple transmitters at the root node, since each 2-hop branch or 1-hop light-tree would use one transmitter. However, it is observed that in a multicast network, a one-to-many multicast connection uses more receivers than transmitters, so that the transmitter resource has more redundancy (given that the number of transmitters is equal to the number of receivers because a transmitter and a receiver are usually built together as a transceiver, the detail is discussed next). Thus using 2-hop branches and 1-hop light-trees to construct 2-hop light-trees would cost no more extra network resources as transmitter resource is already redundant, so using 2-hop branches and 1-hop light-trees to support multicast traffic is expected to achieve the same performance as

using 2-hop light-trees and 1-hop light-trees. Note that 2-hop branches are special cases of 2-hop light-trees, so all 2-hop branches are included in the 2-hop light-trees. With 2-hop constrained light-trees, the number of splitting that an optical light may undergo is one or at most two, which ensures that the optical signal can be recovered with good signal quality. These constrained light-trees are similar to the adjacent node components in Chapter 3, where big light-trees are divided into small adjacent node components to increase the utilization of the resources in dynamic traffic scenario. In this chapter, we restrict the hop number of light-trees to be two. It may be noted that even in this case there may be still a large number of 2-hop light-trees.

Table 6-1: Number of LOHTs with different constraints

	Without constraint	With 1-hop light-trees	With 2-hop light-trees	With 2-hop branches
No. of LOHTs	114674	106	5810	381

To illustrate the above, we give in Table 6-1 the different numbers of LOHTs for the NSFNET topology, when (i) there is no constraint, (ii) with a 1-hop light-tree constraint where light-trees are 1-hop light-trees as shown in Figure 6-2(a), (iii) with a 2-hop light-tree constraint where light-trees are 1-hop light-trees in Figure 6-2(a) or 2-hop light-trees in Figure 6-2(b), and (iv) with a 2-hop branch constraint where light-trees are 1-hop light-trees in Figure 6-2(a) or 2-hop branches in Figure 6-2(c). Compared to the unconstrained case, using hop constraints greatly reduces the number of LOHTs. The case with 1-hop light-tree constraint has the lowest value as only adjacent nodes are considered to be the destinations of LOHTs. While using the 2-hop branch constraint reduces the value substantially from that with the 2-hop light-tree constraint. The results presented in the table illustrates that hop constraint dramatically reduces the number of potential LOHTs.

Using hop-constrained light-trees to support multicast connections may degrade network performance, as using multiple small light-trees instead of one large light-tree would consume more resources (i.e., transceivers and wavelengths). However, with traffic grooming, the network performance is not only related to the routing of each connection but also depends on the resource sharing by multiple connections. In multicast traffic grooming, trees of small size (i.e. short trees with a few destinations) would be preferred as that would increase the sharing of light-trees. This is because a light-tree with fewer destinations is more likely to be shared by multiple connections, and multiple small light-trees can be combined to support larger multicast sessions.

In the next section, we present an ILP formulation to analyze multicast traffic grooming based on constrained light-trees. We compare the number of constraints and variables of ILP formulations between the following cases: (i) no constraint, (ii) 1-hop light-tree constraint, (iii) 2-hop light-tree constraint, and (iv) 2-hop branch constraint. We would like to emphasize that the option of 1-hop light-tree constraint is generally not used in practice. However, it is important to use it as an extreme case benchmark to demonstrate that even with the restriction of 1-hop light-trees, reasonable throughput can still be achieved.

We demonstrate that the proposed method significantly decrease the numbers of constraints and variables and it is therefore substantially easier to solve the formulation. However, as the light-trees are constrained, the results achieved may not be optimal. Generally, the results with the 2-hop light-tree constraint are better than those with the 1-hop light-tree constraint or with the 2-hop branch constraint, because the case with the 2-hop light-tree constraint has a larger number of LOHTs (The latter two cases are sub-cases of the former). In this chapter, we mainly focus on using the 2-hop branch constraint because it is simple, require less computation and can nevertheless achieve almost as good a performance as with 2-hop light-tree constraint.

6.3 Problem Formulation

We are given a set of multicast connection requests with sub-wavelength bandwidth requirements, in which the destination size is randomly distributed. Our goal is to maximize the network throughput with limited resources of wavelengths and transceivers, so some connection requests may be blocked. In the multicast connection request set, there may be some connection requests with a full wavelength capacity requirement. For each of these connection requests, a light-tree from the source to the destinations is the optimal way to accommodate the connection request as less resource would be used than using multiple small constrained light-trees, and the light-tree would be fully utilized by the connection.

6.3.1 Mathematical (ILP) Formulation

We here use ILP formulation to model the multicast traffic grooming problem based on constrained light-trees for a given set of multicast connection requests. As a connection request with a full wavelength capacity requirement can be optimally supported by a light-tree without hop constraint, we include a set of light-trees (which are mapped to LOHT destination sets J_i'') in the routings of the connection requests, where each light-tree can optimally accommodate a connection request that requires a full wavelength bandwidth. Specifically, the LOHT destination set J_i in the ILP formulation contains LOHT (light-tree) destination sets with the constraint J_i' and the LOHT destination sets that may be used by connection requests with a full wavelength bandwidth requirement J_i'' , so $J_i = J_i' \cup J_i''$. Please refer to Table 2-1 for the meanings of notations and variables used in this section.

Our aim is to accommodate multicast connection requests to maximize network throughput for the given system resources (i.e. transceivers and wavelengths). Here the network

throughput is defined as the summation of the required bandwidth time destination size of each accommodated connection request. The ILP optimization given below formulates the multicast traffic grooming problem.

$$\textbf{Maximize:} \quad \sum_i f_i \cdot |D_i| \cdot U_i \quad (6.1)$$

This is the optimization objective function of maximizing the network throughput. This equation is also the definition of network throughput: the summation of bandwidth requirement times destination size of the successfully accommodated requests.

Constraints:

— *Constraints on virtual topology variables*

$$\sum_{J \in J_i} L_{iJ} \leq TR_i \quad \forall i \quad (6.2)$$

$$\sum_i \sum_{d \in J, J \in J_i} L_{iJ} \leq RR_d \quad \forall d \quad (6.3)$$

$$\sum_w L_{iJ}^w = L_{iJ} \quad \forall i, J \in J_i \quad (6.4)$$

Equations (6.2) and (6.3) ensure that the number of light-trees starting from a node and terminating at a node is no more than the number of transmitters and receivers at that node respectively. Equation (6.4) ensures that the number of light-trees mapping to a same LOHT equals the number of light-trees on all wavelengths.

— *Constraints on Physical route variables*

$$\sum_m P_{d,mi}^{iJ,w} = \sum_n P_{d,dn}^{iJ,w} = 0 \quad \forall i, J \in J_i, d \in J, \forall w \quad (6.5)$$

$$\sum_n P_{d,in}^{iJ,w} = \sum_m P_{d,md}^{iJ,w} = L_{iJ}^w \quad \forall i, J \in J_i, d \in J, \forall w \quad (6.6)$$

$$\sum_m P_{d,mq}^{iJ,w} = \sum_n P_{d,qn}^{iJ,w} \quad \forall i, J \in J_i, d \in J, \forall q, q \neq i, q \neq d, \forall w \quad (6.7)$$

For each node d in J , equations (6.5) - (6.9) find a path from the source node i to each node d , and then merge all the paths to form a light-tree, this is referred to as LOHT ($i \rightarrow J$) in the

logical layer. Equation (6.5) ensures that, the source i and the destination d of a light-tree have no incoming and outgoing streams, respectively. Equation (6.6) ensures that for the source i , and the destination d , the number of outgoing and incoming lightpath streams are equal to the number of light-trees mapping to LOHTs ($i \rightarrow J$) on wavelength w . Equation (6.7) ensures that, for the path from source i to destination d , the number of incoming streams of an intermediate node is equal to the number of outgoing streams.

$$\sum_{d \in J} P_{d,mn}^{iJ,w} \geq M_{mn}^{iJ,w} \quad \forall i, J \in J_i, \forall m, n, w \quad (6.8)$$

$$\sum_{d \in J} P_{d,mn}^{iJ,w} \leq |J| \cdot M_{mn}^{iJ,w} \quad \forall i, J \in J_i, \forall m, n, w \quad (6.9)$$

$$\sum_i \sum_{J \in J_i} M_{mn}^{iJ,w} \leq P_{mn} \quad \forall m, n, w \quad (6.10)$$

Equations (6.8) and (6.9) are to merge the lightpaths from the source node to each destination to be a light-tree. Equation (6.8) ensures if the link (m, n) is not traversed by any path, then this link is not used by the light-tree. Equation (6.9) ensures the light-tree traverses a link only the link is traversed by at least one path from the source to the destinations. Equation (6.10) ensures that the wavelength w of a fiber link can be occupied by at most one light-tree.

— *Constraints on traffic routing variables*

$$Q_{iJ,n}^t = \lambda_{iJ}^t \quad \forall t \in R', \forall i, J \in J_i', n \in J \quad (6.11)$$

$$\sum_i \sum_{J \in J_i'} Q_{iJ,s_t}^t = 0 \quad \forall t \in R' \quad (6.12)$$

$$\sum_{J \in J_{s_t}} \lambda_{s_t,J}^t \geq U_t \quad \forall t \in R' \quad (6.13)$$

$$\sum_i \sum_{J \in J_i} Q_{iJ,n}^t = U_t \quad \forall t \in R', \forall n \in D_t \quad (6.14)$$

$$\sum_i \sum_{J \in J_i} Q_{iJ,n}^t \leq U_t \quad \forall t \in R', \forall n, n \notin D_t, n \neq s_t, n \in J \quad (6.15)$$

Each multicast connection request with sub-wavelength requirement would be supported by multiple light-trees. Equations (6.11) - (6.17) constrain logical routes for the connection requests. Equation (6.12) ensures there is no incoming stream at the source node and equation (6.13) ensures the number of outgoing streams is larger than one if this connection request is accommodated. Equation (6.14) ensures there is an incoming stream at each destination of a connection request if this connection request is accommodated. Equation (6.15) ensures other nodes except source and destinations should have no more than one incoming stream.

$$\sum_{J \in J_q'} \lambda_{qJ}^t \leq |J_q'| \cdot \sum_i \sum_{J \in J_i'} Q_{iJ,q}^t \quad \forall t \in R', \forall q, q \neq s_t, q \in J \quad (6.16)$$

$$\sum_{J \in J_q'} \lambda_{qJ}^t \geq \sum_i \sum_{J \in J_i'} Q_{iJ,q}^t \quad \forall t \in R', \forall q, q \notin D_t, q \neq s_t, q \in J \quad (6.17)$$

Equation (6.16) ensures that except the source node, the root of a light-tree that supports the connection request must be a destination of another light-tree which also supports that connection request. With this equation, all traversed light-trees are connected together. Equation (6.17) ensures that except the source and destinations, each intermediate node of that multicast session must be the root of some light-trees which support the connection request.

$$Q_{s_t J, n}^t = \lambda_{s_t J}^t \quad \forall t \in R'', J \in J_{s_t}'', n \in J \quad (6.18)$$

$$\sum_{J \in J_{s_t}''} \lambda_{s_t J}^t = U_t \quad \forall t \in R'' \quad (6.19)$$

$$\sum_{J \in J_{s_t}''} Q_{s_t J, n}^t = U_t \quad \forall t \in R'', \forall n \in D_t \quad (6.20)$$

$$\sum_{n \in J} Q_{s_t J, n}^t = \lambda_{s_t J}^t \cdot |D_t| \quad \forall t \in R'', J \in J_{s_t}'' \quad (6.21)$$

Equations (6.18) - (6.21) constrain routes for the connection requests that require a full wavelength bandwidth, and each of such connection requests can be optimally accommodated by one light-tree. Equation (6.19) ensures that only one light-tree starting

from the source node is used to accommodate the connection request. Equation (6.20) ensures that, if a connection request is successfully accommodated, then all its destinations have to be reached by the light-tree starting from the source node. Equation (6.21) ensures that the light-tree has the same number of destinations as the connection request it supports. Equation (6.22) ensures that the used bandwidth is not larger than the capacity offered.

$$\sum_{t \in R} f_t \cdot \lambda_{iJ}^t \leq L_{iJ} \cdot C \quad \forall i, J \in J_i \quad (6.22)$$

— *On logical route loop-free variables*

$$H_n^t \leq |V| \cdot \sum_i \sum_{J \in J_i'} Q_{iJ,n}^t \quad \forall t \in R', \forall n \quad (6.23)$$

$$H_n^t \geq H_m^t + 1 - (1 - \sum_{J \in J_m'} Q_{mJ,n}^t) \cdot |V| \quad \forall t \in R', \forall n, \forall m \neq n \quad (6.24)$$

Equations (6.23) and (6.24) ensure that the logical routes of connection requests with sub-wavelength requirements are loop-free. Equation (6.23) ensures that, for each multicast request, all the nodes which are not traversed have a zero value, and the source node has a zero value as well. Equation (6.24) ensures that, for a multicast request, if node m and node n are in the same light-tree that supports the multicast request, the hop number from the source of the request to node n is larger than that from the source to node m , where the m is the root of that light-tree; if node m and node n are not in the same light-tree that supports the multicast request, this equation is always satisfied.

6.3.2 Number of Variables and Constraints

The above ILP formulation can be solved to obtain the network throughput with the limited network resources. As the complexity of an ILP problem is decided by the number of variables and constraints, we count these to obtain an insight into the problem complexity.

The set $\{J_i' | i \in V\}$ denotes all possible LOHTs for the sub-wavelength connection requests. Since light-trees for these LOHTs may be defined by different constraints or without any constraint, this ILP formulation can be applied to other constrained light-tree design. Four cases are verified next: (i) no constraint, (ii) 1-hop light-tree constraint, (iii) 2-hop light-tree constraint, and (iv) 2-hop branch constraint. The numbers of constraints and variables are given in Table 6-2 for the NSFNET topology where the number of wavelengths is 8 and the number of connection requests is 20. The table shows that the numbers of constraints and variables with different constraints are far less than that without any constraint. The case with 1-hop light-tree constraint has the lowest value, followed by the case with 2-hop branch constraint, and then, the case with 2-hop light-tree constraint. It is noted that the case with 2-hop branch constraint has much lower value than that with 2-hop light-tree constraint.

Table 6-2: Number of constraints and variables for four different cases

	No constraint	1-hop light-tree constraint	2-hop light-tree constraint	2-hop branch constraint
Constraints	5.83×10^8	1.11×10^5	1.05×10^7	4.58×10^5
Variables	2.55×10^9	1.25×10^5	1.24×10^7	5.49×10^5

We note that the complexity of ILP formulation for each of the four cases depends on the number of constraints and the number of variables involved. When there is no constraint, the number of variables is $O(W|E||V|^2 2^{|V|} + |R||V|^2 2^{|V|})$, and the number of constraints is $O(W|V|^3 2^{|V|} + W|E||V|^2 2^{|V|} + |R||V|^2 2^{|V|})$ where both grow exponentially with the number of nodes. This would make the number of constraints and the number of variables excessive for medium and large size networks. The number of variables with the 1-hop light-tree constraint is $O(W|E||V|^2 2^g + |R||V|^2 2^g)$ (g is the maximal node degree), and the number of constraints is

$O(W|V|^3 2^g + WE|V| 2^g + |R||V|^2 2^g)$. The number of variables with the 2-hop light-tree constraint is $O(W|E||V|^2 2^{g^2} + |R||V|^2 2^{g^2})$, and the number of constraints is $O(W|V|^3 2^{m^2} + W|E||V|^2 2^{g^2} + |R||V|^2 2^{g^2})$. The number of variables with the 2-hop branch constraint is $O(gWE|V|^2 2^m + g|R||V|^2 2^g)$ and the number of constraints is $O(gW|V|^3 2^g + gWE|V| 2^g + g|R||V|^2 2^g)$. So the growth rates of the cases with hop constraints are much lower than the case without any constraint. The lowest is with 1-hop light-trees then comes with 2-hop branch constraint and the next is with 2-hop light-tree constraint.

6.4 Performance Evaluations of the Constrained Light-trees

6.4.1 Running Time Comparison

To show the advantages of the hop constrained light-trees, we compare the running times of various ILP formulations with different constraint sets (including the unconstrained light-tree case). The test network is once again the NSFNET topology and the resource configuration is set as $W = 3$, $TR = RR = 3$. We assume that the capacity C of a wavelength is OC-48, and the required bandwidth of a multicast connection request is randomly chosen to be OC-1, OC-3, OC-12 or OC-48 (the term “randomly” in this case refers to a random choice based on a uniform distribution). Twenty multicast connection requests are generated as the input to the ILP formulation. The source of a multicast connection is randomly chosen from the network nodes, the multicast destination size is also randomly chosen in the range of 1 to 13 and the destination nodes are randomly selected from the network nodes (excluding the source node). The simulations were run on a PC with 2.6 GHz CPU and 2048 MB RAM. The average running times of ten simulation runs for each case are given in Table 6-3. The running time of the ILP formulation without any constraint is declared to be *Infeasible* as it could not be

solved for a very long time, i.e. four weeks in our simulation environment. For the case with the 2-hop light-tree constraint, the time required was feasible but was still very long (9.3 days) as compared to the formulation using the 2-hop branch constraint (only 2.25 hours). This is expected as the running time will be higher with a larger number of variables and a larger number of constraints (i.e. a larger search space). The lowest value is the case subjected to 1-hop light-tree constraint, which is 41 minutes. This is because it has the smallest search space, but the case with 1-hop light-tree constraint has the worst performance as will be shown next.

Table 6-3: Running times of ILP formulations with different constraints

	No constraint	1-hop light-tree constraint	2-hop light-tree constraint	2-hop branch constraint
Running Time	Infeasible	41 minutes	9.3 days	2.25 hours

6.4.2 Efficiency of the Proposed Method

The ILP formulation with unconstrained light-trees should perform better than the other cases with hop constrained light-trees as it has the largest search space. The case with 2-hop light-tree constraint should have better performance than the one with 2-hop branch constraint, and 1-hop light-tree constraint performs the worst. Here we investigate the differences in their performance. In order to shorten the simulation time, we compare simulation results in ten randomly generated six-node test networks which are indexed from 1 to 10 in Figure 6-3. To generate a topology, we randomly (with equal probabilities) choose the number of edges in the range of 5 to 9 (5 is minimal value to connect all nodes and 9 is the case that average node degree is 3 which is a large node degree in practice). Then six

nodes are randomly connected to be a tree, after that, the remaining edges are randomly added into the tree to form the topology. The resource configuration is given as $W = 3$ and $TR = RR = 3$. The capacity of a wavelength is OC-12, and the required bandwidth is randomly chosen to be OC-1, OC-3 or OC-12. It is noted that the capacity of OC-N is the exactly N times OC-1, where N is an integer. Ten random multicast connection requests are generated in each simulation experiment as before. The results are shown in Figure 6-3, where the values at each random topology are the average throughput (in the unit of OC-1) of 20 simulation runs.

As shown in Figure 6-3, the throughput of the case without constraint is highest as it is the optimal result. The lowest is the case with the 1-hop light-tree constraint, where light-trees are built among adjacent nodes, as it has the smallest search space. The throughput of the 1-hop light-tree constraint is much lower than the one without constraint. This means that even though an architecture based on 1-hop light-trees can significantly reduce the number of LOHTs and running time, it will lead to the use of more resources (transmitter and receiver) to support multicast connections. On the other hand, the network throughputs of the two formulations with the 2-hop light-tree constraint and with the 2-hop branch constraint are almost the same as the throughput value obtained from the ILP with unconstrained light-trees. The case with the 2-hop light-tree constraint or with the 2-hop branch constraint performs only slightly poorer ($\sim 0.1\%$) than the unconstrained ILP. For other resource configurations, such as $W = 2$, $TR = RR = 2$, or $W = 4$, $TR = RR = 4$, similar results were observed but have not been shown here for the sake of brevity. Therefore, the formulations with the 2-hop light-tree constraint and with the 2-hop branch constraint perform almost as well as the ILP with unconstrained light-trees in these six-node network tests, and the optimization with the 2-hop light-tree constraint has the same performance as that with the 2-hop branch constraint. This is because with the 2-hop branch constraint, light-trees can be assembled to any 2-hop

light-trees with the cost of more transmitters which is an adequately available resource in multicast networks, and in small test networks, the large difference is not significant as only a small number of LOHTs is reduced.

Figure 6-3: Network throughput of formulations with different constraints in random networks

For large networks, more LOHTs would be reduced by constraints, so smaller search spaces are obtained. Therefore, the difference between the cases without any constraint and with the 2-hop light-tree (or branch) constraint becomes larger. Moreover, the larger differences between alternatives with 2-hop light-tree constraint and with 2-hop branch constraint are also expected. Figure 6-4 compares the network throughputs in NSFNET topology with different constraints except for the case of ILP without any constraint. Note that in NSFNET topology, the ILP formulation without any constraint leads to infeasible running times. In Figure 6-4, the simulation is configured as that in Table 6-3, where each instance in the figure is the average value of five simulation experiments. As shown in Figure 6-4, the lowest performance is still the ILP formulation with 1-hop light-tree constraint,

followed by 2-hop branch constraint and 2-hop light-tree constraint. We can see that the latter two cases have larger difference than in the six nodes networks of Figure 6-3, where the value of the 2-hop branch constraint is about 3.3% higher than that of the 2-hop light-tree constraint, which confirms the prediction of larger differences in larger networks. So, with larger networks, the ILP formulations with 2-hop branch constraint and with 2-hop light-tree constraint still perform very close.

Figure 6-4: Network throughput of formulations with different constraints in NSFNET

6.4.3 Numerical Results of the Proposed Formulation

Further results for the 2-hop branch constrained ILP formulation are provided for different network resources. Consider the six nodes test network depicted in Figure 5-1. A set of connection requests, comprising 10 multicast connection requests, is generated in the same way as before. The total traffic load of the 10 connection requests is OC-186 calculated by equation as follows,

$$\text{traffic load} = \sum_{i=1}^{\text{no. of reqests}} |D_i| \times f_i \quad (6.25)$$

where D_i is the destinations of connection request i and f_i is the required bandwidth of connection request i .

Table 6-4: Network throughput (total traffic is OC-186)

	$W=1$	$W=2$	$W=3$	$W=4$
$TR=RR=1$	OC-54	OC-54	OC-54	OC-54
$TR=RR=2$	OC-102	OC-102	OC-102	OC-102
$TR=RR=3$	OC-103	OC-150	OC-150	OC-150
$TR=RR=4$	OC-103	OC-186	OC-186	OC-186

Table 6-5: Average wavelength link utilization ($TR=RR=3$)

	$W=1$	$W=2$	$W=3$	$W=4$
Link utilization	94%	59%	56.3%	38%

Table 6-6: Average resource utilization ($W=3$)

	$TR=RR=1$	$TR=RR=2$	$TR=RR=3$	$TR=RR=4$
Wavelength Link	20.8%	37.5%	56.3%	60%
Receiver	100%	100%	100%	96%
Transmitter	100%	75%	66.7%	62.5%

Table 6-4 shows the network throughput for different network resource limitations, where network throughput is defined in Equation (6.1) as the summation of bandwidth requirement times destination size of the successfully accommodated requests. We can see that exactly the

same results are obtained when the number of wavelengths per fiber (W) is set to be 2, 3 and 4. This is because the available wavelength resources are sufficient as long as two or more wavelengths per fiber are available. This conjecture can be verified by the average wavelength link utilization provided in Table 6-5, which gives the ratio of the number of wavelength links used to the total number of wavelength links. In this table, the number of transceivers is set to 3, and while increasing W from 2 to 4, the utilization decreases from 59% to 38% as increasing the already abundant wavelength resource will lead to even lower utilizations. When W is set to 1, some throughput values are different from those when it is set to 2. When the number of transceivers is 3 and the number of wavelengths is 1, the network throughput is OC-103. If we increase the number of wavelengths from 1 to 2 and keep the number of transceivers unchanged at 3, the throughput will increase to OC-150, which means that, with 3 transceivers at each node, one wavelength is not sufficient.

However, when we increase transceivers, the network throughput shows significant improvement. When W is set to 3, and the number of transceivers varies from 1 to 4, the corresponding average resource utilizations (wavelength link, receiver and transmitter) are shown in Table 6-6. The average wavelength link utilization increases from 20.8% to 60% as more transceivers produce more light-trees, using more wavelength links. In a multicast network, one-to-many connections may use more receivers than transmitters, in which case the receiver resource is exhausted first while the transmitter resource remains redundant. As shown in the table, the receiver is always fully used as this resource is always needed until all traffic demands are accommodated; however, the transmitter utilization always have redundancy except for $TR = RR = 1$. Note that increasing transceivers decreases the average utilization of the transmitters. This is again a result of increasing the redundant resource which leads to lower utilization. As there are redundant resources of transmitters in multicast networks, and recall that using 2-hop branches and 1-hop light-trees to construct 2-hop

light-trees may consume more transmitters, the 2-hop branch constraint is an efficient way to replace 2-hop light-tree constraint and operate the network.

6.5 Heuristic Approaches

As the network size grows, the number of variables and the number of constraints of the ILP increases. In this case, solving the ILP problem may become computationally prohibitive. Heuristic algorithms are therefore needed to efficiently solve the problem. As a higher layer electronic port is usually connected to a transmitter (receiver) for transmitting (receiving) optical signals, the number of transmitters and receivers used in the network equals the number of higher layer electronic ports they are connected to them. This may lead to a trend where the transceiver resource would be scarcer than the wavelength resource in a network. Based on the results observed in Section 6.4.3, we can conclude that effective heuristic algorithms should make efficient use of transceiver resources, i.e. more specifically the receiver resource, as it tends to become exhausted first.

Motivated by these findings, we here propose a heuristic algorithm, called *Dividable Light-Tree Grooming* (DLTG), which is based on constrained light-trees and can use receiver resource efficiently to improve network throughput. The flow chart of the DLTG algorithm is given in Figure 6-5 (The pseudocode of the DLTG algorithm is given in the Appendix J).

Figure 6-5: Flow chart of Dividable Light-Tree Grooming (DLTG) algorithm

In the DLTG algorithm, all connection requests are sorted first, and it tries to accommodate connection requests with larger throughput first as they may contribute more to the overall network throughput. Connection requests in the sorted list are processed sequentially from the start to the end of the list. There are two loops implementing the idea of the algorithm. The first loop is to select existing light-trees to groom a connection request, and the destinations of a selected existing light-tree must be a subset of the destination set to be reached. In this loop, all the selected light-trees will not be divided. The second loop of the algorithm selects light-trees to be partially groomed, i.e. the destinations of the selected light-tree will be divided into two parts: the intersection with the destination set to be reached (with groomed traffic) and the rest (without groomed traffic). To increase the utilization, in each loop, the algorithm selects the light-tree which can honor the maximum number of nodes in the destination set to be reached.

If necessary, a new light-tree is derived to cover the destinations which have not been groomed, and then, this light-tree is divided into constrained light-trees (1-hop light-trees and 2-hop branches). To divide a light-tree, we use the Light-Tree Division - Adjacent Node Component (LTD-ANC) algorithm in Chapter 3, where 2-hop branches are divided out with a higher priority. It is noted that a connection request with a full wavelength capacity requirement goes directly to build a new light-tree to provide an optimal accommodation.

The complexity of the first loop is $O(|R||V|^2)$. Notice that the upper bound of the number of existing light-trees is $|R||V|$, and the number of light-trees selected is at most $|V|$. The second loop has the same complexity $O(|R||V|^2)$. In the last step, Build a light-tree has complexity of $O(|V|^2 \log |V|)$ and dividing a light-tree into small ones has complexity of $O(|V|)$. In total, the complexity of the algorithm is $O(|R|^2|V|^2 + |R||V|^2 \log |V|)$.

In the numerical results, another heuristic algorithm, Light-Tree Division - Adjacent Node Component based Grooming (LTD-ANCG) in Chapter 3 is also considered to compare its

performance with that of the new proposed DLTD algorithm. The LTD-ANCG algorithm has been proposed for a dynamic multicast traffic scenario with the aim of decreasing the blocking ratio and it is a good heuristic algorithm using light-trees to accommodate multicast connections. It also uses constrained light-trees with hop limitation to support traffic, but cannot divide existing light-trees to increase the resource utilization when grooming traffic. The complexity of LTD-ANCG is $O(|R||V|^2 \log|V|)$, when applied to a multicast connection request set with $|R|$ connection requests.

6.6 Numerical Results

6.6.1 Results for NSFNET Topology

In this section, we present numerical results of the ILP formulation, the LTD-ANCG and the DLTG algorithms, all subject to the 2-hop branch constraint, for the NSFNET topology with the resource configuration $W=5$, $TR=RR=5$. Twenty random multicast connection requests are generated in each simulation experiment as before. The results are shown in Figure 6-6 and Figure 6-7, where the throughput value at each instance is the average over 20 simulation runs.

In Figure 6-6, it is clear that the throughput value of ILP with the 2-hop branch constraint is higher than that of both the LTD-ANCG and DLTD algorithms. This is expected because the ILP formulation provides the optimal result subject to 2-hop branch constraint. Interestingly, the average network throughput of the DLTG heuristic algorithm is only 3.4% less than that of the ILP formulation. The DLTD algorithm performs significantly better than LTD-ANCG with about 40% higher throughput. This is because the DLTG algorithm is designed for static traffic with the objective of maximizing network throughput, and it can

improve the resource sharing by both grooming traffic to light-trees totally and partially, while LTD-ANCG is for dynamic traffic with the objective of minimizing the connection blocking ratio, which may not contribute much to increasing the network throughput. Moreover, LTD-ANCG shares light-trees with the condition that the destinations of the light-tree must belong to the set of connection destinations. In Figure 6-6 we also provide results for average of total traffic load. In each instant, these results are calculated based on Equation (6.25) for each of the 20 simulation runs, and then, the total traffic values are averaged. These values are the offered traffic, and are the optimistic bounds for throughput results.

Figure 6-6: Network throughput comparison

In Figure 6-7, the light-tree sharing degrees (Equation (4.11)) of LTD-ANCG and DLTG algorithms are compared. It represents the degree to which the light-tree resource is being shared. As shown in Figure 6-7, the sharing degree of DLTG is much higher than that of LTD-ANCG, i.e. about 37% higher. A higher sharing degree would benefit the network throughput as more traffic can then be accommodated in the network. This is also the reason

why the DLTD algorithm has a better network throughput performance than LTD-ANCG. The average running times are also compared - these are 3.4 hours, 0.008 seconds and 0.01 seconds for ILP formulation, LTD-ANCG and DLTG, respectively. As expected, the running times of the two heuristic algorithms are much lower than that of the ILP formulation. DLTG has a slightly longer running time than LTD-ANCG because of its higher complexity. Similar results were obtained for other resource configurations (e.g. $W=3$, $TR=RR=3$ or $W=6$, $TR=RR=6$) but have not been shown for the sake of brevity.

Figure 6-7: Comparison of light-tree sharing degrees of LTD-ANCG and DLTG algorithms

6.6.2 Results for Usnet Topology

We have also studied the performance of the DLTG algorithm for USnet topology shown in Figure 5-2, which has 24 nodes and 43 links. These simulations are essentially similar to those done for the NSFNET topology except that here we consider a much larger network. Twenty-five connection request sets are generated as inputs, and each set comprises 200 multicast connection requests. We compare the average network throughput and average

sharing degree of 25 connection request sets for different number of transceivers configured at each node, where each fiber link has sufficient wavelengths (we set W to 128).

In Figure 6-8, we present the offered traffic load and network throughput results obtained by two heuristic algorithms, where the network throughput is also defined as the summation of the required bandwidth time the destination size of each successfully accommodated connection request (Equation (6.1)). The traffic load is calculated by averaging the individual traffic load of each of the 25 multicast connection request sets using Equation (6.25). Notice that this traffic load value remains constant for the different configurations simply because we use the same 25 multicast connection request sets. Again, this traffic load value represents an optimistic bound for other algorithms.

Figure 6-8: Network throughput vs. number of transceivers per node

Another optimistic bound for the throughput can be obtained by considering the maximum throughput that can be supported by the network resources. This bound designated as *resource bound* is equal to the wavelength capacity times the number of receivers times the number of nodes, namely, $C \cdot RR \cdot |N|$, as that would mean that all receivers in the network are fully used for traffic. Accordingly, the minimum of the traffic load and the resource bound

represents an upper optimistic bound for the throughput. As can be seen in Figure 6-8, the DLTG algorithm gives significantly higher throughput than LTD-ANCG, i.e. about 22% higher than LTD-ANCG on average. If we increase the number of transceivers, the network throughputs of DLTG and LTD-ANCG would also increase as more transceivers would build more light-trees to accommodate more traffic. When the number of transceivers is increased to about 45 for DLTG and to 55 for LTD-ANCG, all connection requests can be accommodated. The average network throughput of DLTG algorithm for different number of transceivers configurations is about 5% lower than the average value of the upper bound. It is noted that the optimal value would be smaller than the upper bound value, so the network throughput of DLTG algorithm would be even closer (less than 5%) to the optimal one.

Figure 6-9: Light-tree sharing degree vs. number of transceivers per node

Figure 6-9 compares light-tree sharing degrees of the DLTG and LTD-ANCG algorithms. As shown in the figure, DLTG has a much higher sharing degree than LTD-ANCG, i.e. about 26% higher. It may be noted that when we increase the number of transceivers per node, the sharing degree for DLTG increases. This is mainly because more transceivers build more

light-trees, and then DLTG is better able to groom traffic to light-trees totally and partially, thereby increasing the sharing degree. In contrast, Figure 6-9 also shows that with LTD-ANCG, more light-trees lead to slightly decreasing sharing degree. This occurs mainly because the degree of light-tree sharing cannot increase at the same pace as the increase of the number of light-trees. The average running time of a simulation experiment (comprising 200 multicast connection requests) averaging over all 25 connection request sets and over all configurations was found to be about 0.23 seconds for DLTG and 0.21 seconds for LTD-ANCG. Apart from the fact that these values are low enough to implement in a realistic network, the difference between them is insignificant.

6.7 Summary

We have investigated a light-tree design problem for multicast traffic grooming in WDM networks with hop constraint. The hop constraint reduces the number of possible LOHTs. We have considered examples of 1-hop light-trees and 2-hop branches. With such constrained light-trees, multicast sessions can be supported by combining several small constrained light-trees and the utilization of light-trees can also be improved. We have also proposed an ILP formulation for multicast traffic grooming with the constrained light-trees. By only considering light-trees with 2-hop branch constraint, the search space of the ILP formulation is greatly reduced. Simulations on NSFNET topology have verified that the ILP formulation with the 2-hop branch constraint works much faster than the formulation with the 2-hop light-tree constraint. We have also found that these two ILP formulations give almost the same performance in randomly generated six-node test networks and NSFNET topology.

The ILP results have also revealed that, in a network with multicast traffic, the receiver resource is usually exhausted first, and hence making efficient use of receivers would have a

significantly higher impact on improving the network throughput. Based on these observations, we have proposed a heuristic algorithm with a polynomial complexity - Dividable Light-Tree Grooming (DLTG), which is based on hop constrained light-trees. The algorithm can select constrained light-trees to groom traffic onto them and can also divide light-trees, and then groom traffic onto these divided small light-trees to increase the receiver utilization. We have compared the performance of the ILP formulation with the 2-hop branch constraint, LTD-ANCG and DLTG algorithms in the NSFNET topology. Results have shown that the performance of the newly proposed DLTG algorithm is far better than that of LTD-ANCG, and is close to the performance of the ILP formulation with significantly shorter running times. We have also verified the performance of DLTG algorithm in USnet topology and observed that DLTG not only provides better performance than LTD-ANCG, but it also performs close to an optimistic (upper) bound in this network.

Chapter 7

Multicast Traffic Grooming in Tap-and-Continue WDM Mesh Networks

7.1 Introduction

An optical power splitter which splits incoming light into multiple identical copies in the optical domain is required for a light-tree [15]. Network nodes which have these optical splitters are referred to as *multicast capable nodes* as they can split an incoming optical signal into multiple copies. A cost-effective way to achieve light splitting in the optical domain is by fusing fibers together; however, the power loss due to splitting needs to be compensated by deploying active amplifiers, e.g. erbium-doped fiber amplifiers (EDFAs) [18], so that the signals can still be detected by optical receivers. Unfortunately, optical amplifiers have negative side-effects like non-uniform gain over the operating waveband, gain saturation and additional noise [138]. For a node to provide full multicast capability it would have to be equipped with a large number of splitters for every wavelength of all the input fibers and a large number of optical amplifiers would also be required. This increases the cost of the node and is also difficult to implement. The network cost may be reduced with sparse light splitting [28, 32, 138-140], where only a subset of nodes are multicast capable. The works

reported in [28, 32, 138-140] find the minimal cost light-forest (multiple light-trees) to accommodate a multicast connection request, with the additional constraint that light signals cannot be split at multicast incapable nodes.

Figure 7-1: A trail $s \rightarrow d_1 \rightarrow s \rightarrow d_2 \rightarrow d_3 \rightarrow d_2 \rightarrow n \rightarrow d_4$ supporting a multicast connection request ($s; d_1, d_2, d_3, d_4$)

The high cost, negative side-effects and the control complexity of multicast capable nodes have motivated investigation of the multicast problem in networks without multicast capable nodes. The authors of [141] considered the multicast routing problem in a WDM network with multicast incapable nodes that have a “*Tap-and-Continue*” (TaC) feature. The function of TaC is to tap a small amount of optical power for the local station while forwarding the remainder to an output port. A *trail* is a path that starts from a source node and visits all destination nodes of a multicast connection [141] sequentially. It was introduced to support the multicast connection, in which each destination node can tap the optical signal for the local station. Finding a trail with minimum total link cost is an NP-complete problem for which an approximate heuristic was proposed in [141]. This mechanism substantially reduces the cost of the network in comparison to multicast capable networks, but still has the problem of longer trails to route. Figure 7-1 illustrates a trail $s \rightarrow d_1 \rightarrow s \rightarrow d_2 \rightarrow d_3 \rightarrow d_2 \rightarrow n \rightarrow d_4$ to accommodate a multicast connection request ($s; d_1, d_2, d_3, d_4$), where s is the source and d_1 to d_4 are the destination nodes. At each destination node, the optical signal is tapped to the local station and the remainder is forwarded to the next node in the trail. In this example, the

number of links used by the trail is 7. It is noted that s and d_2 are traversed twice in the trail, links (s, d_1) and (d_2, d_3) are also traversed twice but in opposite directions. If a light-tree is used to support the same connection request, the number of links to be used is 5, and the optical signal would be split at node s and d_2 .

To the best of our knowledge, no research has been done on optimal design of multicast grooming in tap-and-continue WDM networks (based on trails). In this chapter, we investigate the optimal design and provisioning of WDM networks with multicast traffic grooming in multicast incapable networks where network nodes do not have multicast capability but do support the tap-and-continue mechanism, and trails instead of light-trees are used to convey multicast traffic. We propose a simple and efficient node architecture with the tap-and-continue mechanism, which can be a simple and cost effective upgrade to existing network nodes designed for unicast traffic. This new node architecture is also expected to simplify network management and control. Using nodes of this type and for networks using trails instead of light-trees for multicasting, an ILP formulation is proposed to minimize the network cost (computed based on the number of higher layer electronic ports and the number of wavelengths used). A new heuristic algorithm of polynomial complexity, called *Multicast Trail Grooming* (MTG), is proposed to efficiently deal with the multicast traffic grooming problem in large networks. It is noted that both of the proposed ILP formulation and heuristic algorithm are not just limited to networks with our proposed node architectures but may also be applied to other networks with the TaC capability. The solutions obtained from the proposed ILP formulation are used to benchmark the results obtained from our heuristic algorithm. We also investigate the routing of trail and propose a new *Node Adding Trail Routing* (NATR) algorithm which performs better with shorter trails than others reported earlier [141]. (The main content of this chapter has been published in the author's own paper [142].)

7.2 Node Architectures

A node architecture with the TaC mechanism was originally proposed in [141]. It taps a small amount of optical power at each destination on a trail, while forwarding the remaining power to downstream destinations. Such TaC networks can implement multicasting with potentially much lower node costs than networks with other kind of multicast capable nodes (e.g. node architecture with splitter-and-delivery in [34]) because TaC networks do not require signal splitting loss compensation (by optical amplifiers).

However, in the node architecture proposed in [141], only one TaC Module (TCM) is deployed for each wavelength, which means at most one optical signal at each wavelength can be tapped for the local station of the node. When the multicast traffic load is heavy, this architecture may cause a large blocking probability due to the lack of TCM to tap the signal. This can be avoided by a simple extension of the node architecture of [141] as shown in Figure 7-2, with multiple TCMs connected to each wavelength-routing switch (WRS). Here there are M_i TCMs ($1 \leq i \leq W$ and $1 \leq M_i \leq N$, where W is the number of wavelengths per fiber, N is the number of the output fibers of the node) connected to the WRS on wavelength i . Multiple TCMs can support multiple trails of each wavelength to be tapped for the local station. The maximum number of TCMs deployed for each wavelength can be equal to the number of the output fibers N (full tap capability). A TCM is illustrated in Figure 7-3, where 0.5% of an input optical signal is tapped by a TAP (Tapping device) for the local station, and the rest is switched to an output port through a multistage network of 1×2 switching elements. If the maximum number of TCMs is deployed, the number of switching elements introduced is $NW \sum_{i=0}^{\lceil \log_2 N \rceil} 2^i$, and the number of TAPs will be NW . The size of a WRS (i.e. the number of input/output ports) will also increase with the number of TCMs deployed. For example, as shown in Figure 7-2, if there are M_i TCMs, A_i add ports, and B_i drop ports at the WRS on

wavelength i , then the size of each WRS is $(N+A_i) \times (N+M_i+B_i)$, where $1 \leq i \leq W$. If the maximal number of TCMs is deployed, the size would be $(N+A_i) \times (2N+B_i)$. In addition, the control complexity will be much higher, since both the WRSs and the switching elements in the TCMs have to be controlled.

Figure 7-2: A TaC node architecture with multiple TCMs

We propose an alternative node architecture illustrated in Figure 7-4, which has the full tap capability, i.e. any input signal from fibers can be tapped. Here every output port of a WRS has a TaC deployed to tap its signal. This TaC device is simply a 1×2 coupler without using photonic switches, whose function is to tap a small fraction of the signal and forward it to the local station, while the remainder continues to go to the Mux. The size of a WRS used in this

new node architecture is smaller. For example, as shown in Figure 7-4, if there are A_i add ports, and B_i drop ports at the WRS on wavelength i , where $1 \leq i \leq W$, the size of each WRS is $(N+A_i) \times (N+B_i)$, which is smaller than the earlier value $(N+A_i) \times (2N+B_i)$ for the full tap capability if both architectures have the same number of add ports, and drop ports. The number of TaC devices will be NW . The tapped signal can be received or discarded at the receiver bank, and the dropped signals (terminated at the node) are directly switched to drop ports by WRSs, then received by the receiver bank.

Figure 7-3: A 1×8 TCM consisting of $\lceil \log_2 8 \rceil = 3$ stages of 1×2 switches

To build a trail from a node to multiple destinations, a higher layer electronic port at the source node (we also call it transmitting electronic port) is used to manage the traffic transmission. An optical transmitter in the transmitter bank of the node is used to modulate the traffic into a wavelength which is then sent to an add port of the WRS working on that wavelength. For receiving the traffic, two possible scenarios may arise. If this is the end node (i.e. the last node of the trail), then the optical signal is switched to the receiver bank through a drop port of the WRS on that wavelength and a receiver in the bank is used to receive the dropped signal. This is then given to a higher layer electronic component through a higher layer electronic port (we also call this kind of port as a receiving electronic port). Alternatively, at an intermediate node which is a destination of the trail (receiving the traffic),

the optical signal is switched to the next node by a WRS. A TaC device then taps the signal and forwards it to the receiver bank and a receiver in the bank is configured to receive the tapped signal. Finally, the traffic is sent to a higher layer electronic component through a receiving electronic port. This implies that each transmitting electronic port will need one optical transmitter and one add port of a WRS, while each receiver (used by a drop port or a tapped signal) will need to use one receiving electronic port.

Figure 7-4: The proposed TaC node architecture with the full tap capability

The actual amount of signal to be tapped down depends on the sensitivity of the optical receivers used. A receiver with higher sensitivity can successfully detect lower power signals but would be costlier. Generally, for an OC-48 transmitter, the output power of a laser is about 1 mw, a tap device can even tap 1/1000 power [143] out for the local station, and the

receiver sensitivity need to be about -30dBm for reasonably priced receivers. The maximal number of tapping operations or the maximal number of hops of a light trail depends on the output power of laser source, the sensitivity of receivers and the tap ratio of tap devices. Higher output power of laser source and/or higher sensitivity of receivers lead to a larger number of tapping operations. The tap ratio of tap devices has to be guaranteed to tap enough power for the receiver to successfully convert information from optical signal to electrical signal.

Commercially, the output power of a laser is usually a few dBm and the sensitivity of receiver ranges from -20 dBm to -30 dBm. For a receiver with a sensitivity of -20 dBm, it can successfully detect the optical signal if the received power is no less than 1/100 mw. That means, the tap device should have 1/100 tap ratio (or higher, like 1/90, 1/80, 1/70, ...) for the 1 mw (0 dBm) light power. The tap devices with various tap ratios (1/100, 1/90, 1/80, 1/70, ...) are off-the-shelf [144]. In TaC networks with the proposed node architecture of Figure 7-4, a light signal can undergo tens of tapping operations with the commercial tap devices. This maximal number of tapping operations is higher than that in realistic optical networks as most trails will have far fewer physical hops (number of tappings).

Our goal is to design the network with minimal cost, in terms of the number of higher layer electronic ports and the number of wavelengths used in the network. The detailed outputs of the design process are as follows:

- *A set of trails which constitute the logical layer:* A trail has one source node, one end node and multiple intermediate nodes. A higher layer electronic port and one add port of a WRS are used at the source node to send out the multicast traffic, and an electronic port and a drop port of a WRS is also used at the end node as the termination (also destination) of the trail; intermediate nodes configure their receiver banks to receive the tapped signals as destinations (which will take one electronic port each in the higher layer) or

discard the tapped signals. Therefore, the transmission from the source to all the destinations takes only one logical hop and the signal stays in the optical domain from the source to destinations, making the trail as a logical one hop tree (LOHT) in the logical layer.

- *Traffic routing*: the routing information of multicast traffic demands. A multicast connection request may be accommodated by multiple LOHTs in the logical layer, which are trails in the optical layer. Multiple LOHTs (or trails) are connected together in the electronic domain by higher layer electronic components. Usually, electronic packet switching is used for interconnecting LOHTs (trails), implementing traffic forwarding from an upstream LOHT to downstream LOHTs with O-E-O conversion.
- *Trail routing and wavelength assignment (RWA)*: the routing information and wavelength assignment of all the trails in the optical layer.

It has been shown that the unicast traffic grooming based on lightpaths is an NP-complete problem. Since the unicast is a special case of the multicast with only one destination, and since lightpath is a special case of trail with only one destination node, then by generalization, the multicast traffic grooming based on trail is also NP- complete.

7.3 Mathematical (ILP) Formulation

A trail can be identified by the combination of its source node, end node and intermediate nodes along with information on whether a receiver (or a receiving electronic port) is needed to receive the tapped signal or not. There are large combinations of nodes to be included in or excluded from a trail. In ILP formulations, we should try to avoid introducing a large number of variables and constraints caused by the problem of the large number of combinations. We find out that, although the number of combinations of nodes is large, the number of

connections from a node is small. In a realistic WDM mesh network, a node is usually physically adjacent to only a few nodes so the node degree is small. The number of connections sourcing from or terminating to a node is limited by its node degree. Specifically, the number of connections is less than or equal to the node degree times the number of wavelengths W . This implies that the number of trails that start from a node and work on a specific wavelength cannot exceed the node degree.

A trail is denoted by (i, w, k) , where i is the source node of the trail, w denotes the wavelength used by the trail, and k is the index of the trail, $1 \leq k \leq \text{Deg}(i)$, where $\text{Deg}(i)$ is the node degree of node i . The information of the receiver (receiving electronic port) usage at intermediate nodes and the end node of trail (i, w, k) , is denoted by a Boolean parameter $T_{i,w,k}^q$, which takes the value of 1 if node q uses a receiver to receive the tapped/dropped signal on trail (i, w, k) , otherwise 0. Note that the end node of the trail has $T_{i,w,k}^q = 1$ as the end node uses a receiver to terminate the signal. There are two types of intermediate nodes, one of which does not use a receiver (called a *forwarding node*) and has $T_{i,w,k}^q = 0$, and the other with $T_{i,w,k}^q = 1$ which uses a receiver (called *tap-receiving node*). Accordingly, there may be four types of nodes on a trail: source node, end node, forwarding node and tap-receiving node.

7.3.1 The ILP Formulation

We use ILP formulation to model the optimal multicast traffic grooming problem based trails for a set of multicast connection requests in TaC networks. Please check the means of notations and variables used in this section in Table 2-1.

$$\text{Minimize:} \quad \alpha * \sum_n (A_n + B_n) + \beta * \varphi \quad (7.1)$$

This equation shows the optimization objective function of the network cost in terms of the number of transmitting electronic ports, receiving electronic ports and wavelengths used in the network. Here α and β represent the relative costs of a higher layer electronic port and a wavelength channel, respectively.

—*Constraints on network resources variables*

Equation (7.2) ensures that the number of trails starting from node q and working on a wavelength w , is no larger than the number of add ports of the WRS for the wavelength w at node q . Equation (7.3) ensures that the total number of add ports used in a node is no larger than the total number of transmitting electronic ports at the node. Equation (7.4) ensures that for each wavelength, the number of trails terminating at node q is no larger than the number of drop ports of the WRS on that wavelength at node q . It is noted that if trail (i, w, k) is built, its end node has at most one link connected to it as the last physical hop. Equation (7.5) ensures that the number of receivers (receiving electronic ports) used at a node is no larger than the total number of receiving electronic ports at that node.

$$\sum_k U_{q,w,k} \leq A_q^w \quad \forall q, w \quad (7.2)$$

$$\sum_w A_q^w \leq A_q \quad \forall q \quad (7.3)$$

$$\sum_i \sum_k \sum_m M_{i,w,k}^{mq} \leq B_q^w \quad \forall q, w \quad (7.4)$$

$$\sum_i \sum_w \sum_k T_{i,w,k}^q \leq B_q \quad \forall q \quad (7.5)$$

As the objective is to minimize φ , Equation (7.6) and the objective function (7.1) together ensure that φ is the highest index of the used wavelengths. Equation (7.7) ensures that y_w is set to 1 if wavelength w is used by any trail. Equations (7.8) and (7.9) ensure that trail (i, w, k) is built if it uses at least one receiver.

$$\varphi \geq w * y_w \quad \forall w \quad (7.6)$$

$$y_w \geq \sum_i \sum_k U_{i,w,k} / Z \quad \forall w \quad (7.7)$$

$$U_{i,w,k} \geq \sum_q T_{i,w,k}^q / |V| \quad \forall i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.8)$$

$$U_{i,w,k} \leq \sum_q T_{i,w,k}^q \quad \forall i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.9)$$

—Constraints on Physical route variables

Equations (7.10) - (7.19) are the commodity-flow conservation constraints to create a physical route for trail (i, w, k) . Equation (7.10) ensures that, for the source node i , the number of units of outgoing commodity is larger than that of incoming commodity by the number of destinations of the trail. It is noted that a source node may have an incoming stream, e.g., in Figure 7-1 source node s has an incoming stream from d_l . Equation (7.11) ensures that for each trail (i, w, k) , if node q is a destination of the trail, the number of units of incoming commodity is larger than that of outgoing commodity by 1; otherwise, they are equal. Equations (7.12) and (7.13) ensure that if link (m, n) transmits commodity of trail (i, w, k) , then this link is traversed by the trail; otherwise, the link is not traversed.

$$\sum_{n \in \text{adj}(i)} F_{i,w,k}^{in} - \sum_{m \in \text{adj}(i)} F_{i,w,k}^{mi} = \sum_q T_{i,w,k}^q \quad \forall i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.10)$$

$$\sum_{m \in \text{adj}(q)} F_{i,w,k}^{mq} - \sum_{n \in \text{adj}(q)} F_{i,w,k}^{qn} = T_{i,w,k}^q \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall q \neq i \quad (7.11)$$

$$F_{i,w,k}^{mn} \geq M_{i,w,k}^{mn} \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall mn \in E \quad (7.12)$$

$$F_{i,w,k}^{mn} \leq |V| \cdot M_{i,w,k}^{mn} \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall mn \in E \quad (7.13)$$

The main difference between a trail and a light-tree is that: a trail does not split the incoming optical signal, while a light-tree splits the incoming optical signal to multiple copies at certain nodes. Equations (7.14)-(7.19) ensure that trail routing does not split an incoming optical signal. Equation (7.14) ensures that, if trail (i, w, k) is built, a node which is adjacent to source i is traversed as the first hop. Equation (7.15) ensures that, other nodes except the

source node, have no adjacent node traversed as the first hop. Equation (7.16) ensures that, for each trail, the number of outgoing streams forwarded from a specific incoming stream at a node is no larger than 1, so incoming optical signals are not split. Equation (7.17) ensures that if link (q, n) is traversed by a trail, there should be one incoming stream to node q which is forwarded to link (q, n) in the trail.

$$\sum_{n \in \text{adj}(i)} S_{i,w,k}^{i,i,n} = U_{i,w,k} \quad \forall i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.14)$$

$$\sum_{n \in \text{adj}(q)} S_{i,w,k}^{q,q,n} = 0 \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall q \neq i \quad (7.15)$$

$$\sum_{n \in \text{adj}(q)} S_{i,w,k}^{m,q,n} \leq 1 \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall q, m \in \text{adj}(q) \cup \{q\} \quad (7.16)$$

$$\sum_{m \in \text{adj}(q) \cup \{q\}} S_{i,w,k}^{m,q,n} = M_{i,w,k}^{qn} \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall q, n \in \text{adj}(q) \quad (7.17)$$

Equation (7.18) ensures that if a node is traversed and the trail returns to the upstream node along the same link with the opposite direction, this node is a destination. This equation can avoid unnecessary links being made into a trail. Equation (7.19) ensures that the number of units of commodity on the incoming stream is no less than that on the outgoing stream, where the outgoing stream is directly forwarded from the incoming stream. Equation (7.20) ensures that the wavelength w of a fiber link (m, n) can only be occupied by at most one trail.

$$S_{i,w,k}^{m,q,m} \leq T_{i,w,k}^q \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall q, m \in \text{adj}(q) \quad (7.18)$$

$$F_{i,w,k}^{mq} \geq F_{i,w,k}^{qn} - (1 - S_{i,w,k}^{m,q,n}) |V| \quad \forall i, w, 1 \leq k \leq \text{Deg}(i), \forall q, m \in \text{adj}(q), n \in \text{adj}(q) \quad (7.19)$$

$$\sum_i \sum_k M_{i,w,k}^{mn} \leq P_{mn} \quad \forall w, mn \in E, \quad (7.20)$$

—Constraints on traffic routing variables

Each multicast connection request is supported by at least one trail. These trails connect each others in the electronic domain to form a multicast session. As $Q_{i,w,k}^{t,q}$ is set to $\lambda_{i,w,k}^t$ times $T_{i,w,k}^q$, we use Equations (7.21) and (7.22) to assign value to $Q_{i,w,k}^{t,q}$, which ensure that if

connection request t traverses trail (i, w, k) , and node q uses a receiver to receive the signal on that trail, then $Q_{i,w,k}^{t,q}$ will be set to 1, otherwise it will be 0.

$$\lambda_{i,w,k}^t + T_{i,w,k}^q \geq 2Q_{i,w,k}^{t,q} \quad \forall t, q, i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.21)$$

$$\lambda_{i,w,k}^t + T_{i,w,k}^q \leq Q_{i,w,k}^{t,q} + 1 \quad \forall t, q, i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.22)$$

The following equations constrain the incoming and outgoing streams of a source node. Equation (7.23) ensures that, at least one trail would start from the source of the connection request. Equation (7.24) ensures that for every connection request, no trail that supports the connection request terminates at the source of that request.

$$\sum_w \sum_k \lambda_{s_t, w, k}^t \geq 1 \quad \forall t \quad (7.23)$$

$$\sum_i \sum_w \sum_k Q_{i, w, k}^{t, s_t} = 0 \quad \forall t \quad (7.24)$$

Equation (7.25) ensures that each destination of a connection request must have one incoming stream. Equation (7.26) ensures that for every connection request, except the source and destinations of the connection request, other nodes have at most one incoming stream, i.e. each intermediate node of the multicast session has one incoming stream, and other nodes which are not traversed by the session have no incoming stream.

$$\sum_i \sum_w \sum_k Q_{i, w, k}^{t, q} = 1 \quad \forall t, q \in D_t \quad (7.25)$$

$$\sum_i \sum_w \sum_k Q_{i, w, k}^{t, q} \leq 1 \quad \forall t, q \neq s_t, q \notin D_t \quad (7.26)$$

Equation (7.27) ensures that, except the source node of the connection request, the starting node of the trail that supports the connection request must be the end node or a tap-receiving node of another trail which also supports that request. Equation (7.28) ensures that, except the source and destinations of the connection request, each end node or a tap-receiving node on that multicast session must be the starting node of some trails which support the connection request.

$$\sum_w \sum_k \lambda_{q,w,k}^t \leq \text{Deg}(q) \cdot W \cdot \sum_i \sum_w \sum_k Q_{i,w,k}^{t,q} \quad \forall t, q \neq s_t \quad (7.27)$$

$$\sum_w \sum_k \lambda_{q,w,k}^t \geq \sum_i \sum_w \sum_k Q_{i,w,k}^{t,q} \quad \forall t, q \neq s_t, q \notin D_t \quad (7.28)$$

Equation (7.29) ensures that the bandwidth used by all multicast connection requests in the trail (i, w, k) must be equal to or less than the total capacity offered by that trail.

$$\sum_t f_t \cdot \lambda_{i,w,k}^t \leq C \quad \forall i, w, 1 \leq k \leq \text{Deg}(i) \quad (7.29)$$

—Constraints on traffic route loop-free variables

The following constraints ensure that the routing of each connection request is loop-free. Equation (7.30) ensures that, for each connection, nodes which are not traversed by the connection have a zero value, and the source node of the connection has a zero value as well. Equation (7.31) ensures that, for each connection, if trail (p, w, k) is traversed, where p is the source of the trail and node q on this trail uses a receiver, the hop number from the source of the request to node q is larger than that from the source to node p ; if two nodes are not on the same trail used by the connection request, this equation is always satisfied.

$$H_q^t \leq |V| \cdot \sum_i \sum_w \sum_k Q_{i,w,k}^{t,q} \quad \forall t, q \quad (7.30)$$

$$H_q^t \geq H_p^t + 1 - (1 - \sum_w \sum_k Q_{p,w,k}^{t,q}) \cdot |V| \quad \forall t, q, \forall p \neq q \quad (7.31)$$

Note that if *bifurcation* is used then the traffic can be divided into components where each component is routed separately. This would increase the complexity and cost because of traffic bifurcation and reassembly may introduce delay jitter [111]. In the ILP formulation proposed here, if a trail (i, w, k) has been used by a connection request, all traffic of the request is conveyed by this trail, so bifurcation is not allowed.

7.3.2 Number of Variables and Constraints

As the complexity of an ILP problem is decided by the number of variables and constraints, we count these to get an insight into the complexity of the framework. The number of variables is $O(|R|W|V||E|+gWE|^2)$, which grows quadratically with the number of edges in the network, where g is the maximal node degree. The number of constraints is $O(|R|W|V||E|+gWE|^2)$, which also grows quadratically with the number of edges. As the size of the network grows (the number of edges also increases in connected networks), solving the ILP problem becomes very time consuming and prohibitively complicated. Therefore, the ILP based approach cannot scale to large networks. For these, heuristic algorithms are needed to deal with the problem efficiently.

7.4 An Example of the ILP Formulation

Figure 7-5: A six-node network topology

We present here an example of a TaC network which uses the ILP formulation described above for multicast grooming. The test network has six nodes as shown in Figure 7-5 (we show it again in this these for convenience). Ten randomly generated multicast connection requests, shown in Table 7-1, are given as the input to the formulation. The source of a multicast connection request is randomly chosen from the network nodes, the multicast

destination size is also randomly chosen ranging from 1 to 5. The destination nodes are randomly selected from the network nodes (excluding the source node). We assume that the capacity C of a wavelength is OC-12, and the required bandwidth of a multicast connection request is a random integer with uniform distribution from 1 to 12 (an integer i denotes a bandwidth of OC- i). To compare the performance of the proposed trail-based ILP formulation, the light-tree based ILP formulation proposed in Chapter 5 is also examined in the simulation. We used a commercial ILP solver, “CPLEX” to solve the ILP formulations.

Table 7-1: Ten multicast connection requests

Index	Bandwidth requirement(OC)	Source	Destinations
1	2	4	1, 6
2	12	6	1, 3, 5
3	10	6	1, 2, 3, 4, 5
4	1	5	1, 2
5	1	6	2, 3, 5
6	6	3	1, 2, 4, 5
7	6	1	2, 3, 4, 5, 6
8	7	6	1
9	11	2	5, 6
10	3	3	4, 5

The ILP objective is to reduce the network cost in terms of the number of higher layer electronic components and wavelengths deployed. As explained in [111], the values of parameters α and β depend on the network topology as well actual equipment costs. Following [111], we assume that a higher layer electronic port is three times as expensive as

a wavelength in this six-node network, i.e. the values of α and β are set to be 3 and 1, respectively (please refer to the descriptions about the relative costs of an electronic port and a wavelength in Chapter 5.3). Solving the ILP problem, we obtain the optimal solution of accommodating these ten multicast connection requests. These results are given in Tables 7-2 to 7-4.

Table 7-2: The cost and resource (wavelength, transmitter, receiver) usage

Schemes	Cost	φ	A	B	No. of transmitting elec. ports at each node, A_i ($i=1,2,\dots,6$)	No. of receiving elec. ports at each node, B_i ($i=1,2,\dots,6$)
Trail-based	90	3	9	20	1, 2, 1, 1, 1, 3	4, 3, 3, 3, 5, 2
Light-tree based	89	2	10	19	2, 2, 1, 1, 1, 3	4, 2, 3, 3, 5, 2

Table 7-2 shows the network cost, numbers of transmitting and receiving electronic ports, and wavelengths obtained by the trail-based ILP formulation and the light-tree based ILP formulation. The trail-based ILP formulation has a slightly higher cost (about 1.1% higher) than the light-tree based ILP formulation, and this extra cost arises from the one more wavelengths used in the trail-based formulation. However, the number of required higher layer electronic ports is the same in the two formulations. Given a light-tree, a trail with the same source and destinations can be derived by visiting all nodes of the tree (start from the source then go every node in sequence), and some nodes may be visited multiple times, this will lead to more links to travel. Therefore, the trail-based ILP formulation results in the same number of higher electronic ports as the light-tree based formulation, but using more wavelengths. In Table 7-2, the trail-based formulation and the light-tree based case use the same number of electronic ports, but the allocations are different, where nine and twenty

electronic ports are used for transmitting and receiving in the trail-based case, while ten and nineteen electronic ports are used in the light-tree based case. In this example, even though the trail-based ILP formulation has a slightly higher cost than the light-tree based one, it has the advantage that costly multicast capable nodes are not needed.

In this six-node test network, the trail-based ILP formulation has the same number of higher layer electronic ports as the light-tree based ILP formulation. The reason is that in such a small network, a higher layer electronic port is three times as expensive as a wavelength. As a result, increasing one wavelength is more economical than increasing one more electronic port, and also one additional wavelength usage is sufficient for rerouting a light-tree to form a trail with the same number of electronic ports. However, if large networks are applied, the cost of deploying one more wavelength in all optical links of a network could be higher than that of increasing one more electronic port (α and β are changed accordingly). In such case, the trail-based formulation would prefer using more electronic ports than using more wavelengths, and the cost of an electronic port has to be lower than the cost of a wavelength, otherwise the feasible solution with more wavelengths and same number of electronic ports is preferred. This balance procedure arises naturally in the optimization process. It is noted that one more wavelength could benefit many trails as a trail usually uses only a few links in a network, especially in a large network with many optical links. That means, in most cases, the cost of one more wavelength is distributed to multiple trails, which reduces the relative cost of a wavelength. This implies that even in large networks, the solution with the same (or slightly more) number of electronic ports as the light-tree based formulation and with more wavelengths usage is preferred by the trail-based case. So in small networks, the trail-based ILP formulation prefers the solutions with the same number of higher layer electronic ports as the light-tree based ILP formulation and with a few more wavelengths; in large networks, the trail-based formulation may also prefer the solution with

the same (or a slightly more) number of electronic ports as the light-tree based case but with a few more wavelengths. In both scenarios, the trail-based formulation would perform very close to the light-tree based case with the cost difference of at most a few more wavelengths.

Table 7-3: Traffic routings of connection requests based on trail

Index	Trails
1	(4→6) (6→1)
2	(6→1, 3, 5)
3	(6→2, 3, 5) (2→1, 4)
4	(5→1, 2)
5	(6→2, 3, 5)
6	(3→4, 5), (5→1, 2)
7	(1→2, 3, 4, 5), (4→6)
8	(6→1)
9	(2→5, 6)
10	(3→4, 5)

Table 7-3 shows the traffic routings of ten multicast connection requests in the trail-based ILP formulation. In the table, a trail is denoted as a LOHT in the logical layer, e.g. (1→2, 3, 4, 5) denotes a trail started from node 1, using receiving electronic ports to receive tapped/dropped traffic at nodes 2, 3, 4 and 5. Some multicast sessions consist of several trails, and some trails are shared by several multicast connection requests whose required bandwidth is smaller than a full wavelength bandwidth. For example, trails (3→4, 5) and (6→2, 3, 5) are shared by multicast connection requests 6, 10 and 3, 5 separately. It is noted that most of the trails are rooted at the source or one destination node of a connection request. This can reduce one electronic port since an electronic port is needed for receiving the traffic

anyway at each destination node, e.g. the routing of the 6th connection request consists of two trails, (3→4, 5) and (5→1, 2), and these two trails connect each other at node 5, which is a destination of the connection request. Other connection requests, like the 1st, 3rd and the 7th, behave similarly.

Table 7-4: Physical routings of trails

Trail	Physical route
(1→2, 3, 4, 5)	1→2→4→5→6→3 (λ_3)
(2→1, 4)	2→1→4 (λ_1)
(2→5, 6)	2→3→6→3→5 (λ_1)
(3→4, 5)	3→2→4→5 (λ_1)
(4→6)	4→2→3→6 (λ_2)
(5→1, 2)	5→3→2→1 (λ_3)
(6→1)	6→5→4→1 (λ_1)
(6→1, 3, 5)	6→5→3→2→1 (λ_2)
(6→2, 3, 5)	6→3→5→4→1→2 (λ_2)

Table 7-4 shows that the physical routes of trails are evenly distributed over the optical layer so as to minimize the number of wavelengths used (This arises naturally in the optimization process). At times, this will lead a physical route to have the non-shortest routing in the given topology, e.g. the physical routes of two trails, (6→1, 3, 5) and (6→2, 3, 5). The trail (6→2, 3, 5) uses routing 6→3→5→4→1→2 instead of the shortest one 6→5→3→2 in order to reuse the wavelength λ_2 as the wavelength and the links have been traversed by trail (6→1, 3, 5) already. This phenomenon has also been found in the light-tree base ILP formulation with the same objective in Chapter 5.

From Table 7-4, we can see that the nodes denoted with bold number are destinations of trails, which are either tap-receiving nodes or end nodes, where only tap-receiving nodes need tap devices to tap optical signals for local stations; other end nodes can receive optical signals directly from WRSs through drop ports. If we check the resource configurations for the two different node architectures shown in Figures 7-2 and 7-4, we should see the improvement of our newly proposed architecture. From Table 7-4, the numbers of tappings at nodes 1 - 6 are 1, 2, 2, 2, 3 and 1, respectively. This implies that the numbers of TCMs deployed at nodes 1 - 6 are 1, 2, 2, 2, 3 and 1, respectively, for the first node architecture in Figure 7-2. In this case, the number of WRS output ports has to be increased accordingly. However, if the 2nd node architecture with the full tap capability in Figure 7-4 is applied, no extra WRS output ports are needed.

In the above example, it took tens of hours to obtain the optimal results by solving the ILP formulation running on a PC with 2.8GHz CPU and 1024 MB RAM. The ILP approach cannot scale to large networks, so heuristic algorithms are needed. The optimal results obtained by solving the ILP formulation on the test network provide us with following useful observations that lead us to develop the heuristic algorithm presented in Section 7.6.

- Multicast sessions may consist of multiple trails, and a trail may be shared by multiple connection requests.
- The root of a trail is usually either the source or a destination of a connection request.
- The routes of trails are not always the shortest. It may be desirable to distribute the optical link usage over the network so that the number of wavelengths used can be reduced.
- A connection request with a full wavelength capacity requirement would be optimally accommodated by one trail.

7.5 Trail Routing

Before describing our proposed heuristic trail-based grooming algorithm, we first investigate the trail routing problem itself as that would determine the performance of heuristic grooming algorithms. In [141], the problem of routing a trail in a TaC network with the minimal cost is called the multiple destination minimum cost trail (MDMCT) problem, and it has been proved to be a NP-complete problem. The multiple destination trail (MDT) algorithm was proposed to solve MDMCT problem [141]. There are two steps in this MDT algorithm. The 1st step is to use the minimum path heuristic (MPH) algorithm developed in [40] to find a Steiner tree for the multicast connection. The 2nd step is to derive the trail by visiting each branch of the Steiner tree in sequence. Since the cost of the Steiner tree obtained by the MPH is up to twice of the optimal one [40] and the links of the Steiner tree may be traversed twice, the MDT heuristic is a 4-approximation algorithm for the MDMCT problem.

In the MDT algorithm, the idea of visiting each branch of the Steiner tree to derive the trail may result in a longer route for the trail, because all but one of the branches of the tree would be traversed downward and upward. We instead propose a new heuristic routing algorithm, called Node Adding Trail Routing (NATR) algorithm, to efficiently route a trail with fewer links to travel.

7.5.1 Node Adding Trail Routing (NATR) Algorithm

The idea of the NATR algorithm is to add destinations one by one into a trail, while avoiding long routes to be generated. An initial trail will be given at first and then the trail is subsequently developed by adding other destinations of the connection request with minimal costs. In Figure 7-6, three scenarios of adding nodes are given. Figure 7-6(a) is the original

trail. In Figure 7-6(b), a destination node d is attached to a node m of the trail, and the trail will be changed to $s \rightarrow m \rightarrow d \rightarrow m \rightarrow n \rightarrow e$, where node m is traversed twice. Figure 7-6(c) shows the scenario that node d is appended to the end node e of the trail. In Figure 7-6(d), destination node d is added between node m and n , so the trail becomes $s \rightarrow m \rightarrow d \rightarrow n \rightarrow e$, and the edge (m, n) is not traversed in the new trail. Accordingly, the costs of adding node d into the trail in the three scenarios are: (b) $\text{cost}(m-d) + \text{cost}(d-m)$, (c) $\text{cost}(e-d)$, and (d) $\text{cost}(m-d) + \text{cost}(d-n) - \text{cost}(m-n)$, where $\text{cost}(i-j)$ denotes the link cost from node i to node j (with direction). To select a destination to add into, all destinations which are not reached by the trail are checked in the three scenarios, and the destination d is selected with the criteria of the minimal adding cost. The flow chart of the NATR algorithm is given in Figure 7-7 (The pseudocode of the NATR algorithm is given in the Appendix K).

Figure 7-6: Three node-adding scenarios

In the algorithm, the initial trail routing is selected from the shortest paths from the source node to each of destinations, the complexity of this step is $O(|V|\log|V|)$. The initial trail will then be developed to a feasible solution by adding destinations one by one. As the trail can not traverse an edge with same direction twice [141], so we will exclude the edges of the trail already traversed. The destination with the minimal adding cost will be selected, and then the

path(s) will be added into the trail routing. The complexity of improving the solution is $O(|V|^3 \log |V|)$ as there are at most $|V|$ node to be added into and each adding operation would have the complexity of $O(|V|^2 \log |V|)$ (node pairs routing is $O(|V|^2 \log |V|)$, selecting a node is $O(|V|^2)$). Therefore, the overall complexity of NATR algorithm is $O(|V|^3 \log |V|)$.

Figure 7-7: Flow chart of Node Adding Trail Routing (NATR) algorithm

7.5.2 Optimal Trail Routing

We use an ILP formulation to derive the optimal trail routing of a multicast connection. A multicast connection request $(s; D)$ is given, and the ILP formulation is used to find the optimal routing of the trail for the connection request. Please refer to Table 2-1 for the means of notations and variables used in this section.

Minimize:
$$\sum_{mn \in E} M_{mn} \quad (7.32)$$

The objective function is to minimize the number of edges to accommodate the multicast connection request $(s; D)$.

$$\sum_n F_{sn} - \sum_m F_{ms} = |D| \quad (7.33)$$

$$\sum_m F_{md} - \sum_n F_{dn} = 1 \quad \forall d \in D \quad (7.34)$$

$$\sum_m F_{mq} = \sum_n F_{qn} \quad \forall q \neq s, q \notin D \quad (7.35)$$

$$F_{mn} \geq M_{mn} \quad \forall mn \in E \quad (7.36)$$

$$F_{mn} \leq |D| \cdot M_{mn} \quad \forall mn \in E \quad (7.37)$$

Equation (7.33) ensures that the total number of units of the outgoing commodity is larger than that of the incoming commodity by $|D|$ at the source node s . Equation (7.34) ensures that the number of units of the incoming commodity is one larger than that of outgoing commodity at each destination node. Equation (7.35) ensures that the incoming stream and the outgoing stream of an intermediate node of the trail have the same units of commodity. Equations (7.36) and (7.37) ensure that edge (m, n) is used by the trail if it has commodity traversed through it, otherwise, it is not used.

$$\sum_{n \in \text{adj}(s)} S_{s,n}^s = 1 \quad (7.38)$$

$$\sum_{n \in \text{adj}(q)} S_{q,n}^q = 0 \quad \forall q \neq s \quad (7.39)$$

$$\sum_{n \in \text{adj}(q)} S_{q,n}^m \leq 1 \quad \forall q, m \in \text{adj}(q) \cup \{q\} \quad (7.40)$$

$$\sum_{m \in \text{adj}(q) \cup \{q\}} S_{q,n}^m = M_{qn} \quad \forall q, n \in \text{adj}(q) \quad (7.41)$$

$$F_{mq} \geq F_{qn} - (1 - S_{q,n}^m) |D| \quad \forall q, m \in \text{adj}(q), n \in \text{adj}(q) \quad (7.42)$$

Equation (7.38) ensures that, there is an adjacent node of source s traversed as the first hop. Equation (7.39) ensures that, other nodes except the source node, have no adjacent node traversed as the first hop. Equation (7.40) ensures that, the number of outgoing streams forwarded from a specific incoming stream is no larger than 1. Equation (7.41) ensures that if link (q, n) is traversed, there should be one incoming stream to node q which is forwarded to link (q, n) . Equation (7.42) ensures that the number of units of commodity on the incoming stream is no less than that on the outgoing stream, if these two streams are connected at the node.

In this ILP formulation, the number of variables is $O(g|E|)$ and the number of constraints is also $O(g|E|)$, where g is the maximal node degree.

7.5.3 Simulation Results

As discussed above, the MDT algorithm may have long trails to route, which is caused by the rerouting of the Steiner tree (obtained by MPH). The rerouting process will traverse all except one branch to reach all destinations in a trail. Therefore, the best rerouting result is to traverse all the branches downward and upward, except the longest branch (from the source to the farthest leaf node) which is traversed only downward. This is equal to twice the cost of the Steiner tree minus the path cost from the source to the farthest leaf in the tree. To evaluate

our proposed trail routing algorithm, NATR, we compare the performances of NATR algorithm, optimal trail routing derived from ILP formulation and the best result of MDT.

We compare the performances of the three algorithms for the NSFNET network with 14 nodes shown in Figure 3-11. In each simulation experiment, one multicast connection request is generated as the input to each of the three methods. The number of edges of the trail is counted to compare the performance of the different methods, i.e. the number of edges per request (or per experiment as there is only one connection request in an experiment) indicates the efficiency of the routing algorithms. The source of a multicast connection request is randomly chosen from the network nodes, and the destination nodes are randomly selected from the network nodes (excluding the source node). The results are shown in Figure 7-8 where the values at each destination size are the average value of 100 simulation runs.

Figure 7-8: Comparison of numbers of edges per connection request used

As shown in the figure for all algorithms, when the number of destinations increases, more edges are consumed. We can see that the optimal trail routing derived by ILP consumes the least number of edges. Our proposed algorithm NATR uses fewer edges than the best result

of MDT. The difference between NATR and the best result of MDT becomes larger as the number of destinations increases. In this case, the NATR algorithm achieves more gains by connecting destinations one by one in a trail than rerouting a big tree as MDT does. On the average, NATR has 23% lower value than the best result of MDT, and has only 8% higher value than the optimal trail routing result. These simulation results show that our proposed NATR algorithm is close to the optimal results and shows significantly better performance than the MDT algorithm.

7.6 Heuristic Approach

A trail in the logical layer can be denoted as a LOHT, which is referred to as $\{s; D; a\}$, where s is the source node of the trail, D is the set of nodes which use receiving electronic ports to receive the tapped/dropped traffic for local stations, and a is the available bandwidth of the trail. A multicast session may be constructed by multiple trails, and they connect each other to form a bigger multicast session with O-E-O conversions. Motivated by the observations made from the trail-based ILP optimization approach in Section 7.4, we propose a heuristic algorithm, called Multicast Trail Grooming (MTG), to efficiently deal with the multicast traffic grooming problem in large TaC networks. The idea of the algorithm is to groom traffic to multiple small trails (with a few destinations) to form a larger multicast session. We observe that this indeed increases the resource utilization leading to a lower network cost. The flow chart of the MTG algorithm is given in Figure 7-9 (The pseudocode of the MTG algorithm is given in the Appendix L).

In the MTG algorithm, we first construct a trail for every connection request with a full wavelength capacity requirement as this is the optimal way to support these connection

requests. Since the complexity of constructing a trail is $O(|V|^3 \log |V|)$, the complexity of this step is $O(|R||V|^3 \log |V|)$.

Figure 7-9: Flow chart of Multicast Trail Grooming (MTG) algorithm

Then, the remaining connection requests are sorted in ascending order of destination size. The algorithm gives preference to smaller connection requests because smaller connection requests are more likely to be shared by other connection requests. For each connection

request, the algorithm selects existing trails to groom the traffic, where the destinations of the selected trail has to be a subset of the destinations to be reached, and has enough available bandwidth. The complexity of selecting existing trails is $O(|R||V|^2)$ as the upper bound of the number of existing trails is $|R||V|$ and the number of trails selected is at most $|V|$.

Finally, NATR algorithm is called if necessary, to build a new trail from the source node to those destinations which are not accommodated. The complexity of this step is $O(|V|^3 \log |V|)$. The total complexity of MTG algorithm is therefore $O(R^2|V|^2 + |R||V|^3 \log |V|)$.

To reduce the total number of wavelengths used, the link cost would be changed during the operation. If an edge has been traversed, the cost of the edge will be increased by 1. With this method, the routings of trails are more evenly distributed in the optical layer and should help in reducing the number of wavelengths used.

7.7 Numerical Results

In this section, we use simulations to show that our proposed heuristic algorithm, MTG, performs close to the optimal result of multicast grooming in TaC networks (derived by the ILP formulation). It may be noted that the proposed trail-based ILP formulation performs almost as well as the light-tree based ILP formulation except that a few more wavelengths are needed because of the long routes of trails. We first compare the performances, in terms of cost, total numbers of higher layer electronic ports, transmitting electronic ports and receiving electronic ports, and wavelengths, of the three methods for the six-node network of Figure 7-5. We then study the performance of the MTG algorithm for the larger NSFNET network of Figure 3-11.

7.7.1 Results of the Six-node Network Topology

Ten multicast connection requests are randomly generated in each simulation experiment. The source, the multicast destination size and the destination nodes are randomly generated as before in section 7.4. The required bandwidth of a connection request is still a random integer from 1 to C , where C equals OC-12. The results are shown in Figures 7-10 to 7-14, where the values at each instance are the averages over 20 simulation experiments.

Figure 7-10: Cost comparison

In Figure 7-10, the network costs of three methods are compared. Given the numbers of higher layer electronic ports and wavelengths, the network cost can be calculated by Equation (7.1). The ILP formulation based on light-tree has the best performance, followed by the ILP formulation based on trail and MTG algorithm. The cost value of the trail-based ILP formulation is averagely about 0.8% higher than that of the light-tree based one, which is due to the more wavelengths usage. Specifically, the two ILP formulations result in the same number of higher layer electronic ports (shown in Figure 7-11), but the trail-based ILP

formulation uses more wavelengths (shown in Figure 7-14). The trail-based ILP formulation achieves the performance very close to the light-tree based formulation, and network nodes in the former case do not need to have multicast capability, which can save the huge cost of deploying multicast capable devices. The cost value of MTG is quite close to the optimal value derived by the trail-based ILP formulation. It is about 8% higher than the ILP result averagely through 20 instances.

Figure 7-11: Comparison of numbers of higher layer electronic ports used

In Figure 7-11, the numbers of higher layer electronic ports are compared. As discussed before, the number of higher layer electronic ports is the summation of transmitting electronic ports and receiving electronic ports. It is noted that the number of higher layer electronic ports of the trail-based ILP formulation is exactly the same as that of the light-tree based ILP formulation, this is because the solution obtained by the light-tree based ILP formulation can also be achieved by the trail-based ILP formulation in a different form with some additional cost of wavelengths usage. We also see that the number of electronic ports of MTG is higher (~8%) than that of either formulation.

Figure 7-12: Comparison of numbers of transmitting electronic ports used

In Figure 7-12, the numbers of transmitting electronic ports are compared. The ILP formulation based on light-tree has the largest value, followed by the trail-based ILP formulation and MTG. It is very interesting to see that the trail-based ILP formulation uses fewer ports for transmitting than the light-tree based ILP formulation, while two formulations use the same number of higher layer electronic ports. The reason may be that, among the solutions with the same amount usage of electronic ports, the trail-based ILP formulation tries to select the solution that minimizes the usage of wavelengths, so longer trails may be preferred as they can travel more destinations for multiple connection requests, otherwise, more short trails have to be built which may incur higher possibilities that multiple trails traverse same links, and this will increase the usage of wavelengths. So fewer trails would be built which use fewer transmitting electronic ports. It is noted that even though the light-tree based ILP formulation use more transmitting electronic ports, the wavelength usage is the lowest as shown in Figure 7-14. This is because with multicast capable nodes, light-trees can be more efficiently routed to reduce the wavelength usage than the trail style. In Figure 7-12, we also see that MTG algorithm has the lowest value. The reason may be that in the MTG

algorithm, sub-wavelength connection requests are accommodated by the ascending order of destination size, this would increase the sharing of trails and also the sharing of transmitting electronic ports, which reduces the usage.

Figure 7-13: Comparison of numbers of receiving electronic ports used

In Figure 7-13, the numbers of receiving electronic ports are compared. The MTG algorithm has the largest value, followed by the trail-based ILP formulation and the light-tree based ILP formulation. As the light-tree based ILP formulation and the trail-based ILP formulation use the same number of higher layer electronic ports to accommodate connection requests, so the summations of the number of transmitting electronic ports in Figure 7-12 and the number of receiving electronic ports in Figure 7-13 for two different formulations should have the same value. From Figures 7-12 and 7-13, it is clear that the grades of the trail-based ILP formulation and MTG are largely different between transmitting and receiving ports usages. The MTG algorithm uses a slightly fewer transmitting electronic ports than the trail-based formulation in Figure 7-12, but with much more receiving electronic ports consumption in Figure 7-13, which turns out to use more electronic ports. The reason may be

that the sharing of trails can improve the utilization of transmitting ports, but MTG algorithm would consume more receiving ports as a trail has only one source node but multiple destination nodes, and the trail-based ILP method can balance the utilization of transmitting and receiving electronic ports more efficiently to get the optimal result with its vast computation.

Figure 7-14: Comparison of numbers of wavelengths used

In Figure 7-14, the wavelength usages are compared. The value of the light-tree based ILP formulation is lowest, followed by the trail-based ILP formulation and MTG. The best performance of the light-tree based ILP formulation is at the cost of deploying multicast capable devices at network nodes, which can also cause other problems mentioned before. The average difference of two ILP formulations is less than one wavelength, and it is noted that the cost of wavelength contributes much less (2% - 4%) than the cost of higher layer electronic ports (96% - 98%) which has the same value in the two ILP formulations. In the figure, MTG has higher wavelength usages than the trail-based ILP formulation by 17%, but as the small contribution of the wavelengths to the network cost, the cost difference between

two methods is largely depend on the higher layer electronic ports, which finally MTG has about 8% higher network cost than the trail-based formulation.

From these simulations, we can see that the trail-based ILP formulation performs close to the light-tree based ILP formulation with only a few more wavelengths usage, but without the need of deploying multicast capable devices in network nodes, and our proposed heuristic algorithm, MTG, performs close to the trail-based ILP formulation.

7.7.2 Results of NSFNET Topology

These simulations are essentially similar to the simulations of the six-node test network except that here we consider a larger network, i.e. NSFNET with 14 nodes (Figure 3-11). Multicast connection requests are randomly generated for three different scenarios: *i*) the multicast destination size ranging from 1 to 13, *ii*) the multicast destination size ranging from 1 to 9, and *iii*) the multicast destination size ranging from 7 to 13. Therefore, the average destination sizes of three scenarios are 7, 5 and 10, respectively. We assume that the bandwidth required by a multicast connection request is still randomly chosen between 1 and C , where C equals OC-48. 100 multicast connection requests are generated in each experiment and 25 experiments are simulated to obtain the average value as shown in Table 7-5.

Table 7-5: Simulation results in NSFNET

Scenarios	A	B	Electronic ports	φ
Destination Size (1-13)	99.24	453.52	552.76	24.8
Destination Size (1-9)	98.8	365.08	463.88	22.04
Destination Size (7-13)	99.32	694.24	793.56	32.12

In the table, three different scenarios have similar numbers of transmitting electronic ports A , but significantly different usage of receiving electronic ports B and wavelengths ϕ . This is because three scenarios have the same number of connection requests and the same distribution of required bandwidth, but different average size of destination sets (7, 5, and 10). That is, the effective loads of the network are different for the three scenarios, which are of the proportion of 7:5:10. We can verify this proportion from the usages of the receiving electronic ports for the three scenarios, more destination nodes result in more usage of the receiving electronic ports. The numbers of transmitting electronic ports for the three scenarios are about the same, this is because they have same number of connection requests and the sharing of trails contributes more to the receiving electronic port utilization not transmitting electronic port utilization. The total numbers of higher layer electronic ports are different mainly due to the different usage of receiving electronic ports, and the trend is, the larger multicast size, the higher is the usage of electronic ports. The number of wavelengths is highly related to the effective load of the network, a larger destination size may result in a heavier network load, so more wavelengths are needed to accommodate the traffic.

7.8 Summary

We have considered the multicast traffic grooming problem in tap-and-continue (TaC) WDM networks. We have proposed a new low-complexity node architecture with TaC mechanism. With an objective of minimizing the cost associated with the number of higher layer electronic ports and the number of wavelengths used, we have proposed a trail-based ILP formulation for multicast traffic grooming in TaC networks. The results of the trail-based ILP formulation have revealed that small trails can connect each other to form a larger multicast session which improves the utilization of the resources and reduces the cost of the network.

The trail routing is also investigated. We have proposed a new trail routing algorithm with polynomial complexity, namely Node Adding Trail Routing (NATR), to route a trail with shorter length than others. Our study has shown that NATR performs close to the optimum. We have also developed a heuristic multicast traffic grooming algorithm based on trails - Multicast Trail Grooming (MTG), which aims to construct big multicast sessions with multiple small trails, and increase the sharing of the trails. We have compared the performance of the proposed trail-based ILP formulation, the light-tree based ILP formulation and the MTG algorithm in a small test network. Results have demonstrated that: the trail-based ILP formulation performs very close to the light-tree based ILP formulation with a little more cost on wavelengths, but the trail-based network design does not need to deploy multicast capable devices in the network, which can save a lot on network cost; and the proposed heuristic algorithm MTG performs close to the trail-based ILP optimal result. To demonstrate the scalability of our heuristic algorithm, we have studied its performance in a larger network, NSFNET. Our results have shown that the MTG algorithm can scale to a network of practical size with hundreds of connection requests.

Chapter 8

Conclusions and Future Research

8.1 Summary

The ever-increasing popularity and traffic volume of multicast applications motivates the need for development of methodologies for traffic management and network design that especially cater for multicast traffic. Multicast traffic grooming is widely believed to be the key technology for addressing the disparity between the bandwidth offered by a wavelength and the bandwidth required by a single multicast connection, which makes the utilization of network resources more efficient. Unlike one-to-one unicast traffic grooming, where lightpaths are used to support connections, light-tree naturally supports one-to-many multicast connection with much higher efficiency than lightpath. The problem of multicast traffic grooming based on light-tree attracts much interest of research due to their cost-effective and efficient provisioning of multicast applications.

This thesis focuses on multicast traffic grooming research in WDM networks in terms of light-tree design and optimal analysis. This section summarizes the main works carried out in this thesis.

In Chapter 3, two new grooming schemes that lead to efficient resource utilization in WDM networks have been provided. They are called Light-Tree Division - Destination

Branch Node based Grooming scheme (LTD-DBNG) and Light-Tree Division - Adjacent Node Component based Grooming scheme (LTD-ANCG). Both of schemes are based on the idea of dividing a light-tree into smaller sub-light-trees. LTD-DBNG divides light-trees at destination branch node which is a node that forwards optical signal and also is a destination node. Dividing light-trees at destination branch nodes would save on drop ports as optical to electrical conversion is anyway needed at these nodes. LTD-ANCG divides light-trees into adjacent node components which are light-trees within two optical hops. LTD-DBNG and LTD-ANCG improve the efficiency of resource utilization and also lower the optical-electronic-optical (O-E-O) conversion overhead. Computer simulations are used to evaluate the performance of these schemes. Simulations have demonstrated that compared with existing algorithms, these schemes significantly reduce the request blocking probability but can be implemented with very reasonable electronic processing, with LTD-ANCG performing better than LTD-DBNG but with greater complexity. The blocking probabilities of these schemes considering variations in the add/drop ratio have also been evaluated, and the results have demonstrated that a proper choice of this ratio will provide a target blocking probability with low network costs.

To further improve the utilization of network resources when add/drop ratio is low, multicast traffic grooming with leaking strategy is investigated in Chapter 4. Multicast traffic grooming with leaking would allow a light-tree delivers the traffic of a multicast connection to nodes which are not in the destination set of the connection. The leaking strategy improves the sharing of light-trees and add/drop ports, leading to lower blocking ratios. Two multicast traffic grooming algorithms with leaking strategy, namely, multicast traffic leaky grooming (MTLG) and multicast traffic hybrid grooming (MTHG), have been proposed. MTLG grooms traffic to light-trees if the traffic leaked is less than a given threshold value. MTHG first grooms traffic to light-trees without leaking; if some destinations remain, it then grooms

traffic to light-trees with leaking. MTHG is an improvement over MTLG as it can attain higher light-tree sharing with less traffic leaked. Simulations have shown that the two proposed algorithms perform better than other algorithms at low add/drop port ratios with MTHG showing better performance.

Chapter 5 investigates the optimal design of network based on light-tree with multicast traffic grooming. An optimal cost design of WDM networks based on light-tree has been provided. In particular, a light-tree based Integer Linear Programming (ILP) formulation is proposed to minimize the network cost associated with the number of higher layer electronic ports and the number of wavelengths used. Since solving the ILP formulation is time consuming for large networks, a heuristic algorithm, called sub-light-tree saturated grooming (SLTSG) is proposed to achieve scalability. This algorithm tries to construct sub-light-trees which can be fully utilized. Simulations are conducted on several networks to compare the design cost and the required number of electronic ports and wavelengths. The results have demonstrated significant benefits of using a light-tree based design over a design that only uses lightpaths.

To improve the scalability of the analytical model in Chapter 5, Chapter 6 investigates the hop constrained light-trees in multicast traffic grooming. In particular, an ILP formulation is proposed for optimal assignments of hop constrained light-trees for multicast connections so that network throughput can be maximized. Hop constrained light-trees improve the scalability of the approach reducing the search space of the ILP formulation. A heuristic algorithm, called Dividable Light-Tree Grooming (DLTG) algorithm, with a polynomial complexity is also proposed. This algorithm is based on grooming traffic to constrained light-trees and also divides a light-tree to smaller constrained light-trees on which traffic is groomed for better resource utilization. Simulations have shown that the proposed DLTG

heuristic performs better than other algorithms. It achieves network throughputs which are very close to the ILP formulation results, but with far less running times.

In Chapter 7, multicast traffic grooming in Tap-and-Continue (TaC) WDM networks is investigated. Since using light-trees may lead to some serious negative side-effects because of light splitting, a network node with TaC devices can tap a small amount of incoming optical power for the local station while forwarding the remainder to an output is worth to investigated. A simple and efficient node architecture with the TaC mechanism is proposed. An ILP formulation with the objective of minimizing the network cost in terms of the number of higher layer electronic ports and the number of wavelengths used is also proposed. A heuristic algorithm of polynomial complexity, called Multicast Trail Grooming (MTG), is proposed for use in large networks. Simulation results have shown that although the proposed ILP formulation does not require multicast capable nodes, its performance is still very similar to that of a light-tree based ILP formulation where network nodes have multicast capability. The solution obtained by the MTG algorithm is also close to the ILP optimal solution and is shown to work efficiently for typical network topologies like NSFNET.

8.2 Recommendations for Future Research

This thesis mainly contributes to the development of multicast traffic grooming algorithms and evaluation of their respective performance in WDM networks under both static and dynamic traffic scenarios. In the following, we recommend some research topics for the future work.

Chapters 3 and 4 deal with the light-tree based multicast traffic grooming problem with dynamic traffic. From the results of the simulation, we can see that dividing light-tree into small ones can improve the network performance so we should develop more efficient

algorithms to divide light-trees to facilitate the multicast traffic grooming and keep the overhead of multicasting at a low level. In the study of dynamic traffic scenarios, some constraints like limited add/drop ports in a node, sparse multicast capable nodes, or limited number of splitters in multicast capable nodes need to be considered as these constraints are likely to exist in networks. If the number of add/drop ports is limited, it may not be possible to build a new lightpath or light-tree even if there are spare wavelengths. With sparse multicast capable nodes, or limited number of splitters in multicast capable nodes, and as the traffic duplication in optical domain is needed at branch nodes of light-trees, some light-trees cannot be set up properly. As the routing methods would be different and may be improved to adapt the resource constraints, new multicast traffic grooming schemes need to be developed to make resources be efficiently used. For example, in a network with sparse multicast capable nodes, several light-trees (which form a light-forest) would be built to accommodate a multicast session, whether and where to divide the light-trees is a crucial problem because light-trees already become smaller. If the small light-trees are further divided, many O-E-O conversions would be introduced. New schemes should carefully take into account these consequences to avoid the disadvantage and extend the benefit in dividing light-trees. So, it is necessary to develop new grooming schemes that take into account resources constraints, such as wavelength limitation, sparse wavelength converters, sparse multicast capable nodes or add/drop limitation.

In Chapters 5 and 6, general frameworks of light-tree based multicast traffic grooming problem have been proposed. However, like introducing hop constrained light-trees to achieve some degree of efficiency, considerable work needs to be carried out to improve effectiveness and performance of such light-tree based multicast traffic grooming. Secondly, with different resource constraints, like an end-to-end delay, sparse multicasting nodes or sparse wavelength converters, how to make the mathematical formulation works precisely is

an important topic. And since the problem of multicast traffic grooming is NP-complete, some efficient heuristic approaches need to be developed. To verify the validity of heuristic approaches, the mathematical ILP results can be used to compare with the heuristic results when using small networks.

In Chapter 7, preliminary research about multicast traffic grooming in TaC networks has been carried out. There is a need to fully investigate the implementation, complexity and algorithms of multicast traffic grooming in such networks. The existing networks are mainly deployed for one-to-one unicast, it is costly to upgrade the network infrastructure to support multicasting and multicast grooming. For multicast traffic grooming in TaC networks, future works like cost-effective grooming node architectures and grooming algorithms are especially important for the practical technology to be deployed.

Author's Publications

Journal Papers

- [1] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Light-tree configuration for multicast traffic grooming in WDM mesh networks," *Photonic Network Communications*, vol. 20, no. 2, pp. 151-164, 2010. (related to Chapter 3)
- [2] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Leaking strategy for multicast traffic grooming in WDM mesh networks," *Journal of Lightwave Technology*, vol. 30, no. 23, pp. 3709-3719, 2012. (related to Chapter 4)
- [3] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Design of WDM networks with multicast traffic grooming," *Journal of Lightwave Technology*, vol. 29, no. 16, pp. 2337-2349, 2011. (related to Chapter 5)
- [4] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Constrained light-tree design in WDM mesh networks for multicast traffic grooming," *Optical Switching and Networking*, vol. 10, no. 3, pp. 233-245, 2013. (related to Chapter 6)
- [5] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Multicast traffic grooming in Tap-and-Continue WDM mesh networks," *Journal of Optical Communications and Networking*, vol. 4, no. 11, pp. 918-935, 2012. (related to Chapter 7)

Conference Papers

- [1] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Dynamic sub-light-tree based traffic grooming for multicast in WDM networks," IEEE Global Telecommunications Conference (GLOBECOM), 2010. (related to Chapter 3)

- [2] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Light-tree based multicast traffic grooming in WDM mesh networks," OptoElectronics and Communications Conference (OECC), pp. 274-275, 2010. (related to Chapter 6)

- [3] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Heuristic algorithms for multicast traffic grooming in WDM mesh networks," International Conference on Information, Communications and Signal Processing (ICICS), Singapore, 2012. (related to Chapter 6)

Appendixes

A. Light-tree Division at Destination Branch Nodes (LTD-DBN)

Algorithm

INPUT: NLT_r

OUTPUT: $NSLT_r$

BEGIN

$divide_flag = \text{TURE}$;

While $divide_flag$ **do**

$one_divided = \text{FALSE}$;

//Identify where to divide

For each tree T of NLT_r

For each destination d of T

If $out_degree(d) > 0$

//Divide light-tree from d

 Divide T into two trees from d , save two trees into NLT_r , and delete T from NLT_r ;

$one_divided = \text{TRUE}$;

 BREAK; *//break inner For*

If $one_divided = \text{TRUE}$

 BREAK; *//break outer For*

If $one_divided = \text{FALSE}$

$divide_flag = \text{FALSE}$;

$NSLT_r = NLT_r$;

END

B. Light-tree Division - Adjacent Node Component (LTD-ANC)

Algorithm

INPUT: NLT_r

OUTPUT: $NSLT_r$

BEGIN

For each tree T of NLT_r

$split_flag = \text{TRUE}$;

u is the root of T ;

While $split_flag$ **do**

 //Identify where to divide

If $out_degree(u) = 1$

v is the adjacent node of u ;

$category = 4$ if $out_degree(v) = 0$;

$category = 3$ if $out_degree(v) > 0$;

If $out_degree(u) > 1$ **DO**

$category = 2$;

For each adjacent node v of u

If $out_degree(v) > 1$

$category = 1$;

BREAK;

 //Divide component out of the tree

 According to $category$ indicated, starting from u , divide the adjacent node component out of T , delete the links from T , save the component into the list $NSLT_r$.

If $num_edges(T) = 0$

$split_flag = \text{FALSE}$;

END

C. The LTD-DBN based grooming (LTD-DBNG) Algorithm

INPUT: $CN, (s; D; b)$

OUTPUT: $CN, LOHT_r$

BEGIN*STEP 1:*

Call **Logical Layer Grooming Algorithm**($CN, (s; D; b)$), returns $LOHT_r$. If all nodes of D are reached by $LOHT_r$, go to *STEP 6*, otherwise, go to *STEP 2*.

STEP 2:

Call **Build New Light-trees Algorithm**($CN, (s; D; b), LOHT_r$), returns NLT_r . If all destinations which are not honored by $LOHT_r$, can be accommodated by NLT_r , go to *STEP 3*, otherwise block request.

STEP 3:

Call **LTD-DBN Algorithm**(CN, NLT_r), returns sub-light-trees $NSLT_r$.

STEP 4:

For each tree T of $NSLT_r$

If a $LOHT$ with at least a free bandwidth b and having the same source and destinations of T , exists in logical layer of CN

 Delete T from $NSLT_r$ and add the $LOHT$ of T into $LOHT_r$.

STEP 5:

For each tree T of $NSLT_r$

 Allocate a wavelength along links of T and one add port at source of T and one drop port at each destination of T . If fail, block request.

 Add the information of the $LOHT$ of T into the logical layer of CN .

 Add the $LOHT$ of T into $LOHT_r$.

STEP 6:

For each $LOHT$ of $LOHT_r$

 Reduce b bandwidth of corresponding $LOHT$ in logical layer of CN .

END**D. Logical Layer Grooming Algorithm**

INPUT: $CN, (s; D; b)$

OUTPUT: $LOHT_r$

BEGIN

groom flag = TRUE;

While $D \neq \Phi$ && groom flag do

Among all *LOHTs* of *CN* with a free bandwidth of at least b , find the one that can honor the maximum number of nodes in D and whose set of leaves is a subset of D . If more than one *LOHT* are found, choose the one whose root is s or whose root belongs to D . If such a qualified *LOHT* cannot be found, set *groom_flag* to FALSE.

Delete the destinations of the selected *LOHT* from D . If the source of the *LOHT* is neither s nor already be reached by grooming, add the source of the *LOHT* into D .

Save this *LOHT* into $LOHT_r$.

END**E. Build New Light-trees Algorithm**

INPUT: $CN, (s; D; b), LOHT_r$

OUTPUT: NLT_r

BEGIN*STEP 1:*

$AG = \Phi$;

For each *LOHT* of $LOHT_r$

Add the source and destinations of *LOHT* into AG .

For each node u of *LOHT* destinations

Add a directed edge into AG between the source of *LOHT* and u .

STEP 2:

Running *depth_first_search(s)* algorithm in AG , input the source s of request as start node, save all nodes connected to s into set S , including s . Put all the destinations which are not accommodated by $LOHT_r$ into set D' .

STEP 3:

$AG = \Phi$;

Construct an auxiliary graph AG , whose nodes are the nodes of CN and whose links are the links of optical layer of CN which have free wavelength.

STEP 4:

Run Dijkstra's algorithm in AG to calculate shortest paths between any two nodes in optical layer of CN .

STEP 5:

While D' is not empty **do**

Select one node u in set S and one node v in set D' , make sure shortest path between (u, v) is the shortest among all pairs between set S and set D' . Add all nodes traversed by the shortest path (except u) into set S , delete v from set D' . Save shortest path (u, v) , all saved shortest paths may form several disconnected trees with different roots.

STEP 6:

Save disconnected trees with different roots into NLT_r .

END

F. Constrained Light-tree Multicast Routing (CLMR) algorithm

Input:

A network $G(V, E)$, and a multicast connection request $(s; D)$

Output:

A set of adjacent node components T , which forms a larger tree session to support the multicast connection request

Algorithm BEGIN

1. $S = \{s\}$
2. **While** $D \neq \emptyset$ **do**
//2-hop branch building
 3. Start from every node in S , to find a 2-hop branch with the maximal destination size, whose root i is in S and whose destinations D' is a subset of D
 4. **If** such a 2-hop branch exists
 5. Save this 2-hop branch into set T
 6. $D = D \setminus D'$ *//delete D' from D*
 7. $S = S \cup D'$ *//add D' into S*
 8. **Continue** *//start while loop again**//1-hop light-tree building*
 9. Start from every node in S , to find a 1-hop light-tree with the maximal destination size, whose root i is in S and whose destinations D' is a subset of D

```

10.    If such a 1-hop light-tree exists
11.        Save this 1-hop light-tree into set  $T$ 
12.         $D = D \setminus D'$ 
13.         $S = S \cup D'$ 
14.        Continue    //start while loop again
        // extension path building
15.    Check all shortest paths of node pairs between  $S$  and  $D$ , select the path  $P$  with the
        shortest length.
16.    Save the path  $P$  into set  $T$ 
17.     $S = S \cup \{\text{destination of } P\}$ 
18.    Continue    //start while loop again
19. End While
END

```

G. Multicast Traffic Leaky Grooming (MTLG) algorithm

Input:

Ex contains all existing LOHTs in the current network, a multicast connection request $(s; D; f)$, and a leaking ratio threshold b

Output:

A set of LOHTs R which is the traffic routing of the request

Algorithm BEGIN

//select existing LOHTs to groom request with leaking,

```

1.    While  $D \neq \Phi$  do
2.         $D^* = \Phi$ ;    //initialize the maximal destination set
3.        For each LOHT  $i \{s_i; D_i; a_i\}$  in  $Ex$ 
4.             $D' = D \cap D_i$ ;    Leaking ratio  $= (|D_i| - |D'|) / |D_i|$ ;
5.            If  $f > a_i \mid \text{Leaking ratio} > b$  Continue;    //start next  $i$  from the For loop
6.            If  $\{D_i \setminus D'\} \cap R \neq \Phi$  Continue;
7.            If  $|D'| > |D^*|$      $D^* = D'$ ;    //update the set
8.        End For

```

```

9.      If  $D^* == \Phi$       Break;           //end while
10.     If  $s_i \neq s \&\& s_i \notin D \&\& s_i \notin R$      $D = D \cup \{s_i\};$  //add  $s_i$  into  $D$ 
11.      $a_i = a_i - f;$    $D = D \setminus D^*;$    $R = R \cup \{\text{selected LOHT}\};$ 
12.     End While
        //Build new light-trees
13.     If  $D \neq \Phi$ 
14.     If  $f < C$ 
15.         Call CLMR algorithm to build light-trees from  $s$  to  $D$ , if it cannot build the
            trees, block the connection
16.     If  $f == C$ 
17.         Build a light-tree from  $s$  to  $D$ , if can not build the tree, block the connection
18.         Assign wavelengths, add/drop ports to light-tree(s), if cannot assign resources,
            block the connection
19.         Set available bandwidth of the light-tree(s) as  $C - f$ ,  $R = R \cup \{\text{built light-trees}\}$ 
END

```

H. Multicast Traffic Hybrid Grooming (MTHG) algorithm

Input:

Ex contains all existing *LOHTs* in the current network, a multicast connection request $(s; D; f)$, and a leaking ratio threshold b

Output:

A set of *LOHTs* R which is the traffic routing of the request

Algorithm BEGIN

//select existing LOHTs to groom traffic without leaking,

```

1.  While  $D \neq \Phi$  do
2.       $D^* = \Phi;$            //initialize the maximal destination set
3.      For each LOHT  $i \{s_i; D_i; a_i\}$  in  $Ex$ 
4.          If  $f > a_i$       Continue;    //start next  $i$  from the For loop
5.          If  $D_i \subseteq D \&\& |D_i| > |D^*|$      $D^* = D_i;$       //update the set

```

```

6.   End For
7.   If  $D^* == \Phi$    Break;      //end while
8.   If  $s_i \neq s \&\& s_i \notin D \&\& s_i \notin R$     $D = D \cup \{s_i\};$  //add  $s_i$  into  $D$ 
9.    $a_i = a_i - f;$    $D = D \setminus D^*;$    $R = R \cup \{\text{selected LOHT}\};$ 
10.  End While
      //select existing LOHTs to groom traffic with leaking
11.  While  $D \neq \Phi$  do
12.     $D^* = \Phi;$       //initialize the maximal destination set
13.    For each LOHT  $i \{s_i; D_i; a_i\}$  in  $Ex$ 
14.       $D' = D \cap D_i;$    $Leaking\ ratio = (|D_i| - |D'|) / |D_i|;$ 
15.      If  $f > a_i \mid |Leaking\ ratio| > b$    Continue; //start next  $i$  from the For loop
16.      If  $\{D_i \setminus D'\} \cap R \neq \Phi$    Continue;
17.      If  $|D'| > |D^*|$     $D^* = D';$       //update the set
18.    End For
19.    If  $D^* == \Phi$    Break;      //end while
20.    If  $s_i \neq s \&\& s_i \notin D \&\& s_i \notin R$     $D = D \cup \{s_i\};$  //add  $s_i$  into  $D$ 
21.     $a_i = a_i - f;$    $D = D \setminus D^*;$    $R = R \cup \{\text{selected LOHT}\};$ 
22.  End While
      //Build new light-trees
23.  If  $D \neq \Phi$ 
24.    If  $f < C$ 
25.      Call CLMR algorithm to build light-trees from  $s$  to  $D$ , if it cannot build the trees,
        block the connection
26.    If  $f == C$ 
27.      Build a light-tree from  $s$  to  $D$ , if can not build the tree, block the connection
28.      Assign wavelengths, and add/drop ports to light-tree(s), if can not assign resources,
        block the connection
29.      Set available bandwidth of the light-tree(s) as  $C - f$ ,  $R = R \cup \{\text{built light-trees}\}$ 
END

```

I. Sub-light-tree Saturated Grooming (SLTSG) algorithm

Input:

A network $G(V, E)$ with capacity C of each wavelength, and a set of multicast requests $t_1, t_2, \dots, t_T$.

Output:

(a) Set of sub-light-trees in the logical layer, (b) Routings of the multicast requests, (c) RWA of sub-light-trees in the physical layer and (d) Network Cost.

Algorithm:

1. *Step 1.* Construct a light-tree for each request whose required bandwidth is of a full wavelength capacity. If there are multiple such requests then these can be done in any order.
2. *Step 2.* Construct sub-light-trees whose capacity can be fully shared by the remaining requests (details as follows).
3. Sort the remaining requests in a descending order of size of destinations and label them as $t_1, t_2, \dots, t_n$, and the i th request denotes as $t_i = (s_i, D_i, f_i)$
4. $K = |D_1|$; *//the largest destination size*
5. **For** $r = K$ to 1 *//prefer to find a larger sub-light-tree*
6. **For** $i = 1$ to n
7. $S = \Phi$; *//set S to be an empty set*
8. $S \leftarrow t_i$; $D = D_i$; $B = f_i$; *//add t_i into S*
9. **For** $j = i+1$ to n
10. $P = D \cap D_j$ *//find common destinations*
11. **If** $|P| \geq r \ \&\& \ (B+f_j) \leq C$
12. $S \leftarrow t_j$; $D = P$; $B = B + f_j$; *//add t_j into S*
13. **If** $B = C$
14. Construct a sub-light-tree (for D) for the requests in S .
15. Update the destinations of each request in S by deleting the destinations in D of that request.
16. Go back to *step 2* to start over again until no more sub-light-tree with full bandwidth utilization can be built.
17. **End For**
18. **End For**
19. **End For**

-
20. *Step 3.* Construct sub-light-trees whose capacity may not be fully used (details as follows).
 21. Sort remaining requests in an ascending order of destination size, label them as $t_1, t_2, \dots, t_m, m \leq n$
 22. **For** $i=1, \dots, m$ //prefer to construct smaller trees first
 23. Call **Logical Layer Grooming Algorithm** for t_i
 24. Call **Build New Light-trees Algorithm** for t_i
 25. Call **LTD-ANC Algorithm**
 26. **End For**
- END**

J. Dividable Light-Tree Grooming (DLTG) algorithm

Input:

A network $G(V, E)$ with limited number of transceivers $TR(RR)$ at each node, limited number of wavelengths W each of capacity C , and a set of multicast connection requests R .

Output:

(a) A set of constrained light-trees and RWA of these light-trees, (b) Routings of the multicast connection requests $U_i, 1 \leq i \leq R$ and (c) Network throughput.

Algorithm BEGIN

1. Sort connection requests in the descending order of throughput (required bandwidth times destination size) and index them from 1 to R .
2. **For** $i = 1$ to R
//select existing light-trees to groom connection request $\{s_i; D_i; f_i\}$. Ex is the set that contains all existing light-trees, each is denoted as $\{s_j; D_j; a_j\}$, where a_j is available bandwidth of light-tree j
 4. **While** $D_i \neq \Phi$ **do**
 5. Select the existing light-tree j in Ex , which can honor the maximum number of nodes in D_i , and $D_j \subseteq D_i$ && $a_j \geq f_i$. If the tree cannot be found, **Break**
 6. **If** $s_j \neq s_i$ && $s_j \notin D_i$ && $s_j \notin U_i$
 7. $D_i = D_i \cup \{s_j\}$ *//add s_j into D_i*

```

8.       $a_j = a_j - f_i$ 
9.       $D_i = D_i \setminus D_j$       //delete  $D_j$  from  $D_i$ 
10.      $U_i = U_i \cup \{light-tree\ j\}$ 
11.     End While
        // divide existing light-trees to groom traffic
13.     While  $D_i \neq \Phi$  do
14.         Select the existing tree  $j$  in  $Ex$ , which can honor the maximum number of nodes
            in  $D_i$ , and  $a_j \geq f_i$ . If the tree cannot be found, Break
15.          $D = D_j \cap D_i$ 
16.          $D' = D_j \setminus D$ 
17.         Delete the existing tree  $j$ 
18.         Build a light-tree from  $s_j$  to  $D'$ , set available bandwidth as  $a_j$ 
19.         Build a light-tree from  $s_j$  to  $D$ , set available bandwidth as  $a_j - f_i$ 
20.          $U_i = U_i \cup \{new\ light-tree\ from\ s_j\ to\ D\}$ 
21.          $D_i = D_i \setminus D$ 
22.         If  $s_j \neq s_i \ \&\& \ s_j \notin D_i \ \&\& \ s_j \notin U_i$ 
23.              $D_i = D_i \cup \{s_j\}$       // add  $s_j$  into  $D_i$ 
24.         End While
        //build light-trees if necessary
25.     If  $D_i \neq \Phi$ 
26.         Build a light-tree from  $s_i$  to  $D_i$ , if can not build the tree, block request  $i$ 
27.         Call LTD-ANC Algorithm to divide the light-tree if  $C > f_i$ 
28.         Allocate physical resource and set available bandwidth of the light-trees as  $C - f_i$ ,
            if cannot success, block connection request  $i$ 
29.          $U_i = U_i \cup \{new\ light-trees\}$ 
30.     End For
END

```

K. Node Adding Trail Routing (NATR) algorithm**Input:**

A network $G(V, E)$, and a multicast connection request $(s; D)$

Output:

The trail T for the connection request

Algorithm:**BEGIN**

1. Among all shortest paths from s to each destination in D , select the one which traverses the largest number of destinations in D as the initial trail T .
2. $D = D - \{\text{destinations traversed by } T\}$ *//delete destinations traversed by the initial trail*
3. **While** $D \neq \Phi$ **do**
4. Generate a graph G' , which is a copy of G but excludes the edges of the trail T
5. Find shortest paths from every node of T to every node of D on G' , so that all paths are edge disjointed to the trail T .
6. Find the node d in D , which has the lowest cost to add into the trail T .
7. Add downward and upward (no upward when add to end) paths of d into the trail T .
8. $D = D \setminus d$ *//delete d from D*
9. **End While**

END**L. Multicast Trail Grooming (MTG) algorithm****Input:**

A network $G(V, E)$ with capacity C of each wavelength, and a set of multicast requests.

Output:

(a) A set of trails in the physical layer, (b) the traffic routing T_i of each multicast connection request, (c) RWA of trails in the physical layer and (d) number of electronic ports and wavelengths used and network cost.

Algorithm:**BEGIN**

-
1. Call **NATR algorithm** to construct a trail for each connection request whose required bandwidth is of a full wavelength capacity. If there are multiple such connection requests then these can be done in any order.
 2. Sort the remaining connection requests in ascending order of destination size and label them as $t_1, t_2, \dots, t_n$, and the i th connection request denotes as $t_i = (s_i; D_i; f_i)$
 3. **For** $i = 1$ to n
//select existing trails to groom connection request $(s_i; D_i; f_i)$, Ex contains all existing trails, each is denoted as a LOHT $\{s_j; D_j; a_j\}$, a_j is available bandwidth of trail j
 4. **While** $D_i \neq \Phi$ **do**
 5. Select the existing trail j in Ex , which can honor the maximum number of nodes in D_i , and with $D_j \subseteq D_i$ && $a_j \geq f_i$. If the trail cannot be found, **break**
 6. **If** $s_j \neq s_i$ && $s_j \notin D_i$ && $s_j \notin T_i$
 7. $D_i = D_i \cup \{s_j\}$ *//add s_j into D_i*
 8. $a_j = a_j - f_i$
 9. $D_i = D_i \setminus D_j$ *//delete D_j from D_i*
 10. $T_i = T_i \cup \{\text{trail } j\}$
 11. **End While**
// build a new trail for destinations which are not accommodated
 12. **If** $D_i \neq \Phi$
 13. Call **NATR algorithm** to build a new trail from s_i to D_i , set the available bandwidth of the trail as $C - f_i$, save the trail into T_i
 14. **End For**
 - END**
-

References

- [1] B. Mukherjee, *Optical WDM Networks*. Berlin, Germany: Springer-Verlag, 2006.
- [2] H. G. Weber, S. Ferber, M. Kroh, C. Schmidt-Langhorst, R. Ludwig, V. Marembert, C. Boerner, F. Futami, S. Watanabe, and C. Schubert, "Single channel 1.28 Tbit/s and 2.56 Tbit/s DQPSK transmission," in *European Conference on Optical Communication*, 2005, pp. 3-4
- [3] L. Leng, S. Stulz, B. Zhu, L. E. Nelson, B. Edvold, L. Gruner-Nielsen, S. Radic, J. Centanni, and A. Gnauck, "1.6-Tb/s (160 x 10.7 Gb/s) transmission over 4000 km of nonzero dispersion fiber at 25-GHz channel spacing," *IEEE Photonics Technology Letters*, vol. 15, no. 8, pp. 1153-1155, 2003.
- [4] B. Zhu, X. Liu, S. Chandrasekhar, D. W. Peckham, and R. Lingle, "Ultra-long-haul transmission of 1.2-Tb/s multicarrier no-guard-Interval CO-OFDM superchannel using ultra-large-area fiber," *IEEE Photonics Technology Letters*, vol. 22, no. 11, pp. 826-828, 2010.
- [5] C. A. Brackett, "Dense wavelength division multiplexing networks: principles and applications," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 6, pp. 948-964, 1990.
- [6] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Communications Magazine*, vol. 31, no. 2, pp. 78-88, 1993.
- [7] K. Iwatsuki, J. Kani, H. Suzuki, and M. Fujiwara, "Access and metro networks based on WDM technologies," *Journal of Lightwave Technology*, vol. 22, no. 11, p. 2623, 2004.
- [8] A. Carena, V. D. Feo, J. M. Finochietto, R. Gaudino, F. Neri, C. Piglione, and P. Poggiolini, "RingO: An experimental WDM optical packet network for metro applications," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 8, pp. 1561-1571, 2004.
- [9] P. Kulkarni, A. Tzanakaki, C. M. Machuka, and I. Tomkos, "Benefits of Q-factor based routing in WDM metro networks," in *ECOC*, 2005, pp. 981-982.

-
- [10] "IPTV Global Forecast - 2009 to 2013: Semiannual IPTV Global Forecast Report," Multimedia Research Group, Inc. 2009 [Online]. Available: <http://www.mrgco.com/iptv/gf1109.html>.
 - [11] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital video: an introduction to MPEG-2*: Kluwer Academic Publishers, 1996.
 - [12] I. Chlamtac, A. Ganz, and G. Karmi, "Purely optical networks for terabit communication," in *INFOCOM*, 1989, pp. 887-896.
 - [13] I. Chlamtac, A. Ganz, and G. Karmi, "Lightnet: lightpath based solutions for wide bandwidth WANs," in *INFOCOM*, 1990, pp. 1014-1021.
 - [14] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high-bandwidth optical WAN's," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171-1182, July 1992.
 - [15] L. H. Sahasrabudhe and B. Mukherjee, "Light trees: optical multicasting for improved performance in wavelength routed networks," *IEEE Communications Magazine*, vol. 37, no. 2, pp. 67-73, 1999.
 - [16] L. H. Sahasrabudhe, N. Singhal, and B. Mukherjee, "Light-trees for optical networks: Optimization problem formulation for unicast and broadcast traffic," presented at the Proc. International Conference on Communications Computers and Devices (ICCCD), Kharagpur, India, 2000.
 - [17] M. Sivakumar and K. M. Sivalingam, "Limited grooming architectures and groomer-port placement in optical WDM mesh networks," in *International Conference on Broadband Communications, Networks and Systems*, 2006, pp. 1-10.
 - [18] E. Desurvire, *Erbium-doped fiber amplifiers: principles and applications*. New York: Wiley, 1994.
 - [19] S. Banerjee and C. Chen, "Design of wavelength-routed optical networks for circuit switched traffic," in *Global Telecommunications Conference*, 1996, pp. 306-310.
 - [20] Y. D. Jin and M. Kavehrad, "Optical cross connect based on WDM and space-division multiplexing," *IEEE Photonics Technology Letters*, vol. 7, no. 11, pp. 1300-1302, 1995.
 - [21] W. D. Zhong, J. P. R. Lacey, and R. S. Tucker, "Multiwavelength cross-connects for optical transport networks," *Journal of Lightwave Technology*, vol. 14, no. 7, pp. 1613-1620, 1996.
 - [22] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 47-60, 2000.
 - [23] J. M. Yates, M. P. Rumsewicz, and J. P. R. Lacey, "Wavelength converters in dynamically-reconfigurable WDM networks," *IEEE Communications Surveys & Tutorials*, vol. 2, no. 2, pp. 2-15, 1999.
-

-
- [24] J. M. H. Elmirghani and H. T. Mouftah, "All-optical wavelength conversion: technologies and applications in DWDM networks," *IEEE Communications Magazine*, vol. 38, no. 3, pp. 86-92, 2000.
 - [25] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1061-1073, 1998.
 - [26] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 744-754, 1998.
 - [27] C. Qiao, M. Jeong, A. Guha, X. Zhang, and J. Wei, "WDM multicasting in IP over WDM networks," in *Proc. Int. Conf. Network Protocols(ICNP)*, 1999, pp. 89-96.
 - [28] X. Zhang, J. Wei, and C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting," *Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1917-1927, 2000.
 - [29] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 1, no. 1, pp. 176-186, 2009.
 - [30] R. S. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. W. A. Ayre, and W. V. Sorin, "Evolution of WDM optical IP networks: A cost and energy perspective," *Journal of Lightwave Technology*, vol. 27, no. 3, pp. 243-252, 2009.
 - [31] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy consumption in optical IP networks," *Journal of Lightwave Technology*, vol. 27, no. 13, pp. 2391-2403, 2009.
 - [32] R. Malli, X. Zhang, and C. Qiao, "Benefits of multicasting in all-optical networks," in *Proc. SPIE All Optical Networking*, 1998, pp. 209-220.
 - [33] N. K. Singhal, L. H. Sahasrabudhe, and B. Mukherjee, "Optimal multicasting of multiple light-trees of different bandwidth granularities in a WDM mesh network with sparse splitting capabilities," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 1104-1117, 2006.
 - [34] W. S. Hu and Q. J. Zeng, "Multicasting optical cross connects employing splitter-and-delivery switch," *IEEE Photonics Technology Letters*, vol. 10, no. 7, pp. 970-972, 1998.
 - [35] G. N. Rouskas, "Optical layer multicast: rationale, building blocks, and challenges," *Network, IEEE*, vol. 17, no. 1, pp. 60-65, 2003.
 - [36] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*: Elsevier Science, 1992.
 - [37] S. L. Hakimi, "Steiner's problem in graphs and its implications," *Networks*, vol. 1, no. 2, pp. 113-133, 1971.
-

-
- [38] R. M. Karp, *Reductibility among combinatorial problems: Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972.
 - [39] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms: Second Edition*, Cambridge, MA: MIT Press, 2001.
 - [40] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," *Math. Japonica*, vol. 24, no. 6, pp. 573-577, 1980.
 - [41] Y. Zhou and G. S. Poo, "Optical multicast over wavelength-routed WDM networks: A survey " *Elsevier Journal on Optical Switching and Networking*, vol. 2, no. 3, pp. 176-197, 2005.
 - [42] X. Jia, H. XiaoDong., R. Lu, and J. Sun, "Multicast routing, load balancing, and wavelength assignment on tree of rings," *IEEE Communications Letters*, vol. 6, no. 2, pp. 79-81, 2002.
 - [43] L. Du and Y. Jia, "A dynamic multicast RWA algorithm for WDM network with sparse splitting and wavelength conversion capability," in *International Conference on Electronics, Communications and Control*, 2011, pp. 613-616.
 - [44] Y. Zhou and G. S. Poo, "Multicast wavelength assignment for sparse wavelength conversion in WDM networks," in *INFOCOM*, 2006, pp. 1-10.
 - [45] B. Chen and J. Wang, "Efficient routing and wavelength assignment for multicast in WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 97-109, 2002.
 - [46] E. Modiano and P. J. Lin, "Traffic grooming in WDM networks," *IEEE Communications Magazine*, vol. 39, no. 7, pp. 124-129, 2001.
 - [47] B. Mukherjee, C. Ou, H. Zhu, K. Zhu, N. Singhal, and S. Yao, "Traffic grooming in mesh optical networks," in *Proc. IEEE OFC*, Los Angeles, CA, March 2004 (Invited Paper).
 - [48] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 285-299, 2003.
 - [49] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 122-133, 2002.
 - [50] K. Zhu and B. Mukherjee, "A review of traffic grooming in WDM optical networks: Architectures and challenges," *Optical Networks Magazine*, vol. 4, no. 2, pp. 55-64, 2003.
 - [51] K. Zhu, H. Zhu, and B. Mukherjee, "Traffic engineering in multigranularity heterogeneous optical WDM mesh networks through dynamic traffic grooming," *IEEE network*, vol. 17, no. 2, pp. 8-15, 2003.
-

-
- [52] K. Zhu, H. Zhu, and B. Mukherjee, *Traffic grooming in optical WDM mesh networks*: Springer, 2005.
 - [53] S. Zhang, D. Shen, and C.-K. Chan, "Energy-efficient traffic grooming in WDM networks with scheduled time traffic," *Journal of Lightwave Technology*, vol. 29, no. 17, pp. 2577-2584, 2011.
 - [54] G. Shen and R. S. Tucker, "Sparse traffic grooming in translucent optical networks," *Journal of Lightwave Technology*, vol. 27, no. 20, pp. 4471-4479, 2009.
 - [55] H. Alshaer and J. M. H. Elmirghani, "Multilayer dynamic traffic grooming with constrained differentiated resilience in IP/MPLS-over-WDM networks," *IEEE Transactions on Network and Service Management*, vol. 9, no. 1, pp. 60-72, 2012.
 - [56] B. Chen, W.-D. Zhong, and S. K. Bose, "A path inflation control strategy for dynamic traffic grooming in IP/MPLS over WDM network," *IEEE Communications Letters*, vol. 8, no. 11, pp. 680-682, 2004.
 - [57] W. Yao and B. Ramamurthy, "A link bundled auxiliary graph model for constrained dynamic traffic grooming in WDM mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 8, pp. 1542-1555, 2005.
 - [58] R. Berry and E. Modiano, "Reducing electronic multiplexing costs in SONET/WDM rings with dynamically changing traffic," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1961-1971, Oct. 2000.
 - [59] K. Zhu and B. Mukherjee, "On-line approaches for provisioning connections of different bandwidth granularities in WDM mesh networks," in *Proc. OFC*, Mar. 2002, pp. 549-551.
 - [60] S. Thiagarajan and A. Somani, "Traffic grooming for survivable mesh networks," in *Proc. OPTICOMM*, Aug. 2001, pp. 54-65.
 - [61] B. Chen, G. N. Rouskas, and R. Dutta, "On hierarchical traffic grooming in WDM networks," *IEEE/ACM Transactions on Networking* vol. 16, no. 5, pp. 1226-1238, 2008.
 - [62] C. Xin, "Blocking analysis of dynamic traffic grooming in mesh WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 721-733, 2007.
 - [63] C. Xin, "Computing blocking probability of dynamic traffic grooming in mesh WDM optical networks," in *Proceedings BroadNets*, 2004, pp. 190-199.
 - [64] C. Xin, C. Qiao, and S. Dixit, "Traffic grooming in mesh WDM optical networks - performance analysis," in *IEEE GLOBECOM*, 2003, pp. 3732-3736 vol.7.
 - [65] C. Xin, C. Qiao, and S. Dixit, "Traffic grooming in mesh WDM optical networks - performance analysis," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 9, pp. 1658-1669, 2004.
-

-
- [66] C. Xin and C. Qiao, "Performance analysis of multihop traffic grooming in mesh WDM optical networks," in *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, 2003, pp. 237-242.
 - [67] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network*, vol. 16, no. 6, pp. 46-56, 2002.
 - [68] X. Zhang and C. Qiao, "On scheduling all-to-all personalized connection and cost-effective designs in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 435-445, 1999.
 - [69] A. Chiu and E. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *Journal of Lightwave Technology*, vol. 18, no. 1, pp. 2-12, 2000.
 - [70] R. S. Barr and R. A. Patterson, "Grooming telecommunication networks " *SPIE Optical Networks Magazine*, vol. 2, no. 3, pp. 20-23.
 - [71] J. Wang, V. R. Vemuri, W. Cho, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multihop connections," *Journal of Lightwave Technology*, vol. 19, pp. 1645-1653, Nov. 2001.
 - [72] P. J. Wan, G. Calinescu, and O. Frieder, "Grooming of arbitrary traffic in SONET/WDM BLSRs," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1995-2003, Oct. 2000.
 - [73] C. Xin, B. Wang, X. Cao, and J. Li, "Logical topology design for dynamic traffic grooming in WDM optical networks," *Journal of Lightwave Technology*, vol. 24, no. 6, pp. 2267-2275, 2006.
 - [74] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 186-198, 2001.
 - [75] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840-851, 1996.
 - [76] Y. Xin, G. N. Rouskas, and H. G. Perros, "On the physical and logical topology design of large-scale optical networks," *Journal of Lightwave Technology*, , vol. 21, no. 4, pp. 904-915, 2003.
 - [77] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 73-89, 2000.
 - [78] B. Ramamurthy and A. Ramakrishnan, "Virtual topology reconfiguration of wavelength-routed optical WDM networks," in *GLOBECOM 2000*, pp. 1269-1275.
-

-
- [79] A. Gençata and B. Mukherjee, "Virtual-topology adaptation for WDM mesh networks under dynamic traffic," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 236-247, 2003.
- [80] H. Zang, J. P. Jue, L. Sahasrabudhe, R. Ramamurthy, and B. Mukherjee, "Dynamic lightpath establishment in wavelength routed WDM networks," *IEEE Communications Magazine*, vol. 39, no. 9, pp. 100-108, 2001.
- [81] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489-500, 1995.
- [82] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 903-908, 1996.
- [83] G. N. Rouskas, *Routing and wavelength assignment in optical WDM networks*: Wiley Encyclopedia of Telecommunications, 2001.
- [84] J. Zheng and H. T. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks," in *ICC*, 2002, pp. 2722-2726.
- [85] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville, "Routing and wavelength assignment of scheduled lightpath demands," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1231-1240, 2003.
- [86] A. E. Ozdaglar and D. P. Bertsekas, "Routing and wavelength assignment in optical networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 259-272, 2003.
- [87] A. Khalil, A. Hadjiantonis, C. M. Assi, A. Shami, G. Ellinas, and M. A. Ali, "Dynamic provisioning of low-speed unicast/multicast traffic demands in mesh-based WDM optical networks," *Journal of Lightwave Technology*, vol. 24, no. 2, pp. 681-693, 2006.
- [88] A. Khalil, A. Hadjiantonis, G. Ellinas, and M. Ali, "Sequential and hybrid grooming approaches for multicast traffic in WDM networks," in *Proc. IEEE Globecom*, 2004, pp. 1808-1812.
- [89] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, and M. A. Ali, "On multicast traffic grooming in WDM networks," in *Proc. IEEE International Symposium on Computers and Communications*, June 2004, pp. 282-287.
- [90] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, N. Abdellatif, and M. A. Ali, "Multicast traffic grooming in WDM networks," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, May 2004, pp. 785-788.
- [91] C. Lu, X. Nie, S. Wang, and L. Li, "Efficient dynamic multicast traffic grooming algorithm on WDM networks," in *Proc. SPIE*, 2005, pp. 602230.1-602230.10.
-

-
- [92] X. Huang, F. Farahmand, and J. P. Jue, "Multicast traffic grooming in wavelength-routed WDM mesh networks using dynamically changing light-trees," *Journal of Lightwave Technology*, vol. 23, no. 10, pp. 3178-3187, 2005.
- [93] S. Lee, H. Yen, and A. Chen, "Traffic grooming for IP multicast over WDM networks using light-path and light-tree schemes," in *Proc. of International Conference on Networks (ICN)*, 2010, pp. 291-293.
- [94] L. Guo, X. Wang, J. Cao, W. Hou, and L. Pang, "Multicast grooming algorithm in waveband switching optical networks," *Journal of Lightwave Technology*, vol. 28, no. 19, pp. 2856-2864, Oct. 2010.
- [95] W. Hou, L. Guo, J. Cao, J. Wu, and L. Hao, "Green multicast grooming based on optical bypass technology," *Optical Fiber Technology*, vol. 17, pp. 111-119, 2011.
- [96] G. V. Chowdhary and C. S. R. Murthy, "Dynamic multicast traffic engineering in WDM groomed mesh networks," in *Proc. IEEE Broadnets*, Oct. 2004.
- [97] B. Chen, S. Bose, and W. D. Zhong, "Applying saturated cut method in dynamic multicast traffic grooming in IP/MPLS over WDM mesh networks," in *Conf. on Information Commun. and Signal Processing*, 2005, pp. 1207-1211.
- [98] L. Liao, L. Li, and S. Wang, "Dynamic multicast traffic grooming in WDM mesh networks," in *Next Generation Internet Design and Engineering*, 2006, pp. 366-370.
- [99] L. Liao, L. Li, S. Wang, and H. Wen, "Traffic grooming algorithm for multicast sessions in WDM optical networks," in *Conference on Communications, Circuits and Systems Proceedings*, 2006, pp. 1913-1917.
- [100] J. E. Sierra, L. F. Caro, F. Solano, J. L. Marzo, R. Fabregat, and Y. Donoso, "Dynamic unicast/multicast traffic grooming using S/G light-tree in WDM networks," in *Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 2008, pp. 263-268.
- [101] N. Sreenath, B. Palanisamy, and S. R. Nadarajan, "Grooming of multicast sessions in sparse splitting WDM mesh networks using virtual source based trees," in *International Conference on Systems and Networks Communications*, 2006, pp. 11-11.
- [102] B. Chen, W. D. Zhong, S. Bose, and Y. H. Jin, "Dynamic multicast traffic grooming in IP/MPLS over WDM mesh networks," in *Conf. on Optical Internet (COIN)*, Chongqing, China, 2005, pp. 184-189.
- [103] S. Iyer and S. P. Singh, "A novel hybrid node architecture for reducing time delay in Wavelength Division Multiplexed (WDM) translucent network," in *National Conference on Communications (NCC)*, 2012, pp. 1-5.
- [104] C. N. Jessop, R. B. Jenkins, and R. J. Voigt, "Routing in an optical local area network using wavelength conversion," in *IEEE Conference Avionics Fiber-Optics and Photonics*, 2006, pp. 24-25.
-

-
- [105] E. C. Tien and G. Mohan, "Differentiated QoS routing in GMPLS-based IP/WDM networks," in *GLOBECOM*, 2002, pp. 2757-2761.
- [106] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz, "Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios," in *14th Conference on Optical Network Design and Modeling (ONDM)*, 2010, pp. 1-6.
- [107] W. Van Heddeghem, M. De Groote, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester, "Energy-efficiency in telecommunications networks: link-by-link versus end-to-end grooming," in *14th Conference on Optical Network Design and Modeling (ONDM)*, 2010, pp. 1-6.
- [108] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz, "Dynamic routing at different layers in IP-over-WDM networks—maximizing energy savings," *Optical Switching and Networking*, vol. 8, no. 3, pp. 181-200, 2011.
- [109] P. Wiatr, P. Monti, and L. Wosinska, "Green lightpath provisioning in transparent WDM networks: pros and cons," in *IEEE 4th International Symposium on Advanced Networks and Telecommunication Systems (ANTS)*, 2010, pp. 10-12.
- [110] S. Zhang, D. Shen, and C.-K. Chan, "Energy efficient time-aware traffic grooming in wavelength routing networks," in *GLOBECOM*, 2010, pp. 1-5.
- [111] R. Ul-Mustafa and A. E. Kamal, "Design and provisioning of WDM networks with multicast traffic grooming," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, pp. 37-53, Apr. 2006.
- [112] A. E. Kamal and R. Ul-Mustafa, "Multicast traffic grooming in WDM networks," in *Proc. Optical Networking and Communications (OptiComm)*, Dallas, TX, Oct. 2003, pp. 25-36.
- [113] H.-H. Yen, S. S. W. Lee, and B. Mukherjee, "Traffic grooming and delay constrained multicast routing in IP over WDM networks," in *IEEE International Conference on Communications.*, 2008, pp. 5246-5251.
- [114] Y. Zhu, Y. Jin, W. Sun, W. Guo, W. Hu, W. Zhong, and M. Wu, "Multicast flow aggregation in IP over optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 5, pp. 1011-1021, 2007.
- [115] D. N. Yang and W. Liao, "Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks," in *INFOCOM*, San Francisco, 2003, pp. 32-41.
- [116] A. Billah, B. Wang, and A. Awwal, "Multicast traffic grooming in WDM optical mesh networks," in *Proc. IEEE GLOBECOM*, 2003, pp. 2755-2760.
- [117] G. V. Chowdhary and C. S. R. Murthy, "Grooming of multicast sessions in WDM mesh networks," in *First Annual International Conference on Broadband Networks*, San Jose, California, 2004.
-

-
- [118] H. Madhyastha, G. Chowdhary, N. Srinivas, and C. Siva Ram Murthy, "Grooming of multicast sessions in metropolitan WDM ring networks," *Computer Networks*, vol. 49, no. 4, pp. 561–579, 2005.
- [119] H. V. Madhyastha, N. Srinivas, G. V. Chowdhary, and C. S. R. Murthy, "Grooming of multicast sessions in WDM ring networks," in *Proc. SPIE Opticomm*, Oct. 2003.
- [120] D. Buchfuhrer, T. Carnes, B. Tagiku, E. Celis, and R. Libeskind-Hadas, "Traffic grooming for single-source multicast communication in WDM rings," in *IEEE International Conference on Communications (ICC)*, 2005, pp. 1613–1619.
- [121] A. Rawat, R. La, S. Marcus, and M. Shayman, "Grooming multicast traffic in unidirectional SONET/WDM rings," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 70–83, 2007.
- [122] S. Huang, R. Dutta, and G. N. Rouskas, "Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, pp. 66–82, Apr. 2006.
- [123] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Light-tree configuration for multicast traffic grooming in WDM mesh networks," *Photonic Network Communications*, vol. 20, no. 2, pp. 151–164, 2010.
- [124] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Dynamic sub-light-tree based traffic grooming for multicast in WDM networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.
- [125] Q. Huang and W. D. Zhong, "Traffic performance evaluation of an optical packet switch with multicast operation," *IEEE Communications Letters*, vol. 12, no. 12, pp. 894–896, 2008.
- [126] P. Wiatr, P. Monti, and L. Wosinska, "Power savings versus network performance in dynamically provisioned WDM networks," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 48–55, 2012.
- [127] W. D. Zhong, X. Niu, B. Chen, S. K. Bose, and Y. Jin, "Performance comparison of overlay and peer models in IP/MPLS over optical networks," *Photonic Network Communications*, vol. 9, no. 1, pp. 121–131, 2005.
- [128] 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, 2003.
- [129] O. Gerstel, P. Lin, and G. Sasaki, "Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings," in *INFOCOM*, 1998, pp. 94–101.
- [130] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Leaking strategy for multicast traffic grooming in WDM mesh networks," *Journal of Lightwave Technology*, vol. 30, no. 23, pp. 3709–3719, 2012.
- [131] *ILOG CPLEX*, *ILOG, Inc., Mountain View, CA*[Online]. Available: <http://www.ilog.com/products/cplex/>.
-

-
- [132] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Design of WDM networks with multicast traffic grooming," *Journal of Lightwave Technology*, vol. 29, no. 16, pp. 2337-2349, 2011.
- [133] N. K. Singhal, L. H. Sahasrabudde, and B. Mukherjee, "Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks," *Journal of Lightwave Technology*, vol. 21, no. 11, pp. 2587-2594, 2003.
- [134] M. Herzberg, S. J. Bye, and A. Utano, "The hop-limit approach for spare-capacity assignment in survivable networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 775-784, Dec. 1995.
- [135] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Constrained light-tree design in WDM mesh networks for multicast traffic grooming," *Optical Switching and Networking*, vol. 10, no. 3, pp. 233-245, 2013.
- [136] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Light-tree based multicast traffic grooming in WDM mesh networks," in *Optoelectronics and Communications Conference (OECC)*, 2010, pp. 274-275.
- [137] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Heuristic algorithms for multicast traffic grooming in WDM mesh networks," in *International Conference on Information, Communications and Signal Processing (ICICS)*, Singapore, 2012.
- [138] S. Yan, J. S. Deogun, and M. Ali, "Routing in sparse splitting optical networks with multicast traffic," *Computer Networks*, vol. 41, pp. 89-113, 2003.
- [139] F. Zhou, M. Molnar, and B. Cousin, "Avoidance of multicast incapable branching nodes for multicast routing in WDM networks," *Photonic Network Communications*, vol. 18, no. 3, pp. 378-392, 2009.
- [140] F. Zhou, M. Molnar, and B. Cousin, "Hypo-Steiner heuristic for multicast routing in all-optical WDM mesh networks," *Photonic Network Communications*, vol. 20, no. 1, pp. 33-42, 2010.
- [141] M. Ali and J. S. Deogun, "Cost-effective implementation of multicasting in wavelength-routed networks," vol. 18, no. 12, pp. 1628-1638, 2000.
- [142] R. Lin, W.-D. Zhong, S. Bose, and M. Zukerman, "Multicast traffic grooming in Tap-and-Continue WDM mesh networks," *Journal of Optical Communications and Networking*, vol. 4, no. 11, pp. 918-935, 2012.
- [143] P. Prucnal, E. Harstead, and S. Elby, "Low-loss, high-impedance integrated fiber-optic tap," *Optical Engineering*, vol. 29, pp. 1136-1142, Sept. 1990.
- [144] <http://www.thorlabs.hk/thorproduct.cfm?partnumber=FCMM50-99A>.
-