Link-state update in wavelength-routed networks

Shu, Shen

2006

Shu, S. (2006). Link-state update in wavelength-routed networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3440>

<https://doi.org/10.32657/10356/3440>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:03:26 SGT

LINK-STATE UPDATE IN WAVELENGTH-ROUTED NETWORKS

Shen Shu

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2005

Copyright © 2005 by Shen Shu

All rights reserved.

To my parents

Acknowledgment

This thesis concludes my study at Nanyang Technological University. I would not have gone so far without the support given by many individuals.

Any success I achieved in my life I attribute to my parents Ligang and Manqin. They have done enormously to give me a great life and education. I thank my sister Xin, who has always been supportive, helpful, and inspiring.

My supervisor, Assistant Professor Xiao Gaoxi, have continuously motivated and guided me from the beginning of my academic research. I am very grateful to him for giving me the right mix of guidance and freedom. I also thank Associate Professor Cheng Tee Hiang for advising selflessly during the past several years.

I gratefully acknowledge financial support of my research from the Nanyang Technological University and the Network Technology Research Centre. I thank Associate Professor Shum Ping, Director of Network Technology Research Centre, for creating an outstanding environment in which I was provided with numerous resources. I am also grateful to many Network Technology Research Centre members for their friendship and advice.

Finally, I thank my loving fiancée Yin, who has been supporting me with great patience for the past five years. Her love and care can never be more comforting, and her passion for life sparks mine. Having her by my side makes this all the more meaningful.

Summary

Optical networking has been the mainstream technology for backbone networks in offering huge bandwidth and great reliability. Wavelength division multiplexing (WDM) technology further improves the utilization of optical fibers significantly by multiplying the capacity of optical fiber strands. Incoming customer signal to a WDM optical network is carried over specific wavelengths and multiplexed. The emerging wavelength-routed network (WRN) extends WDM technology from point-to-point connections to end-to-end connections, which are called lightpaths. WRN promises to reduce cost and support dynamic wavelength services. The link-state based routing protocol is introduced to wavelength-routed network to take on the challenge of establishing lightpath dynamically. It offers the advantages of faster convergence, lower blocking probability, and better support of traffic engineering. Nevertheless, trade-off has to be made so that the link-state update messages will not overload the control channels and cause failures.

This thesis investigates the performance versus overhead trade-off for link-state based routing in WRNs. We introduce a general model of the important parameters affecting the performance and overhead of Link-State based Lightpath Establishment (LSLE) and the fundamentals of balancing performance and overhead are investigated. We implemented the routing and signaling protocols as well as a simulator for WRNs. With an extensible and unified architecture, the simulator supports vari-

SUMMARY

ous routing and wavelength assignment (RWA) algorithms, topologies, and statistics collections.

We present the first in-depth study on the benefits of advertising wavelength availability. Extensive simulation results show that such advertisement provides significant performance gain, however, only under light traffic load. It is also shown that other network parameters, such as the density of wavelength conversion, the routing methods, or a higher degree of network connectivity, have no significant impact on the performance gain achieved by link-state advertisement. The study provides the basis of our research and justifies our objective to enhance network performance under light traffic load.

Then we evaluate existing link-state update methods for WRNs, namely periodic update and triggered update. For the first time, an analytical blocking model is proposed for periodic update and its accuracy validated by extensive simulations. The stale link-state information introduced by periodic update significantly degrades the blocking performance under light traffic loads. Four triggers of link-state update are then evaluated and we find that smaller threshold does not necessarily lead to better performance. Triggered update does not provide sufficient resolution and reliable control of the trade-off between performance and overhead.

With in-depth understanding of the existing link-state update methods, we then address the problem of how to refine the link-state update for routing lightpaths in WRNs. We propose two novel link-state update algorithms designed to enhance the performance of WRNs. The generic update rate algorithm (GURA) is able to avoid overloading control channel with excessive update messages and tolerate the inherent burstiness of link-state changes to reduce stale information. In addition to GCRA, the event-discriminating update rate algorithm (EURA), which fine-tunes the timing of link-state update, is proposed to improve further the blocking performance

SUMMARY

with even less number of link-state updates. Comprehensive performance evaluations show that the proposed algorithms successfully enforce the control bandwidth quota while offering much lower blocking probability than previously known link-state update methods.

Contents

Acknowledgment	I
Summary	II
Contents	V
List of Figures	VIII
List of Abbreviations	XI
List of Mathematical Notations	XII
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	5
1.3 Major contributions of the thesis	5
1.4 Organization of the thesis	7
2 Lightpath Establishment in All-Optical Networks	8
2.1 Overview of lightpath establishment	9
2.2 Distributed dynamic lightpath establishment	12
2.3 Control plane and its reliability	14
2.3.1 Routing protocol	14
2.3.2 Signaling protocol	15
3 Modeling Link-State Based Lightpath Establishment	17
3.1 Routing and signaling model	19
3.1.1 RWA computation	19

CONTENTS

3.1.2	Link-state update	21
3.1.3	Signaling	25
3.2	Network and traffic model	26
3.3	Evaluation metrics	28
3.4	Simulation environment	30
3.4.1	Simulator design	31
3.4.2	Features and usage	34
4	Benefits of Link-State Update	37
4.1	Performance study	38
4.1.1	Different traffic loads	42
4.1.2	Different routing and wavelength assignment methods	44
4.1.3	Different densities of wavelength conversions	46
4.1.4	Different network topologies	48
4.1.5	Segment-by-segment wavelength assignment	50
4.2	Conclusions	53
5	Periodic Link-State Update	55
5.1	Analysis of the fixed routing case	57
5.1.1	Framework of analysis	57
5.1.2	Calculating the routing blocking probability B_R^R	60
5.1.3	Calculating the setup blocking probability B_R^S	62
5.1.4	Calculating the conditional blocking probability $B_{R X_j=m}$. .	64
5.1.5	Computational complexity	65
5.2	Limited applicability to adaptive routing	66
5.3	Numerical results and discussions	66
5.3.1	PacNet	67
5.3.2	12-node ring	71
5.3.3	16-node mesh torus	71
5.3.4	Comparison of networks with different connectivities	75
5.4	Conclusions	77
6	Triggered Link-State Update	78
6.1	Link-State Update Triggers	79
6.1.1	A General Description	79
6.1.2	Specific Link-State Update Triggers	81

CONTENTS

6.1.3	Evaluating the Triggers	82
6.2	Numerical Results and Discussions	83
6.2.1	Absolute Threshold Trigger	84
6.2.2	Relative Threshold Trigger	88
6.2.3	Absolute and Relative Counter Threshold Triggers	90
6.3	Conclusions	94
7	A Novel Method of Link-State Update	95
7.1	A novel link-state update method	97
7.2	Performance evaluation	102
7.2.1	Performance improvement over existing methods	105
7.2.2	The impacts of control bandwidth and tolerance of burstiness .	107
7.2.3	The impact of wavelength conversion availability and net- work connectivity	109
7.3	Conclusions	112
8	Conclusion	113
8.1	Research Contributions	114
8.2	Future Research Directions	116
	Author's Publications	118
	Bibliography	118
	Appendix	124
A	Network Topologies	124

List of Figures

3.1	A typical IP packet format to carry the wavelength availability opaque LSA.	22
3.2	Illustration of the link-state update methods and the concepts of pseudo-surplus and pseudo-deficient information.	23
3.3	Sequence diagram of lightpath establishment signaling procedures. . .	27
3.4	Architecture of the simulator	32
4.1	Illustration of source wavelength assignment and segment-by-segment wavelength assignment.	40
4.2	Performance for Fixed Shortest-Path (FSP) routing with SWC, PacNet, and $C = 8$	42
4.3	Performance gains for different routing methods with SWC, PacNet, and $C = 8$	43
4.4	Performance comparison of first-fit, random, and weighted wavelength assignment methods. With sparse wavelength conversion (SWC), Shortest-First Least-Congested-Path (SF-LCP) routing, PacNet, and 8 wavelengths per link.	45
4.5	Performance for different wavelength conversions with SF-LCP routing, PacNet, and $C = 8$	46
4.6	Performance gains for different wavelength conversions with SF-LCP routing, PacNet, and $C = 8$	47
4.7	Performance gains for different network topologies with SWC, SF-LCP routing, and $C = 8$	48
4.8	Performance gains for different numbers of wavelengths per link with SWC, SF-LCP routing, and PacNet. The x-axis is zoomed out by 4 and 8 times for $C = 32$ and $C = 64$	49

LIST OF FIGURES

4.9	Performance comparison for FSP routing with source wavelength assignment and segment-by-segment wavelength assignment. With SWC, PacNet, and $C = 8$	50
4.10	Performance gains for different routing methods with segment-by-segment wavelength assignment. With SWC, PacNet, and $C = 8$. . .	51
4.11	Performance gains for different routing methods with segment-by-segment wavelength assignment. With FWC, PacNet, and $C = 8$. . .	52
5.1	Results of analysis and simulation for PacNet with 8 wavelengths per link. Fixed routing case.	68
5.2	Results of analysis and simulation for PacNet with 8 wavelengths per link. Adaptive routing case.	70
5.3	Results of analysis and simulation for 12-node ring network with 8 wavelengths per link. Fixed routing case.	72
5.4	Results of analysis and simulation for 12-node ring network with 8 wavelengths per link. Adaptive routing case.	73
5.5	Results of analysis and simulation for 16-node mesh torus network with 8 wavelengths per link. Fixed routing case.	74
5.6	Results of analysis and simulation for 16-node mesh torus network with 8 wavelengths per link. Adaptive routing case.	76
5.7	Comparing the blocking performances of the three networks. 8 wavelengths per link, fixed routing, simulation results only.	77
6.1	Blocking performance of Absolute Threshold Triggers (ATT), PacNet, 8 wavelengths per link.	86
6.2	Link-state update rates of Absolute Threshold Triggers (ATT), PacNet, 8 wavelengths per link.	87
6.3	Blocking performance of Relative Threshold Triggers (RTT), PacNet, 8 wavelengths per link.	91
6.4	Link-state update rates of Relative Threshold Triggers (RTT), PacNet, 8 wavelengths per link.	92
6.5	Comparisons between Absolute Counter Threshold Triggers (ACTT) and Absolute Threshold Triggers (ATT), PacNet, 8 wavelengths per link.	92
6.6	Comparisons between Relative Counter Threshold Trigger (RCTT) and Relative Threshold Triggers (RTT), PacNet, 8 wavelengths per link.	93

LIST OF FIGURES

7.1	The generic update rate algorithm (GURA)	98
7.2	Different sensitivities of the blocking performance to pseudo-surplus information and pseudo-deficient information. Results are for PacNet with SWC.	100
7.3	The event-discriminating update algorithm (EURA)	103
7.4	Performance comparisons against existing link-state update methods. Results are for PacNet with SWC.	106
7.5	The impact of control bandwidth. Results are for PacNet with SWC. .	108
7.6	The impact of burstiness. Results are for PacNet with SWC.	109
7.7	The impact of wavelength conversion. Results are for PacNet.	110
7.8	The impact of network connectivity. Results are for networks with SWC.	111
A.1	PacNet topology. It has 15 nodes and 21 links. Each link in the network is composed of two fibers of opposite directions. The numbers next to the links denote the physical lengths in 10 kilometers, which are used as the metric of calculating the shortest path. In SWC case, nodes 3, 4, 9, and 11 are capable of wavelength conversion.	125
A.2	A 12-node ring network. In SWC case, nodes 0, 4, and 8 are capable of wavelength conversion.	125
A.3	A 13-node mesh network. In SWC case, nodes 5, 6, and 9 are capable of wavelength conversion.	126
A.4	A 16-node mesh torus network.	126

List of Abbreviations

ACTT	Absolute Counter Threshold Trigger
ASP	Adaptive Shortest-Path routing
ATT	Absolute Threshold Trigger
DLE	Dynamic Lightpath Establishment
EURA	Event-discriminating Update Rate Algorithm
FOA	Fuzzy Origination Approach
FSP	Fixed Shortest-Path routing
FWC	Full Wavelength Conversion
GMPLS	Generalized Multiple Protocol Label Switching
GURA	Generic Update Rate Algorithm
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
LSA	Link-State Advertisement
NE	Network Element
NWC	No Wavelength Conversion
OSPF	Open Shortest Path First
RCTT	Relative Counter Threshold Trigger
RTT	Relative Threshold Trigger
RWA	Routing and Wavelength Assignment
SDH	Synchronous Digital Hierarchy
SF-LCP	Shortest-First Least-Congested-Path routing
SLE	Static Lightpath Establishment
SONET	Synchronous Optical NETwork
SWC	Sparse Wavelength Conversion
TE	Traffic Engineering
WDM	Wavelength Division Multiplexing

List of Mathematical Notations

Symbol	Section	Definition
$\alpha_j(m)$	5.1.1	state dependent arrival rate when there is exactly m wavelengths are free on link j
B	3.3	overall network blocking probability
B_R	3.3	routing blocking probability
B_S	3.3	setup blocking probability
B_R^R	5.1.1	routing blocking probability on a given route R
B_R^S	5.1.1	setup blocking probability on a given route R
$B_{T=x}$	4.1	blocking probability when $T = x$
C	4.1	number of wavelengths per link
δ_a	6.1.2	absolute availability change
δ_r	6.1.2	relative availability change
$g_{i,j}$	5.1.2	steady-state probability that a given set of i wavelengths are free on link j
$G_{T=x}$	4.1	blocking performance gain when $T = x$
H	5.1.2	number of links on route R
J	5.1.1	number of links in the network
λ_R	5.1.1	average arrival rate on a given route R
N_a	6.1.2	absolute counter of link-state changes
N_r	6.1.2	relative counter of link-state changes
$N_{s_d^w}$	3.1	number of successfully established lightpaths using wavelength w to destination d
$N_{t_d^w}$	3.1	total number of attempts to establish lightpaths using wavelength w to destination d
$\omega_{j-1,j}$	5.1.3	conditional probability that a path request is not blocked on link j given that it was not blocked on link $j - 1$
P_d^w	3.1	weight assigned to wavelength w for each destination d

LIST OF MATHEMATICAL NOTATIONS

$P_{R,R' X_j=m}^B$	5.1.3	conditional probability that a connection request on route R is blocked at link j by another request on route R' , given that $X_j = m$
$q_j(m)$	5.1.1	probability that exactly m wavelengths are free on link j
ρ	7.1	average rate of packets
σ	7.1	maximum burst size of packets
S_j	6.1.1	link state of link j
$\hat{S}_j$	6.1.1	link-state image of link j
th_a	6.1.2	ATT threshold
th_{ac}	6.1.2	ACTT threshold
th_r	6.1.2	RTT threshold
th_{rc}	6.1.2	RCTT threshold
T	4.1	interval of periodic update
TAT	7.1	Theoretical Arrival Time
(T, τ)	7.1	increment and limit parameters of GURA
$\frac{1}{\mu}$	5.1	duration mean of connections
X_j	5.1.1	random variable representing the number of free wavelengths on link j , $j = 1, 2, \dots, J$
$\hat{X}_j$	6.1.1	number of free wavelengths given $\hat{S}_j$
$Y_{i,j}$	5.1.2	random variable representing the state of wavelength i on link j

CHAPTER 1

Introduction

1.1 Motivation

Users of telecommunication services have been demanding for more bandwidth than ever. Service providers are driven to continuously evaluate and deploy new technologies. Since late 1980s, optical networks have overtaken copper cables and become the mainstream technology for backbone networks, offering obvious performance and reliability advantages. At the early stage, synchronous digital hierarchy (SDH) [1], or synchronous optical network (SONET) [2,3] in North America, was the best and only transport technologies for service providers. Several years later, wavelength-division multiplexing (WDM) emerged as the great technology to unlock the full bandwidth potential of optical fibers. WDM multiplies the capacity of optical fiber strands by as-

CHAPTER 1. INTRODUCTION

signing a separate wavelength for each communication channel. Incoming customer signal is carried over specific wavelengths and multiplexed. WDM enables transparent transportation of a multitude of disparate data protocols such as Ethernet, Fast Ethernet, SONET, and others simultaneously and at native speeds without impact on their performances [4, 5].

As the WDM technology matures, it goes beyond the point-to-point links. Wavelength-routed networks are now becoming commercially viable [5–10]. In a wavelength-routed network, customer traffic is carried by end-to-end all-optical connections called *lightpaths* [9]. A lightpath may span multiple fiber links connected by optical switches. To establish a lightpath, a route has to be selected with a wavelength channel chosen on each span along the route. This is called the *routing and wavelength assignment* (RWA) problem [10]. When there is no wavelength conversion available, RWA is subject to the *wavelength continuity constraint* that requires each lightpath uses the same wavelength on all the links transversed [9–11].

Research on the RWA problem has drawn extensive efforts ever since WDM technology was introduced [9, 12, 13]. The RWA problem can be classified into the *static lightpath establishment* (SLE) problem and the *dynamic lightpath establishment* (DLE) problem. In SLE, connection requests are known in advance and the problem is to setup lightpaths to carry the connections in a global fashion while minimizing the consumptions of network resources. In DLE, a lightpath is established for each connection request as it arrives, and is removed when the connection terminates after some finite amount of time. The typical objective of DLE is to provide RWA solution of each connection request minimizing the probability of getting connection requests blocked.

DLE may be handled in a *centralized* or a *distributed* manner. In centralized control, a single controller is responsible for receiving connection requests, performing

CHAPTER 1. INTRODUCTION

RWA computations, and configuring the network to establish the lightpaths. In distributed control [14, 15], network nodes coordinate with each other to compute the RWA solution and establish the lightpath when a connection request arrives. Distributed control may achieve better scalability and reliability since it eliminates the performance bottleneck of having a single central controller handling everything.

If global information is available, distributed DLE can make the optimal routing and wavelength assignment decisions [12–14]. The link-state based approach to the distributed DLE based on global information is proposed in [14]. In this approach, link update messages are flooded to all nodes in the network whenever the state of the network changes, such as the establishment or removal of a lightpath. Previous research [12] has shown that the link-state based approach has advantages in short stabilizing delays and low blocking probability under light traffic load. More importantly, the link-state based approach has an advantage in traffic engineering, since it supports explicit routing. This attribute can add more fault tolerance to the network. For example, it is simple and fast to compute two link-disjoint routes at the source node, and possible to implement shared protection with the knowledge of full network topology. Therefore, we anticipate the link-state based approach would be widely deployed and hence keep the focus of the thesis on it.

Protocols designed for distributed control have demonstrated their feasibility and efficiency in packet networks. It is natural to extend these protocols to manage wavelength-routed networks, if their efficiency can be retained in the new environment. The Internet Engineering Task Force (IETF) is standardizing a set of distributed protocols within the generalized multiple protocol label switching (GMPLS) framework [16], which is aiming at providing a common control plane for a variety of heterogeneous network architectures. The link-state based routing protocols, such as the open shortest path first (OSPF) routing protocol, are also extended to advertise

CHAPTER 1. INTRODUCTION

link-state information that can be used by RWA computation for lightpaths [17, 18].

Although it is always desirable to maintain up-to-date global information for optimal RWA solutions, trade-off has to be made so that the link-state update messages will not overload the control network and cause protocol failure [19]. Therefore, the link-state update policy, which determines when and what information to be advertised, is a critical component of a link-state based routing protocol [20]. Various link-state update methods can be adopted [20, 21]. *Periodic update* advertises the link state at a constant interval; it has predictable control overhead, while its major drawback is the amount of stale information introduced by the delay between link-state changes and updates. *Triggered update* decides whether the link state shall be advertised according to preset condition(s) of link-state changes, e.g., when the accumulated changes since the last link-state update exceed certain thresholds. Compared with periodic update, triggered update sets up a bound for link-state changes between two consequent updates, which leads to better flexibility in handling dynamic traffic loads under certain circumstances. On the other hand, its blocking performance and control overhead are less predictable and can be significantly affected by the detailed design of the update trigger.

Recent research has shown that the link-state update policy significantly affects the blocking performance and control overhead of QoS routing in packet networks [20]. In wavelength-routed networks, however, it remains an open question about what are the impacts of different link-state update policies. In particular, considering the circuit-switching nature of wavelength-routed networks as well as the wavelength continuity constraint, will the impact be the same as that in packet networks? Will the link-state update policy designed for packet networks remain efficient? Could the link-state based routing ever justify its control overhead in wavelength-routed networks?

CHAPTER 1. INTRODUCTION

To answer these questions, we are motivated to evaluate the feasibility and efficiency of link-state update schemes in the new context of wavelength-routed networks.

1.2 Objectives

Our focus is on evaluating, and improving if possible, the performance of link-state updates in wavelength-routed networks. The objectives of the thesis can be summarized as follows:

1. To examine the feasibility of link-state based routing in wavelength-routed networks. An evaluation of the benefits of link-state update will be conducted to understand how much the performance may be improved, and under what condition(s) the improvement may justify the cost of control overhead. Such evaluations provide a solid basis justifying the follow-up studies.
2. To examine the performance of several most popular link-state update methods. Extensive evaluation helps to understand how link-state update may affect the blocking performance. Meanwhile, a comparative study of different update methods unveils their advantages and disadvantages, and tells whether it is possible to improve these methods in wavelength-routed networks.
3. To design new link-state update methods, or to improve known ones for better performance of link-state based routing protocol in wavelength-routed networks.

1.3 Major contributions of the thesis

Major contributions of the thesis can be summarized as below.

CHAPTER 1. INTRODUCTION

- The first examination of the benefits of link-state update for distributed lightpath establishment in wavelength-routed networks. We show that advertising wavelength availability can only improve network blocking performance under light traffic load. In addition, we demonstrate how the performance gain is affected by routing methods, wavelength conversion, and network connectivity.
- The first examination of the performances of both periodic and triggered link-state updates in wavelength-routed networks on a comprehensive range of configurations with different routing methods, wavelength conversions and network connectivities. For both update methods, we show that exclusive use of wavelength channels and the wavelength continuity constraint cause the impact of stale information highly load-sensitive. Moreover, we found all known triggering schemes not effective in fine-tuning the trade-off between blocking performance and control overhead.
- The first analytical blocking model of distributed lightpath establishment with periodic link-state update. This highly-accurate model shows how different components of network blocking probability are affected by inaccurate information under different traffic loads, and explains the high sensitivity of blocking performance to link-state update interval under light traffic loads.
- A novel link-state update method that performs better than all known link-state update methods. It adapts to available control bandwidth by actively regulating both the average rate and the burstiness of the update rate at the same time. More significantly, under limited control bandwidth, it puts the stale link-state information of more negative impact at a higher priority to be removed, and therefore further improves the blocking performance.

1.4 Organization of the thesis

The rest part of this thesis is organized as follows.

Chapter 2 presents a review of lightpath establishment in wavelength-routed networks. Basic theory and important results in this field are included in a comprehensive literature review.

Chapter 3 describes the model of link-state based lightpath establishment. We discuss in detail the routing and signaling protocols, as well as the performance evaluation methodology used in our research.

Chapter 4 examines the benefits of link-state update under different traffic loads. We also study the impact of different routing methods, wavelength conversions, and network connectivities.

Chapter 5 examines periodic link-state update. We present a comprehensive performance analysis and show how the blocking performance may be significantly impacted by stale link-state information.

Chapter 6 examines triggered link-state update. We study four different triggering schemes. Two of them are widely adopted and the others are first proposed. The study shows that the high sensitivity of the blocking performance to stale link-state information invalidates adjusting the trigger parameter to fine-tune the trade-off between blocking performance and control overhead.

Chapter 7 introduces a novel link-state update method that improves network performance significantly. This method regulates both the average rate and the burstiness of the update rate at the same time. More significantly, under limited control bandwidth, it puts the stale link-state information of more negative impact at a higher priority to be removed, and therefore further improves the blocking performance as demonstrated in a variety of configurations.

In Chapter 8 conclusions and future directions of the research are presented.

CHAPTER 2

Lightpath Establishment in All-Optical Networks

Wavelength Division Multiplexing (WDM) has been the core backbone technology of modern telecommunication networks. It is also expanding to the edge of telecommunication networks, i.e., the metropolitan and access networks. The next generation all-optical WDM networks, namely the wavelength-routed networks, can eliminate the high cost of optical-to-electronic-to-optical (O/E/O) conversion on intermediate network nodes [4–7]. The all-optical connection between a pair of source and destination nodes is called a lightpath, which concept was first introduced more than ten years ago [9]. Since then the establishment of lightpath has been one of the most challenging problems in the design of wavelength-routed networks. Various kinds of routing and wavelength assignment (RWA) algorithms and protocols have

 CHAPTER 2. LIGHTPATH ESTABLISHMENT IN ALL-OPTICAL NETWORKS

been developed [5, 13]. The algorithms select routes and wavelengths for connection requests either to minimize the blocking probability of connection requests or to achieve the best utilization of network resources. When there is no wavelength conversion available, RWA is subject to the *wavelength continuity constraint* that requires each lightpath uses the same wavelength on all the links transversed [9–11]. In addition to RWA algorithms, the control plane plays a critical part in the overall network performance [12, 19]: a routing protocol is used to provide information exchange between network nodes, and a signaling protocol to configure network nodes to establish lightpath physically.

In this chapter, first we review the full scope of lightpath establishment to provide the context of our research work. Then a close examination of distributed dynamic lightpath establishment is conducted, where we justify our focus on the link-state based lightpath establishment. Finally, we discuss the issues and challenges in the design of control plane.

2.1 Overview of lightpath establishment

When the traffic pattern is known in advance, or it will not change significantly over time, we have the so-called static lightpath establishment (SLE) problem [9]. The SLE problem can generally be formulated as an integer linear programming (ILP) problem [9, 10], which under most cases is an NP-complete [9] problem and therefore may only be solved for very small networks. For larger networks, heuristic methods have to be used, e.g. [10, 13]. Generally the solution of SLE problem is computed in an off-line manner during network planning or in an on-line manner by a central controller. The SLE problem has been well studied and many solutions have been proposed [13].

CHAPTER 2. LIGHTPATH ESTABLISHMENT IN ALL-OPTICAL NETWORKS

When the traffic pattern is constantly changing, lightpaths have to be established and removed dynamically in response. This imposes the main challenge of the dynamic lightpath establishment (DLE) problem [9, 13]. The DLE is an NP-complete problem that is usually divided into the more manageable routing and wavelength assignment sub-problems.

Various methods for solving the routing sub-problem have been proposed. They mainly include [13]:

- fixed routing, which always uses the same predetermined route for a given pair of source-destination nodes. An example of fixed routing is the Fixed Shortest-Path (FSP) routing, which uses the shortest-path route and leaves only wavelength assignment to be done upon connection requests. Fixed routing is very simple; however, it may lead to high blocking probabilities when resources along the path are scarce.
- fixed-alternate routing, which selects the route from an ordered list of several predetermined routes for a given pair of source-destination nodes. For example, these routes may include the shortest-path route, the second-shortest-path route, and the third-shortest-path route, etc. Fixed-alternate routing can significantly reduce the connection blocking probability compared with fixed routing.
- adaptive routing, which selects the route by dynamic searching all feasible routes between a given pair of source-destination nodes based on current network state. The selection from feasible routes may have preference, such as giving short paths priority. Adaptive routing has the advantage of lower blocking probability than fixed and fixed-alternate routing, but it requires more complicated online computations and controls.

The main concern of the wavelength assignment sub-problem is how to select a

 CHAPTER 2. LIGHTPATH ESTABLISHMENT IN ALL-OPTICAL NETWORKS

wavelength for a given path when there are multiple wavelengths satisfying wavelength continuity constraint. The most popular wavelength assignment methods include first-fit selection and random selection. In first-fit selection, the wavelengths are indexed and the lowest-indexed feasible wavelength will be selected. In random selection, one wavelength is randomly selected from the set of feasible wavelengths, usually with uniform probability. Both of the two wavelength assignment methods can be combined with aforementioned routing methods into the RWA algorithms for the dynamic lightpath establishment problem. The implementation of the RWA algorithms can be either in a *centralized* manner by a single central controller, or in a *distributed* manner by individual network nodes [5, 13].

In centralized DLE, the central controller maintains the state of network resources, performs RWA computations, and provisions network nodes to establish the lightpaths. Global network state information is available to the RWA algorithm running at the control controller, which may be able to compute the solutions in favor of global optimization. When the network grows in size, however, the central controller has to increase significantly its processing power and communication bandwidth. Therefore, the limited resources of the central controller are the bottleneck of network scalability, which may lead to serious reliability problems. In practice, centralized DLE may not suit large networks [14, 22].

In distributed DLE, the bottleneck of central controller is eliminated. The computation burden to run RWA algorithm is distributed to individual nodes in the network. The network nodes learn current network state by exchanging information with each other. Well-designed distributed DLE scales with large networks, and is adaptive to topological changes such as adding or removing network nodes. In the emerging GMPLS framework [16], distributed DLE may be implemented by coordinating a set of distributed protocols running on the common control plane [18, 23]. We will examine

distributed DLE in more detail in the next section.

2.2 Distributed dynamic lightpath establishment

The performance of distributed DLE can be significantly affected by the amount of state information available to each node. If no global information is exchanged, i.e., only local information is available, the network incurs less control overhead. However, distributed DLE will not have necessary information to make optimal routing and wavelength decisions [24–26]. In this case, wavelength assignment must be done in a hop-by-hop manner, which increases the complexity of the signaling protocol. Moreover, if a routing method other than fixed routing is used, additional problems need to be addressed. For example, for adaptive routing, such as the deflection routing [25], using only local information may lead to long paths that consume a lot of network resources, unless some additional strategies are used to keep the route length within a reasonable range.

If global information is available, distributed DLE often makes more efficient routing and wavelength assignment decisions [12–14]. However, it has to be carefully designed to deal with the task of maintaining a potentially large amount of state information that changes constantly. There are two popular approaches to implement a distributed DLE algorithm based on global information. In a link-state based approach [14], each node maintains its own global link-state database. Whenever the state of the network changes, such as the establishment or removal of a lightpath, link update messages are flooded to all nodes in the network. In a distance-vector based approach [22], complete link-state information is not maintained at each node. Instead, only a routing table is maintained by running distributed routing algorithm such as the distributed Bellman-Ford algorithm [27]. The routing table indicates the next

CHAPTER 2. LIGHTPATH ESTABLISHMENT IN ALL-OPTICAL NETWORKS

hop and distance to each destination node on each wavelength. As in the link-state based approach, it is required that the nodes must update their routing table whenever a lightpath causes a change in the routing table. The link-state based and distance-vector based approaches are easy to implement by extending existing protocols, such as OSPF [17] and RIP [28], respectively.

Previous research [12] has shown that the distance-vector based approach has advantages in shorter connection setup delays and lower blocking probability under heavy traffic load, while the link-state based approach has advantages in shorter stabilizing delays and lower blocking probability under light traffic load. More importantly, the link-state based approach has an advantage in traffic engineering, since it supports explicit routing. This attribute can add more fault tolerance to the network. For example, with the knowledge of global link-state information, it is simple and fast to compute two link-disjoint routes at the source node and possible to implement shared protection. Therefore, we anticipate the link-state based approach would be widely deployed and hence keep the focus of the thesis on it.

If the frequency of lightpath establishment and removal is high, the network may suffer intensive control overhead. Although it is always desirable to maintain up-to-date global information for optimal RWA solutions, trade-off has to be made so that the link-state update messages will not overload the control channels of the network and cause control plane failure [19]. In the following section, we will examine the design of control plane for wavelength-routed networks, which is critical to overall performance.

2.3 Control plane and its reliability

The control plane is responsible for communicating the network nodes to exchange information of network topology and the state of network resources, as well as to establish or remove lightpaths [19, 29]. These functions are primarily achieved through two control protocols: a routing protocol for topology and resource discovery, and a signaling protocol for lightpath provisioning, maintenance, and removal. The Internet Engineering Task Force (IETF) is standardizing a set of distributed protocols within the generalized multiple protocol label switching (GMPLS) framework [16], which is aiming at providing a common control plane for a variety of heterogeneous network architectures. Routing and signaling protocols designed for packet networks have demonstrated their feasibility and efficiency. Knowledge about these protocols and well-trained professionals are widely available. Therefore, extending these protocols to manage wavelength-routed networks are naturally preferable than designing new ones from scratch. For example, the open shortest path first (OSPF) routing protocol is extended to implement distributed lightpath establishment [17, 18, 23].

2.3.1 Routing protocol

The routing protocol is responsible for reliably advertising the optical network topology and resource information within and between network routing domains. The information exchanged in wavelength-routed networks may be either *quasi-static* or *dynamic* [19]. Quasi-static information usually remains constant over a long duration; it includes optical impairment parameters and shared risk link groups (SRLGs) [30]. Dynamic information includes wavelength availability and changes frequently upon lightpath operations (e.g., lightpath establishment and deletion). Quasi-static information may be provided by manual configuration or infrequent advertisements, while

CHAPTER 2. LIGHTPATH ESTABLISHMENT IN ALL-OPTICAL NETWORKS

dynamic information needs to be updated upon lightpath operations to allow RWA computation to use the current state information. Dynamic information is the major factor concerning the scalability and reliability of the routing protocol, especially when global information is required. For example, in the link-state based approach, the routing protocol has to decide carefully when to advertise wavelength availability so that the control network will not be overloaded by excessive link-state update messages, even if the frequency of lightpath establishment and removal is high. In the next chapter, we will examine in detail the methods used to control updating dynamic information in link-state based lightpath establishment.

2.3.2 Signaling protocol

The signaling protocol is responsible for network resource reservation and release. Resource reservation has two primary methods, namely the parallel method [14, 31] and the sequential method [26, 31, 32]. In the parallel method, the reservation request is sent out to all the nodes along the route at the same time by the source node. Such method requires the presence of knowledge about to which nodes the request should be sent. In most cases, global information is required. In sequential method, the reservation request passes through each node along the path in a hop-by-hop manner.

Generally, the sequential method can be further divided into two classes, namely forward reservation and backward reservation [26, 31, 32]. In forward reservation, network resources are directly reserved by a reservation request sent by the source node from source to destination on a hop-by-hop basis. Further details of the forward reservation method depend on how much information is available to the source node and the strategies of reservation (e.g., how many wavelengths are reserved on each hop when there are multiple available channels). For example, if global information is available to the source node, explicit routing may be used, and the source node may

CHAPTER 2. LIGHTPATH ESTABLISHMENT IN ALL-OPTICAL NETWORKS

only try to reserve the minimum resource required for setting up a lightpath. If the source node has only local information about its neighboring links, it generally would try to reserve more than one wavelength along the path and then a single wavelength will be confirmed by the destination (if applicable).

In backward reservation, network resources are not reserved when the path request is traveling from source to destination. Instead, network resources are reserved by the reservation request initiated by the destination node after the path request is received by the destination node. If global information is not available, the path request may collect and carry resource availability information along forward direction to help the destination node to make better decisions. Backward reservation may reduce the idle duration between network resources are reserved and data transmission actually begins [26, 31, 32]. In addition, it avoids reserving multiple wavelength channels along the route, and hence does not have the over-reservation problem that may exist in forward reservation.

Although both forward reservation and backward reservation can be used to support all the three routing methods aforementioned, forward reservation is more favorable when global information is available, while backward reservation is usually a better choice when we have only local information [26, 32].

CHAPTER 3

Modeling Link-State Based Lightpath Establishment

In link-state based lightpath establishment, each node in the wavelength-routed network maintains its own link-state database. When a connection request arrives, the source node performs the RWA computation based on its link-state database. If a feasible lightpath is found by the RWA computation, the source node would then use the signaling protocol to initialize the establishment of the lightpath. Notified by the signaling protocol, the intermediate nodes and the destination node may either accept or reject the connection request, generally depending on whether the wavelength channel requested by the source node is still available or not.

Link-state based lightpath establishment has the advantages of faster stabilizing delay and lower blocking probability under light traffic load. Besides, the availability

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

of global link-state information supports important functions in traffic engineering, such as explicit routing and shared protection. However, if the frequency of lightpath establishment and removal is high, the network may suffer intensive control overhead, since each node must maintain its own view of wavelength availability on the links, distribute link-state information to other nodes, and compute and establish lightpaths for new connection requests.

The routing protocol offers considerable flexibility in limiting control overheads. In particular, various link-state update methods can be used to control the frequency of link-state updates. Link-state information can be advertised in a periodic fashion or in response to a significant change in the wavelength availability. However, large intervals and coarse triggers result in stale link-state information, which can cause the RWA computation to select a suboptimal route or a route that uses a wavelength no longer available. Hence, tuning the frequency of link-state update messages requires a good understanding of the trade-off between control overheads and the accuracy of the RWA computation.

In this chapter, we develop a model of link-state based lightpath establishment that enables investigation of the fundamental trade-off between link-state update overhead and the network blocking performance. We describe the primary components of the routing and signaling model, namely RWA computation, link-state updates, and signaling. These components, along with the network and traffic models, are necessarily complex to allow us to study a wide variety of policies and configurations. Therefore, we have built for wavelength-routed networks a simulator that can efficiently handle a variety of network and routing configurations. Section 3.4 of this chapter describes the features and the high-level design of the simulator.

3.1 Routing and signaling model

3.1.1 RWA computation

In the thesis, three routing methods are evaluated to study their capabilities in exploiting dynamic link-state information, wavelength availability information in particular:

- fixed shortest-path (FSP) routing always uses the shortest-path between each source-destination pair, and wavelength availability information is only used for random selection in wavelength assignment.
- adaptive shortest-path (ASP) routing performs dynamic calculations based on the link-state database and looks for the shortest path with at least one free wavelength; when there are multiple paths with the same distance, ASP randomly chooses one of them.
- shortest-first least-congested-path (SF-LCP) routing [33] searches for the shortest in a pre-determined path set for a source-destination pair. If there is a tie, the one with the largest number of available wavelengths is selected. In our implementation, the pre-determined set between a node pair includes all paths with a hop length not more than the diameter of the network. After the route is determined, random selection is applied for wavelength assignment.

The evaluation of the above routing methods assumes that the same metric is used in determining the shortest path. This ensures the comparisons among the methods measure only their different capabilities in exploiting dynamic link-state information. In particular, a comparison between ASP and FSP will show how much the adaptive routing can better utilize wavelength availability information than the fixed routing; and SF-LCP is studied to show whether load-balanced routing using the wavelength availability information can further improve network performance over ASP. Unless

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

otherwise stated, the hop length is always used in this thesis as the metric of the shortest path.

In the wavelength assignment sub-problem, we will consider the most popular *first-fit* selection and *random* selection, as well as the recently-proposed *weighted* selection [34, 35]. The first-fit selection chooses the smallest-indexed available wavelength on the route so that the higher-indexed wavelengths are more likely to remain available for future connection requests. In a distributed environment, however, it may cause many lightpaths to compete for the small-indexed wavelengths, and hence be outperformed by the random selection [36]. Specially designed for wavelength assignment in distributed environment, the weighted selection reduces the probability of wavelength conflict by giving the lightpath between each node-pair a preference to certain wavelengths based on historical record. In detail, each source node maintains a weight assigned to wavelength w for each destination d as defined below [34],

$$P_d^w = \frac{N_{s_d^w}}{N_{t_d^w}} \quad (3.1)$$

where $N_{s_d^w}$ is the number of successfully established lightpaths using wavelength w to destination d , and $N_{t_d^w}$ is the total number of (successful or unsuccessful) attempts to establish lightpaths using wavelength w to destination d . The weight is used rank each available wavelengths and the one with the highest weight is chosen to carry the lightpath. This means the available wavelength with the highest successful rate in history record will be given a priority. In the following chapter, we will compare the three wavelength assignment methods in the context of link-state based lightpath establishment.

3.1.2 Link-state update

The routing protocol is required to disseminate dynamic link-state information, such as wavelength availability, within the optical network domain. Kompella and Rekhter presented an Internet draft in which they discussed the information that needs to be flooded by any routing protocol in support of GMPLS [18]. In that draft, a generic approach to handle networks comprised of packet switch capable (PSC), time-division-multiplexing capable (TDMC), lambda switch capable (LSC), and fiber switch capable (FSC) equipments was presented. This generic approach is later extended to handle the specific information in wavelength-routed networks [23], so that more efficient routing and better provisioning of lightpaths can be realized. The extension introduces a new link-state advertisement (LSA) to advertise wavelengths availability. The LSA uses the OSPF opaque LSA option and the OSPF traffic engineering (TE) extension as a vehicle for advertising the link state. The OSPF opaque LSA option defined in [37] provides a generalized mechanism for the OSPF protocol [17] to carry additional information. The TE extension of OSPF defined in [38] provides a way of describing the traffic engineering topology (including bandwidth and administrative constraints) and distributing this information within a given OSPF area. A typical IP packet format to carry the wavelength availability opaque LSA is illustrated in Fig. 3.1.

Advertisement of link-state information is under the control of the link-state update methods. Once the link-state update method decides to advertise the link state, the link-state information is flooded to the network by the same mechanism used in OSPF [17]. Every network node has accurate information about the wavelength availability of its own neighboring links and potentially stale information about the other links in the network. Stale link-state information causes RWA computation to make invalid decisions and leads to higher blocking probability. As illustrated in Fig. 3.2, if

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

(a) Periodic update.

(b) Triggered update.

Figure 3.2: Illustration of the link-state update methods and the concepts of pseudo-surplus and pseudo-deficient information.

a wavelength channel is advertised as free but actually has been reserved, it is said to be *pseudo-surplus*; if a wavelength channel is advertised as occupied but actually has been released, it is said to be *pseudo-deficient*. Note that Fig. 3.2 is solely for illustrating the concepts of pseudo-surplus and pseudo-deficient link-states. The details of different link-state update methods will be discussed in later chapters.

The link-state update methods have to be carefully designed to deal with the task of maintaining a potentially large amount of state information that changes constantly.

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

If the frequency of lightpath establishment and removal is high, the network may suffer intensive control overhead. Therefore, although it is always desirable to maintain up-to-date global information for optimal RWA solutions, trade-off has to be made so that the link-state update messages will not overload the control channels of the network and cause control plane failure [19]. Various link-state update methods have been proposed to handle such trade-off in wavelength-routed networks [21, 23]. They generally fall into two classes:

1. timer-based methods, which regularly advertise link state at preset intervals regardless of when the link state may change. For example, periodic update [21] advertises the link state at a constant interval, as illustrated in Fig. 3.2(a); and the so-called *fuzzy origination approach* (FOA) proposed in [23] uses fuzzy inference system to dynamically set smaller intervals when network resources are more utilized. The major advantage of timer-based methods is their predictable control overhead; while their major drawback is the amount of stale information introduced by the delay between link-state changes and updates. Since link-state updates are not aligned in time scale with changes, such delay is inevitable even if the update interval is dynamically adjusted as that in the FOA.
2. change-based methods, which decide whether the link state shall be advertised or not when a change happens according to preset condition(s). For example, several triggered update schemes were proposed in [21] that advertise the link state when the accumulated changes since the last link-state update exceed certain thresholds. In Fig. 3.2(b), we illustrate the immediate trigger that advertise the link-state upon every link-state change, i.e., threshold equals to 1 change. Compared with periodic update, triggered update sets up a bound for link-state changes between two consequent updates, which leads to better flexibility in

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

handling dynamic traffic loads under certain circumstances. On the other hand, its blocking performance and control overhead are less predictable and can be significantly affected by the detailed design of the update trigger. Therefore, in practice, a hold-down timer is usually adopted as a complement to enforce a minimal interval between link-state updates, so that the control overhead will not exceed a maximum limit [17].

In the following chapters, we will examine periodic update and triggered update by comprehensive performance analyses and evaluations. In addition, we will propose a novel link-state update method to improve the network performance.

3.1.3 Signaling

A typical signaling protocol using forward reservation is proposed in [12]. The signaling procedures are depicted in Fig. 3.3 by a UML sequence diagram [39], where a dotted vertical line represents time and the vertical box represents the lifetime of a process. Once a feasible lightpath is found by RWA computation, the source node would reserve local resources and then send the next hop on the lightpath a `PATH_REQUEST` message that contains the information of the selected route and wavelength. Upon receiving a `PATH_REQUEST` message, each intermediate node would check on the next hop the availability of the wavelength requested by the lightpath. If the wavelength is free, the node reserves the wavelength and forwards the `PATH_REQUEST` to the next node; otherwise, the node rejects the lightpath and sends a `PATH_NACK` message back to the source node. A node will release the reserved wavelength upon receiving a `PATH_NACK` message. If the `PATH_REQUEST` message successfully reaches the destination node, a `PATH_ACK` message would be returned to the source node and data transmission would then begin. Once the transmission is done, a `PATH_RELEASE` message would be sent out by the source node to

let all the other nodes along the lightpath release the reserved wavelength and updates their local link-state databases accordingly. If a change-based update method is used, every time when a node receives a PATH_ACK or a PATH_RELEASE message, the node calls the update method to decide if an update should be advertised.

3.2 Network and traffic model

Studying protocol behavior faces a key challenge of how to represent the underlying network topology and traffic patterns. The constantly changing and distributed nature of current carrier's networks makes it difficult to define a *typical* configuration. Adding to the challenge are observations that conclusions about algorithm or protocol performance may in fact vary dramatically with the underlying network models. We consider a range of network topologies with differences in important parameters such as the wavelength conversion configuration and the degree of connectivity.

Wavelength conversion relaxes wavelength continuity constraint and reduces blocking probability [11]. However, wavelength converters are still expensive at present. It is costly to equip all network nodes with wavelength converters, in which case the network is said to be with full wavelength conversion (FWC). By carefully choosing a subset of network nodes to equip with wavelength converters, sparse wavelength conversion (SWC) may offer slightly worse blocking performance than FWC but at a much lower cost [40–42]. In later chapters, we will evaluate the impact of wavelength conversion availability by comparing the performance of SWC against both FWC and no wavelength conversion (NWC). In SWC, the network nodes of largest degrees are chosen to be equipped with wavelength conversion [41]. We assume each node capable of wavelength conversion has enough converters to convert all incoming wavelengths to outgoing wavelengths without blocking.

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

Figure 3.3: Sequence diagram of lightpath establishment signaling procedures.

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

Network connectivity is also an important parameter affecting the network blocking performance. Higher degree of connectivity allows more lightpaths with short hop lengths to be established, which generally helps to lower the impacts of outdated information. Besides, a network with a higher degree of connectivity may have more balanced load on its links, which also helps avoid connection blockings. Therefore, we are motivated to carry out our studies in networks with different degrees of connectivity. We will consider a 12-node ring, a 13-node mesh, a 16-node mesh torus, and an inter-connected ring network (PacNet). The topologies of these networks are shown in Appendix A. We assume that the topology remains fixed throughout each simulation experiment; that is, we do not model the effects of link failures.

In our study, the traffic load is modeled as follows. Connection requests are generated between each pair of source-destination nodes according to a Poisson process with rate λ . We assume the duration of each connection is exponentially distributed with a mean value of 1 unit time. Therefore, the traffic load has the same value as λ when denoted in units of Erlang.

3.3 Evaluation metrics

In order to evaluate the performance and costs of link-state based lightpath establishment, we identify a set of metrics that unveil a variety of effects. Many of the results in later chapters focus on the two important metrics for performance and overhead, namely the connection request blocking probability and the link-state update rate, respectively. The blocking probability demonstrates the ability of the RWA algorithm and update policies to find feasible paths for connection requests. Meanwhile, the link-state update rate gives an indication of the required control bandwidth and processing overhead necessary to achieve the corresponding blocking probability. Below

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

we describe these metrics and identify some of their important characteristics in the context of wavelength-routed networks.

In link-state based lightpath establishment, the network blocking probability is composed of two parts: (1) *routing blocking probability*, which is the probability that the source node finds no feasible lightpath for a connection request according to the link-state information in its database; we denote routing blocking probability as B_R ; (2) *setup blocking probability*, which is the conditional probability that the lightpath cannot be set up at an intermediate node, given that an RWA solution is found by the source node; we denote setup blocking probability as B_S . The overall network blocking probability B then equals to

$$B = 1 - (1 - B_R) \times (1 - B_S). \quad (3.2)$$

With pseudo-deficient wavelengths, RWA computation may block a request that actually could be accommodated by the network, thus increasing B_R . With pseudo-surplus wavelengths, RWA computation may select a wavelength that actually has been occupied and have the request for that wavelength blocked during lightpath establishment procedure, thus introducing B_S . Closer observation reveals that these two components of blocking probability are related with each other. For example, pseudo-surplus link-state information leads to higher setup blocking probability and hence reduces network utilization; while at the same time the routing blocking probability may be reduced by pseudo-surplus link-state information. As we will see later, such relationship between stale link-state information and the two components of blocking probability makes the overall blocking performance highly complicated. We will examine how B , B_R and B_S would be impacted differently by pseudo-surplus information and pseudo-deficient information under a variety of configurations in later

chapters.

The link-state update rate measures the average rate of link-state update per link. It gives an indication of the required control bandwidth and processing overhead to achieve the corresponding blocking probability. Since the packet size of a wavelength availability opaque LSA is known, the link-state update rate can be easily converted into the consumed bandwidth in byte per second. For example, if the update rate is 10 messages per second, then the link-state update messages consume an average bandwidth of 1640 bytes per second with the typical 164-byte LSA packet length.

A key objective of our study is to build the relationship between the link-state update policy and the blocking performance. The blocking probability and link-state update rate provide important measurements to achieve this objective.

3.4 Simulation environment

To evaluate the link-state based lightpath establishment in wavelength-routed networks, we developed an extensible discrete event-driven simulator written in C++ language. The simulator is object-oriented and fully extensible by making use of polymorphism support of C++. Therefore, it is able to provide complete support within a unified architecture for the model parameters and evaluation metrics described in Section 3.1-3.3. Unless stated otherwise, for each simulation configuration in this thesis, five simulations runs were performed, each with a different seed for the random-number generator, resulting in a different connection arrival sequence for each run. Each simulation run consisted of 500,000 connection requests. The reported simulation data are within the 95% confidence interval. Following we describe the features and the high-level design of the simulator.

3.4.1 Simulator design

One of our primary goals in designing the simulator was to build a robust and extensible one while keeping its memory footprint small. Although other general-purpose simulation packages were available [43], we found that they would require substantial customization or addition of some key missing features. Moreover, unnecessary general-purpose components in these packages significantly complicate bug location and increase memory requirements. Therefore, we design our simulator to be a specialized one for wavelength-routed networks, which allows us to apply optimization tools to maximize speed and minimize memory footprint. In addition, by constructing the simulator with object-oriented C++ language and polymorphism, we are able to extend the simulator easily to support different parameters and evaluation metrics in our current and future models.

Fig. 3.4 illustrates the architecture of the simulator, which has five major components. The key specialized component is the *wavelength-routed network element* (NE), which implements all routing and signaling protocols as in an actual network. The *application layer emulator* and the *physical layer emulator* simulates the application layer and physical layer, respectively. The application layer emulator is responsible for generating traffic and monitoring the network performance from customers' perspective. The physical layer emulator simulates physical devices and channels, modeling processing delays, propagation delays, and timers. The layer emulators interact indirectly with multiple instances of network elements through the *discrete event-driven simulation engine*. The engine creates and inserts events into queues when invoked by the emulators or the network element. A system clock is maintained by the simulation engine and automatically advanced after each event. The simulation engine also invokes the *data collection, analysis and report* component upon interesting events.

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

Figure 3.4: Architecture of the simulator

Each component may contain functional modules that perform specific tasks as described below.

- **Lightpath manager:** this module serves as the controller for all lightpath establishment functions. It implements the signaling protocol, accepts connection requests from application layer, and coordinates other modules to carry out the procedures of lightpath establishment.
- **Lightpath router:** This module maintains the routing table based on the link-state database. It is invoked by the lightpath manager to perform RWA computation upon connection requests from the application layer. This module may be extended to implement different RWA algorithms.
- **Local resource manager:** This module is responsible for reservation of local resources, including switching fabric and neighboring links. It also accepts

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

query of the states of local resources.

- **Link-state update manager:** This module maintains the link-state database and implements the link-state update policy. It processes link-state update messages received from the peering module of other network elements and may forward the messages if required by the flooding procedure. When notified by the local resource manager about reservations, it updates link-state database and decides whether an update messages shall be sent out or not according to the policy. It is also notified by the lightpath manager about the states of remote links of passing-through lightpaths. This module may be extended to implement different link-state update policies.
- **NE performance monitor:** This module collects NE-specific performance statistics such as the number of blocked connection requests, the number of link-state update messages, and the number of link-state changes that is not updated.
- **Traffic generator:** This module is responsible for determining characteristics of each connection request according to the specified traffic parameters. It supplies each new connection request with a source node, a destination node, an inter-arrival time relative to the last request between a given node-pair, and a connection duration. During initialization of the simulation, this module creates an initial connection request from each source-destination pair. Then, for each connection request that arrives and is handled, two new events may be generated: a lightpath release event (in the case of successful lightpath establishment), and the next new connection request for the same source-destination pair.
- **Application layer performance monitor:** This module collects specific per-

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

formance statistics of the application layer. Usually these statistics are network-level aggregations of NE-specific statistics, such as the overall number of blocked connection requests in the network.

- **Control channel:** This module emulates a physical control channel; it accepts control messages from one network element and transfer to another network element after the link delay. This module may be extended for more complex channel modeling if necessary.
- **Data channel:** This module emulates a physical data channel. In our current study, it is a dummy module that does nothing. However, future study may extend this module to consider data channel characteristics such as chromatic dispersion.
- **Timer:** This module emulates a physical timer. A network element may invoke the timer module to set a timer that expires after a specific duration. When the timer expires, the timer handler of the network element will be invoked.

3.4.2 Features and usage

The modules of the simulator have been extended as we evaluate different RWA algorithms and link-state update policies. Since we have kept extensibility in mind when designing the architecture of the simulator, new features can be added to the simulator with small effort in configuration management, testing, and debugging. Below we discuss some important features other than the support for the routing and signaling model and performance metrics described in Section 3.1-3.3.

- **Flexible configuration:** The simulator is configurable at run-time by specifying command-line switches and the data file that contains network topology.

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

For example, the user may turn on a switch to use a specific RWA algorithm or a specific link-state update policy. Moreover, running simulation over different networks can be done by simply passing a different data file to the simulator, since the simulator is independent of the specific network configuration.

- **Tracing and debugging assistance:** The simulator has two levels of tracing that can be turned on independently by a command-line switch to assist debugging. The first is event tracing offered by the discrete event-driven simulation engine. Event tracing supports precise examination of all discrete events processed by the engine. The second is function tracing offered by all components to report function calls. Function tracing is especially useful when inspecting signaling and link-state update procedures. For example, in the link-state update manager, each operation on the link-state database may be traced with the full list of parameters. The link-state database may also be dumped when necessary.
- **Simulation validation:** Ideally, simulation results should be validated by comparing with an actual implementation of the NEs running in a real network. In the absence of such a test-bed, we rely on analytical techniques to validate that various portions of the simulation give expected results. We developed an analytical blocking model for specific routing cases, which validates our simulation results, as will be presented in Chapter 5. In addition, analytic expressions for simpler quantities other than the blocking probability may be used to validate the simulation results. For example, the offered traffic load may be compared with simulation statistics to check that the traffic generator is working correctly. Similarly, by setting the link-state update trigger to 0, it is possible to develop an expression for the expected update rate and compare it to the collected statis-

CHAPTER 3. MODELING LINK-STATE BASED LIGHTPATH ESTABLISHMENT

tics. Although these quantities do not validate the overall operation of the simulator, they offer considerable evidence of correct results where the analytical blocking model cannot be applied directly.

CHAPTER 4

Benefits of Link-State Update

The link-state based lightpath establishment has the advantages of having shorter stabilizing delays and lower blocking probability under light traffic load. More importantly, it has distinct advantage in traffic engineering since it supports explicit routing. This attribute can add more fault tolerance to the network. For example, with the knowledge of global link-state information, it is simple and fast to compute two link-disjoint routes at the source node, and make possible the implementation of shared protection. Nevertheless, these benefits do not come without drawbacks. The link-state based routing protocol has to make trade-off so that the link-state update messages will not overload the control channels and cause control plane failure. This motivates us to study the conditions under which the blocking performance improvement offered by advertising wavelength availability justifies the control overhead.

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Previous studies have evaluated the performance of different link-state update schemes in wavelength-routed networks. However, researchers have some conflicting views on the benefits of link-state update, as well as under what conditions such benefits justify the cost of control overhead. While some focus their discussions on the network performance under light traffic load and others study the performance under heavy traffic load [23, 36], the missing piece of previous research works is a measurement of how much the link-state update may improve the blocking performance. In this chapter, we will quantify the benefit of link-state update and develop a solid understanding on how much link-state update can help improve blocking performance with various network configurations. We will conduct the study of distributed lightpath establishment with different values for various important parameters, which include different routing methods, densities of wavelength conversions, and network topologies. We will show that advertising wavelength availability can only improve network blocking performance under light traffic load, and demonstrate how the performance gain is affected by these important parameters in a distributed environment.

4.1 Performance study

By extensive simulations, we study three typical link-state update schemes, which are either periodic update or triggered update:

- Immediate triggered update, where wavelength availability is immediately known to all network nodes whenever a lightpath is established or released over a link. This scheme is equivalent to the case of centralized control. Lightpaths are blocked only when there is no sufficient resource. Although this scheme is usually impractical due to propagation delay and control bandwidth limitation, its performance is the best and can be used for benchmarking. We denoted this

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

scheme as $T = 0$.

- No update, where no update is advertised at all. In this scheme, RWA computation assumes all wavelengths are available and tries lightpaths blindly, which causes lightpath to be blocked by wavelength conflicts during the establishment process. The performance is expected to be the worst. If in any other case the performance is nearly the same, then advertising wavelength availability does not help regardless of how frequent it is. We denoted this scheme as $T = \infty$.
- Periodic update with intervals of $T = 1$ and $T = 0.1$, assuming that the average duration of connections is unit time. Periodic update advertises link state for every fixed interval of T . It has predictable control overhead and is normally practical in real-world networks. How much the performance of periodic update can approach the performance of $T = 0$ will show the headroom that more sophisticated schemes, e.g., triggered updates that will be discussed in Chapter 6 and Chapter 7, may be able to achieve.

When wavelength conversion is available, a route from the source to the destination can be composed of one or more *segments*. The two end nodes of each segment can only be the source node or the destination node, or an intermediate node with wavelength conversion [42]. A segment may consist of one or more links in SWC, while each segment is a single link in FWC. Because the end nodes of each segment are more likely to have accurate information about the wavelength availability within the segment, other than assigning wavelengths in all segments solely by the source node as aforementioned, better performance may be achieved by assigning the wavelength in a segment-by-segment manner. Specifically, we let the wavelength assignment on each segment to be decided by the end node closer to the source node, called the *upstream* end node hereafter. In this case, although the source node no

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.1: Illustration of source wavelength assignment and segment-by-segment wavelength assignment.

longer decides wavelength assignment for the whole lightpath, link-state information could still be useful in helping the source node to find the best route. Fig. 4.1 illustrates the signaling of setting up a lightpath with segment-by-segment assignment. Later in Section 4.1, we will investigate the performance of the segment-by-segment wavelength assignment method.

Network connectivity is an important parameter of network topology. Higher degree of connectivity allows more lightpaths with short hop lengths to be established, which generally helps to lower the impacts of outdated information. Besides, a network with a higher degree of connectivity may have more balanced load over its links, avoiding link congestion and blocking. We will consider a 12-node ring, a 13-node mesh, and an inter-connected ring network (PacNet). These networks are shown in Appendix A. Another important parameter of network topology is the number of wavelengths per link, which we denote as C . The blocking performance may be significantly different when C differs. We will investigate three cases of $C = 8$, $C = 32$, and $C = 64$ in this chapter. These numbers of wavelengths per link are most common

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

in today's WDM networks, which are based on ring or mesh topologies, though C can be as large as 160 in some ultra long-haul point-to-point WDM links.

The traffic model assumed in our performance study is as follows. Connection requests between each pair of network nodes arrive at a uniform rate from a Poisson process. The duration of each connection is exponentially distributed with a mean value of 1 unit time. The traffic load is measured on per source-destination pair basis. The network is modeled as follows. Each node capable of wavelength conversion has enough converters to convert all incoming wavelengths to outgoing wavelengths without blocking. In SWC case, the network nodes of largest degrees are chosen to be equipped with wavelength conversion [41]. Wavelength assignment on each segment of a lightpath is performed independently, which means wavelength conversion may be used even when neighboring segments have a free wavelength in common. Unless otherwise stated, we assume that the networks are capable of SWC SP-LCP routing and random wavelength assignment are used, and there are $C = 8$ wavelengths per link.

To quantify the benefit of advertising wavelength availability, we define the *blocking performance gain* when link-state update interval $T = x$ as the difference in decibel between the blocking probability when $T = x$ and the blocking probability when $T = \infty$. Let $B_{T=x}$ denote the blocking probability when $T = x$, then we have the blocking performance gain defined as follows:

$$G_{T=x} = 10 \cdot \log_{10} \left(\frac{B_{T=\infty}}{B_{T=x}} \right). \quad (4.1)$$

$G_{T=x}$ is an important performance indicator in our study.

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.2: Performance for Fixed Shortest-Path (FSP) routing with SWC, PacNet, and $C = 8$.

4.1.1 Different traffic loads

We plot the blocking probability of FSP routing against traffic loads in Fig. 4.2. The most important observation is that advertising wavelength availability information makes almost no difference under traffic loads heavier than 1 Erlang per source-destination pair¹, while a lot of difference under light traffic loads, e.g., 0.3 Erlang. This observation gives us an idea of the *domain* of link-state based lightpath establishment where advertising wavelength availability is useful. It is under light traffic load that the performance gain of advertising wavelength availability is significant, as shown in Fig. 4.3. Later, this conclusion will be verified in all cases. Fortunately, a wavelength-routed network in the real world generally operates under light traffic load where it can achieve an acceptable service quality of a low blocking probability.

¹Since there are 8 wavelengths per link, the network capacity is far from being used up under the traffic load of 1 Erlang per source-destination pair.

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.3: Performance gains for different routing methods with SWC, PacNet, and $C = 8$.

We also notice that a smaller update interval T leads to higher performance gain: $G_{T=0.1}$ is much closer to $G_{T=0}$ than $G_{T=1}$. However, unlike $G_{T=0}$, which always increases exponentially when traffic load decreases, $G_{T=0.1}$ has an inflexion point around 0.2 Erlang after which it no longer increases significantly. The inflexion point of $G_{T=1}$ is less obvious, but $G_{T=1}$ approaches its saturation around 4 dB when traffic load decreases to 0.01 Erlang. We offer the following explanation: If T is large, a significant amount of inaccurate link-state information causes the blocking probability to remain at a high level even when traffic load decreases; however, if T is smaller, the amount of inaccurate information is smaller as well, and the blocking probability can reduce to a lower level when traffic load decreases. A more detailed discussion on how the update interval T influences blocking performance could be found in the next chapter.

4.1.2 Different routing and wavelength assignment methods

Different routing methods use wavelength availability information in different ways, and result in different blocking performances when wavelength availability information is present. In Fig. 4.3, we plot the blocking performance gains for routing methods FSP, ASP, and SF-LCP. When accurate information of wavelength availability is provided, the improvement is significant for all routing methods under light traffic loads. Even FSP, which uses link-state information only for wavelength assignment, has significant performance gain under light traffic load. With $T = 0$, adaptive routing outperforms fixed routing under light traffic load. However, the difference becomes less apparent with a larger T , due to less reliable information for adaptive decision-making.

Comparing ASP and SF-LCP, we can see that choosing the least-congested path to break ties does improve the performance of SF-LCP, but only slightly in link-state based lightpath establishment. This is because most blocks under light traffic are caused by wavelength conflicts during the establishment process rather than by insufficient resource.

In Fig. 4.4, the performance of different wavelength assignment methods are compared. When the link-state information is accurate with $T = 0$, all three methods achieve nearly the same blocking probability. In other cases, where the link-state information is inaccurate, the random selection significantly outperforms the first-fit selection and the weighted selection further improves the performance over random selection, especially under heavy traffic load. Despite the performance improvement offered by the weighted wavelength selection, however, it is shown in Fig. 4.4(b) that the impact of link-state update remains significant. This is because all the wavelength selection methods have to use the link-state information to decide which wavelength is available before more sophisticated techniques can be applied to reduce the proba-

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

(a) Blocking performance

(b) Performance Gain.

Figure 4.4: Performance comparison of first-fit, random, and weighted wavelength assignment methods. With sparse wavelength conversion (SWC), Shortest-First Least-Congested-Path (SF-LCP) routing, PacNet, and 8 wavelengths per link.

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.5: Performance for different wavelength conversions with SF-LCP routing, PacNet, and $C = 8$.

bility of wavelength conflicts.

4.1.3 Different densities of wavelength conversions

Adding wavelength conversion in link-state based lightpath establishment has two opposing impacts on the blocking performance. It reduces blocking probability by relaxing the wavelength continuity constraint and allowing different wavelengths to be assigned independently in different segments of a lightpath, delimited by intermediate nodes with wavelength conversion. On the other hand, it could increase blocking probability, since in the presence of inaccurate link-state information, two lightpaths sharing multiple segments have multiple chances of wavelength conflict, whereas there is only one such chance in networks without wavelength conversion.

Under heavy traffic load, the first impact is dominating and FWC has the best

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.6: Performance gains for different wavelength conversions with SF-LCP routing, PacNet, and $C = 8$.

performance and NWC has the worst, as shown in Fig. 4.5. Under light traffic load, resources are sufficient and the first impact is less noticeable. Therefore, when T is large, the second impact becomes significant enough to make FWC having slightly worse performance than NWC. As shown in Fig. 4.6, the difference in blocking performance *gains* among FWC, SWC, and NWC is even less than among different routing methods.

In summary, the performance gain of advertising wavelength availability under light traffic load remains significant in all three cases, and it is not significantly affected by different densities of wavelength conversion (except when $T = 0$, which is equivalent to centralized control)

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.7: Performance gains for different network topologies with SWC, SF-LCP routing, and $C = 8$.

4.1.4 Different network topologies

The comparison of blocking performance gains in networks with different degrees of connectivities is shown in Fig. 4.7. From ring to mesh, networks with higher degrees of connectivity provide more choices of possible routes between each node pair, and hence give adaptive RWA computation more flexibility to make use of wavelength availability information. Therefore, although the performance gain in the 12-node ring network is already remarkable, PacNet and the 13-node mesh network get even better. Higher degree of connectivity is critical for better blocking performance, as well as better performance *gain* with wavelength availability information.

Similar to the case when the degree of network connectivity increases, a larger number of wavelengths supports more lightpaths simultaneously, and hence reduces the blocking probability significantly. In Fig. 4.8, the performance gains under dif-

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.8: Performance gains for different numbers of wavelengths per link with SWC, SF-LCP routing, and PacNet. The x-axis is zoomed out by 4 and 8 times for $C = 32$ and $C = 64$.

ferent numbers of wavelengths are shown together by zooming out the x-axis by 4 and 8 times for $C = 32$ and $C = 64$, respectively, where C denotes the number of wavelengths per link. After the traffic load is normalized with regard to C , the performance gains all increase when traffic load reduces, even with large update intervals. It is also shown that the number of wavelengths does not make significant difference when link-state information is inaccurate, such as in the case of $T = 1$. When link-state information is more up-to-date, e.g., when $T = 0.1$, a larger number of wavelengths offer larger performance gain, though such performance gain may become marginal when the traffic load becomes ultra-light.

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.9: Performance comparison for FSP routing with source wavelength assignment and segment-by-segment wavelength assignment. With SWC, PacNet, and $C = 8$.

4.1.5 Segment-by-segment wavelength assignment

In segment-by-segment wavelength assignment, the upstream end node of each segment is responsible for deciding the wavelength assignment in the segment, based on link-state information that is more accurate than what is available to the source node. Specifically, an upstream end node has accurate information about its neighboring links. If the segment consists of only a single link, then there is no stale link-state information used by its upstream end node. Nevertheless, even if every intermediate node has accurate information when making wavelength assignment decisions, such as that in the FWC case, the source node may still use stale information provided by link-state update when it computes the route adaptively. Therefore, the link-state update could still affect the performance significantly in segment-by-segment wavelength assignment.

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.10: Performance gains for different routing methods with segment-by-segment wavelength assignment. With SWC, PacNet, and $C = 8$.

Fig. 4.9 compares the blocking probability of FSP routing with source wavelength assignment and segment-by-segment wavelength assignment in the PacNet network with SWC. The accurate information at intermediate node does help to lower the blocking caused by stale information at source node and reduce the blocking probability to nearly an order lower than that of source wavelength assignment. Because $B_{T=\infty}$ is reduced while $B_{T=0}$ remains the same, the performance gain of immediate update $G_{T=0}$ is reasonably reduced.

We show the performance gains of FSP, ASP, and SF-LCP routing in Fig. 4.10. Compared with source wavelength assignment in Fig. 4.3, segment-by-segment wavelength assignment does reduce the performance gain for immediate update; in the presence of stale information, however, it accelerates the trend of the performance gain reaching the highest level when traffic load decreases. In addition, the performance gain of ASP routing is closer to FSP routing than to SF-LCP, which means

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

Figure 4.11: Performance gains for different routing methods with segment-by-segment wavelength assignment. With FWC, PacNet, and $C = 8$.

link-state information may offer a similar level of benefit to FSP routing and ASP routing when wavelength assignment is done in segment-by-segment manner.

This is more obvious when FWC is available in the network, as shown in Fig. 4.11. Only the results for traffic loads larger than 0.1 Erlang are shown in this figure, because the combined effect of FWC and segment-by-segment wavelength assignment is so significant that the blocking probability becomes extremely low under ultra-light traffic load. It is shown that link-state update makes no difference for FSP routing and ASP routing in this case. As aforementioned, link-state information is used by FSP only for wavelength assignment. When all intermediate nodes are capable of wavelength conversion and allowed to make wavelength assignment decisions by themselves, no remote link-state information is ever used in the lightpath establishment. Therefore, the blocking probability of $T = \infty$ is as low as $T = 0$ and the performance gain of link-state update equals to zero. With FWC, ASP routing only

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

differs from FSP routing when remote link-state information indicates that the shortest path used by FSP no longer has *any* free wavelength on one or more links, which is unlikely under light traffic load. Together with segment-by-segment wavelength assignment, this leads to the minimal impact of link-state update on ASP routing as well. Nevertheless, link-state update remains a major factor of the performance of SF-LCP routing. SF-LCP routing method totally depends on remote link-state information to compute the least-congested path so that the traffic load can be distributed among several routes with the same hop-length. Remote link-state information contributes significantly to the low blocking probability achieved by SF-LCP routing. Therefore, the performance gain remains significant for SF-LCP routing, though it is reduced by segment-by-segment wavelength assignment considerably.

4.2 Conclusions

In this chapter, we studied the benefits of advertising wavelength availability in a distributed lightpath establishment environment. Various routing methods, densities of wavelength conversions, and network topologies were studied by means of extensive simulations. We have shown that advertising wavelength availability information can provide significant performance gain only under light traffic load. Neither the degree of network connectivity nor the number of wavelengths per link leads to significant difference in performance gain. Moreover, we evaluated the impacts of the routing methods and the density of wavelength conversion in the cases of source wavelength assignment and segment-by-segment wavelength assignment respectively. We have shown that their impacts to the performance gain are marginal in most cases. For the special case where FWC and segment-by-segment wavelength are used at the same time, the performance gains of FSP and ASP routing reduce to nearly zero, while

CHAPTER 4. BENEFITS OF LINK-STATE UPDATE

that of SF-LCP routing remains significant due to load-balancing based on remote link-state information. The work in this chapter provides a solid ground for justifying the follow-up performance studies of link-state based lightpath establishment with a focus on enhancing network performance under light traffic load where the major objective is to reduce the effects of outdated information and keep the control overhead as low as possible.

CHAPTER 5

Periodic Link-State Update

Periodic update advertises the link state at regular intervals. It has predictable control overheads and guarantees a maximum age of the information used in RWA computations. Therefore, periodic update is a robust method commonly used in link-state based routing protocols. The performance of periodic update has been evaluated for QoS routing in IP and ATM networks [20, 44, 45]. Nevertheless, it has been demonstrated that the wavelength-continuity constraint substantially increases the network blocking probability and its sensitivity to traffic loads in wavelength-routed networks [11].

A number of previous works have analyzed the blocking probability in wavelength-routed networks for both centralized control [46–48] and distributed control [26] cases. In [47], *state-dependent arrival model* developed for circuit-switched

CHAPTER 5. PERIODIC LINK-STATE UPDATE

networks in [46] is extended to approximate the blocking probability in wavelength-routed networks. It assumes that the wavelength used on each link is independent of other wavelengths and links. The model has an exponential computational complexity and is shown to be accurate for small networks. The work in [48] uses the *reduced load approximation* scheme to avoid the exponential complexity while achieving the same accuracy as that in [47]. In addition, it proposes a *link-correlation* model to account for the correlation between wavelengths on neighboring links, and hence maintains a good accuracy for arbitrary traffic patterns and topologies. The work in [26] extends the link-correlation model to analyze distributed lightpath establishment where only limited link-state information is collected upon lightpath establishment. It evaluates the effects of outdated information caused by propagation delay and processing delay of control messages.

In this chapter, we evaluate the performance of periodic update method for link-state based lightpath establishment. Specifically, we analyze how outdated information caused by periodic update affects network blocking probability under different traffic loads. By modifying and extending the link-correlation model for distributed control in [26], we propose an accurate model to analyze the case where link-state information is outdated for a much longer duration than propagation and processing delays. Our extensive analytical and simulation results demonstrate the high sensitivity of blocking performance to link-state update interval under light traffic loads, as well as how this sensitivity is affected by network connectivity. Our study provides insightful understanding of the performance of periodic link-state update in wavelength-routed networks. The analytical model, with its low computational complexity and sufficient accuracy, would be useful for choosing a proper update interval to achieve good balance between blocking performance and control overhead.

This chapter is organized as follows. In Section 5.1, we provide a detailed analysis

of the fixed routing case and discuss the computational complexity of the model. Then we discuss the conditional applicability of our model to adaptive routing in Section 5.2. Numeric results and discussions are presented in Section 5.3. Finally, Section 5.4 concludes the chapter.

5.1 Analysis of the fixed routing case

5.1.1 Framework of analysis

The main idea of our analysis is based on the link correlation model [26, 48]. However, we take into account the outdated information that accumulates over a rather long duration between two sequential link-state updates as lightpaths being established and released.

On a given route R , let B_R denote the blocking probability of a connection request, B_R^R the routing blocking probability, and B_R^S the setup blocking probability. Following Equation (3.2) we have

$$B_R = 1 - (1 - B_R^R) \times (1 - B_R^S). \quad (5.1)$$

We make the following assumptions regarding the network and offered traffic: the network consists of J links connected in an arbitrary fashion, where each link has C wavelengths; connection requests between each pair of network nodes arrive according to a Poisson process at an average rate of λ_R , where the subscript R denotes the fixed route between the two nodes; the duration of each connection is exponentially distributed with a mean value of $\frac{1}{\mu}$; there is no wavelength converter in the network; propagation delay and processing delay of control messages can be neglected¹. We

¹Note that propagation delay and processing delay of control messages can cause outdated information and increase network blocking probability, which has been studied in [26]. In real world,

CHAPTER 5. PERIODIC LINK-STATE UPDATE

further assume that all nodes are synchronized to update simultaneously the state of neighboring links at the same interval of T . Meanwhile, random selection is assumed for wavelength assignment as aforementioned.

Let X_j ($j = 1, 2, \dots, J$) be the random variable representing the number of *free* wavelengths on link j . Let

$$q_j(m) = \Pr \{X_j = m\} \quad m = 1, 2, \dots, C \quad (5.2)$$

be the probability that exactly m wavelengths are free on link j . Following [48] we assume that all X_j 's are mutually independent, that is,

$$q(\mathbf{m}) = \prod_{j=1}^J q_j(m_j) \quad (5.3)$$

where

$$\mathbf{m} = (m_1, m_2, \dots, m_J). \quad (5.4)$$

We assume that given exactly m free wavelengths on link j , i.e., $X_j = m$, the time until the next connection setup on j is exponentially distributed with parameter $\alpha_j(m)$, called *state dependent arrival rate*. It then follows that the number of free wavelengths on link j can be modeled as the outcome of a birth-death process [46] such that

$$q_j(m) = \frac{\mu^m C(C-1) \dots (C-m+1)}{\alpha_j(1)\alpha_j(2) \dots \alpha_j(m)} \cdot q_j(0), \quad m = 1, \dots, C \quad (5.5)$$

link-state update interval is normally much longer than propagation and processing delay, and hence has the dominating impact on the blocking probability. Therefore, we assume that the propagation delay and processing delay of control messages can be neglected so as to focus solely on evaluating the impact of link-state update interval.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

where

$$q_j(0) = \left[1 + \sum_{m=1}^C \frac{\mu^m C(C-1) \dots (C-m+1)}{\alpha_j(1)\alpha_j(2) \dots \alpha_j(m)} \right]^{-1}. \quad (5.6)$$

When $X_j = m$, $\alpha_j(m)$ is obtained by combining the contributions from all the routes that pass through link j . We have

$$\alpha_j(m) = \sum_{R: j \in R} \alpha_j^R(m) \quad (5.7)$$

where

$$\alpha_j^R(m) = \begin{cases} 0 & m = 0 \\ \lambda_R(1 - B_{R|X_j=m}) & m = 1, 2, \dots, C \end{cases} \quad (5.8)$$

and $B_{R|X_j=m}$ is the conditional probability that a connection request on route R is blocked given $X_j = m$.

Based on the above steady-state model, the blocking probability B_R can be iteratively calculated as follows:

1. Initialize $B_{R|X_j=m} = 0, \forall j, m$.
2. Calculate $\alpha_j(m)$, $j = 1, \dots, J$ by (5.7) and (5.8).
3. Calculate $q(\mathbf{m})$ through equations (5.3), (5.5) and (5.6).
4. Calculate B_R^R and B_R^S as described in Section 5.1.2 and Section 5.1.3 below.
5. Calculate B_R by (5.1). If for each route R in the network B_R has converged, then stop; otherwise, go to step 6.
6. Re-calculate $B_{R|X_j=m}$ as described in Section 5.1.4 and go to step 2.

The above framework is similar to the one in [26], however, the calculations performed in step 4) and 6) are quite different. Below we will present the details of these two steps, i.e., the calculations of B_R^R , B_R^S , and $B_{R|X_j=m}$.

5.1.2 Calculating the routing blocking probability B_R^R

Routing blocking happens when the source node cannot find an RWA solution according to its local link-state database. Therefore, B_R^R equals to the probability that there is no *free* wavelength on all the links along route R .

We assume that all nodes perform RWA computation based on the same set of link-state information that is *accurate* at the moment of link-state update. This assumption simplifies the model and enables us to calculate B_R^R by using the same method as calculating the blocking probability in networks with centralized control [48].

Let g_i^R denote the probability that a given set of i wavelengths are *free* on route R at the moment of link-state update. From the inclusion-exclusion principle and the assumption of random wavelength assignment, we have

$$B_R^R = 1 - \sum_{i=1}^C (-1)^{i-1} \cdot \binom{C}{i} \cdot g_i^R. \quad (5.9)$$

Let H denote the number of links on route R . For symbolic convenience, let the links on route R be numbered as $1, 2, \dots, j, \dots, H$ in ascending order from source to destination. Therefore, the link one hop closer to the source than link j is denoted as link $j - 1, \forall j > 1$; while the link one hop closer to the destination than link j is denoted as link $j + 1, \forall j < H$.

Let $Y_{i,j}$ be the random variable representing the state of wavelength i on link j . Define

$$Y_{i,j} = \begin{cases} 0 & \text{if wavelength } i \text{ is free on link } j, \\ 1 & \text{if wavelength } i \text{ is busy on link } j. \end{cases} \quad (5.10)$$

Following the link correlation model [48], a set of assumptions as described below are adopted to account for link correlation, while keeping the complexity of calcula-

CHAPTER 5. PERIODIC LINK-STATE UPDATE

tion tractable:

- On a given route R , the state of a wavelength i on link j is independent of the state of any *other* wavelength k on link $j - 1$, given the state of the same wavelength i on link $j - 1$, or the state of wavelength k on the same link j .
- On a given route R , the state of a wavelength i on a link j is independent of the state of the *same* wavelength on link $j - 1$ or link $j + 1$, given the state of the wavelength on link $j-1$.

Let $g_{i,j}$ denote the steady-state probability that a given set of i wavelengths are *free* on link j . Based above assumptions on link correlation, it is derived in [48] that

$$g_i^R = g_{i,H} \cdot \prod_{j=1}^{H-1} \prod_{k=1}^i \left(1 + \gamma^{(j)} \cdot \left(\frac{1}{\eta_{k,j}} - 1 \right) \right)^{-1} \quad (5.11)$$

where $\gamma^{(j)}$ is the probability that a connection occupying wavelength λ on link j does not use link $j + 1$ and

$$\eta_{i,j} = \begin{cases} g_{i,j} & \text{if } i = 1, \\ \frac{g_{i,j}}{g_{i-1,j}} & \text{otherwise.} \end{cases} \quad (5.12)$$

From the definition of $q_j(m)$, we have

$$g_{i,j} = \sum_{m=i}^C q_j(m) \cdot \frac{\binom{m}{i}}{\binom{C}{i}} \quad (5.13)$$

and

$$\gamma^{(j)} = \frac{\sum_{\substack{R: j \in R \\ j+1 \notin R}} \sum_{m=1}^C \alpha_j^R(m) \cdot q_j(m)}{\sum_{m=1}^C \alpha_j(m) \cdot q_j(m)}. \quad (5.14)$$

5.1.3 Calculating the setup blocking probability B_R^S

Setup blocking is caused by outdated information. For example, if RWA computation at the source node selects a wavelength that was advertised as free but has been occupied, a request will be blocked at an intermediate node during lightpath establishment procedure. In this sense, setup blocking in periodic link-state update should follow a similar model as blocking caused by propagation delay in [26]. However, update interval T is normally much larger than propagation delay along any route in the network. As a result, it is possible that some lightpaths are established and then released between two sequential updates, which was not a concern in [26]. Therefore, we have to take a different approach to calculate setup blocking probability so that sufficient accuracy over a wide range of T can be achieved.

When $T = 0$, link-state information is updated without any delay. Therefore, the link-state information is always accurate and $B_R^S = 0$. Hereafter we assume $T \neq 0$. If $H = 1$, i.e., the route consists of a single link, $B_R^S = 0$; otherwise,

$$B_R^S = 1 - \prod_{j=2}^H \omega_{j-1,j} \quad (5.15)$$

where $\omega_{j-1,j}$ denotes the conditional probability that a path request would *not* be blocked on link j given that it was not blocked on link $j - 1$. We have

$$\omega_{j-1,j} = \sum_{m=1}^C q_{j|1}(m) \cdot \prod_{\substack{R': j \in R' \\ j-1 \notin R'}} \left(1 - P_{R,R'|X_j=m}^B\right) \quad (5.16)$$

where $q_{j|i}(m)$ denotes the probability that m wavelengths are free on link j given that

CHAPTER 5. PERIODIC LINK-STATE UPDATE

a specific set of i wavelengths are free on this link,

$$q_{j|i}(m) = q_j(m) \cdot \frac{\binom{m}{i}}{\binom{C}{i}} \cdot \left(\sum_{m'=i}^C q_j(m') \frac{\binom{m'}{i}}{\binom{C}{i}} \right)^{-1}, \quad (5.17)$$

and $P_{R,R'|X_j=m}^B$ denotes the conditional probability that a connection request on route R is blocked at link j by another request on route R' , given that $X_j = m$.

$P_{R,R'|X_j=m}^B$ is directly related to the update interval T . Denote the moment when the last link-state update occurred as time 0. Then the next link-state update will happen at time T . Assuming that link j is a member link of both route R and route R' , we let t denote the moment a request on R arrives, t_a the moment a request on R' arrives, and t_r the moment a connection on R' , which was set up after time 0, is released, respectively. We have $0 \leq t \leq T$ and $0 \leq t_a \leq t_r$.

A request on R will be blocked on link j by a request on R' if and only if the request on R' fulfills the following conditions: 1) it arrived before time t ; 2) it reserved the same wavelength; 3) the connection it established is not released at time t . These three conditions can be expressed as follows:

$$\begin{aligned} P_b(x) &= \frac{1}{m} Pr(t_a \leq t \leq t_r | t = x) \\ &= \frac{1}{m} \cdot \int_0^x Pr(x - t_a \leq t_r - t_a) \cdot Pr(t_a = y) dy \\ &= \frac{1}{m} \cdot \int_0^x e^{-\mu(x-y)} \cdot \alpha_j^{R'}(m) e^{-\alpha_j^{R'}(m)y} dy \end{aligned} \quad (5.18)$$

where $\alpha_j^{R'}(m)$ denotes the state dependent arrival rate of R' .

CHAPTER 5. PERIODIC LINK-STATE UPDATE

Then $P_{R,R'|X_j=m}^B$ can be calculated as

$$\begin{aligned}
 P_{R,R'|X_j=m}^B &= \frac{\int_0^T P_b(x) \cdot Pr(t=x) dx}{Pr(t \leq T)} \\
 &= \frac{\int_0^T P_b(x) \cdot \alpha_j^R(m) e^{-\alpha_j^R(m)x} dx}{1 - e^{-\alpha_j^R(m)T}} \\
 &= \frac{\alpha_j^R(m) \alpha_j^{R'}(m) (\eta e^{-\zeta T} - \zeta e^{-\eta T} + \theta)}{m \zeta \eta \theta (1 - e^{-\alpha_j^R(m)T})}
 \end{aligned} \tag{5.19}$$

where $\zeta = \alpha_j^R(m) + \alpha_j^{R'}(m)$, $\eta = \alpha_j^R(m) + \mu$, and $\theta = \alpha_j^{R'}(m) - \mu$.

As we have mentioned earlier, the third condition reflected by $P_b(x)$ is reasonably ignored in [26], where a connection does not likely hold only for a duration shorter than propagation delay. For periodic link-state update, however, update interval T is usually comparable to average holding time of connections and this condition must be taken into account in order to achieve sufficient accuracy.

5.1.4 Calculating the conditional blocking probability $B_{R|X_j=m}$

Given the number of free wavelengths on link j is m , extending (5.1) we have

$$B_{R|X_j=m} = 1 - (1 - B_{R|X_j=m}^R) \cdot (1 - B_{R|X_j=m}^S). \tag{5.20}$$

Calculating the conditional blocking probabilities $B_{R|X_j=m}^R$ and $B_{R|X_j=m}^S$ is straightforward, given the equations we have derived throughout Section 5.1.2 and 5.1.3.

From (5.9), $B_{R|X_j=m}^R$ can be expressed as

$$B_{R|X_j=m}^R = 1 - \sum_{i=1}^m (-1)^{i-1} \binom{C}{i} g_{i|X_j=m}^R, \tag{5.21}$$

CHAPTER 5. PERIODIC LINK-STATE UPDATE

where

$$\begin{aligned}
 g_{i|X_j=m}^R &= g_{i,H} \cdot \prod_{k=1}^i \left(1 + \gamma^{(j)} \cdot \left(\frac{1}{\eta_{k,j|X_j=m}} - 1 \right) \right)^{-1} \\
 &\quad \cdot \prod_{\substack{n=1 \\ n \neq j}}^{H-1} \prod_{k=1}^i \left(1 + \gamma^{(n)} \cdot \left(\frac{1}{\eta_{k,n}} - 1 \right) \right)^{-1} \\
 &\hspace{15em} \text{if } j \neq H, \\
 &= g_{i,H|X_j=m} \cdot \prod_{n=1}^{H-1} \prod_{k=1}^i \left(1 + \gamma^{(n)} \cdot \left(\frac{1}{\eta_{k,n}} - 1 \right) \right)^{-1} \\
 &\hspace{15em} \text{if } j = H,
 \end{aligned} \tag{5.22}$$

and

$$\begin{aligned}
 g_{i,j|X_j=m} &= \frac{\binom{m}{i}}{\binom{C}{i}}, \\
 \eta_{i,j|X_j=m} &= \begin{cases} g_{i,j|X_j=m} & \text{if } i = 1, \\ \frac{g_{i,j|X_j=m}}{g_{i-1,j|X_j=m}} & \text{otherwise.} \end{cases}
 \end{aligned} \tag{5.24}$$

If the route R consists of a single link, $B_{R|X_j=m}^S = 0$; otherwise $B_{R|X_j=m}^S$ can be expressed as,

$$B_{R|X_j=m}^S = 1 - \prod_{j=2}^H \omega_{j-1,j|X_j=m} \tag{5.25}$$

where

$$\omega_{j-1,j|X_j=m} = \prod_{\substack{R': j \in R' \\ j-1 \notin R'}} \left(1 - P_{R,R'|X_j=m}^B \right) \tag{5.26}$$

5.1.5 Computational complexity

We now study the computational complexity of the analytical model presented above. Let $\hat{H}$ denote the maximal number of hops of any route in the network, and I denote the maximal number of routes transversing the same link. Computing B_R^R in (5.9) re-

quires $O(2HC) + O(C)$ operations; and computing B_R^S requires $O(JC^2I)$ operations as from (5.15)-(5.16).

5.2 Limited applicability to adaptive routing

By applying adaptive routing in an empty network, we would get a route between each pair of source-destination nodes. Let Γ denote the set of the routes between all source-destination node pairs. As will be shown later, the blocking performance is highly load-dependent: when traffic load decreases, routing blocking probability decreases drastically. Therefore, despite of using an adaptive algorithm, the solutions of adaptive routing fall into Γ in most cases. By using Γ as the fixed set of routes and assigning routing blocking probabilities $B_R^R = 0$ and $B_{R|X_j=m}^R = 0$ for every route R in Γ , the model presented for the fixed routing in Section 5.1 can be applied to adaptive routing without significant error under light traffic loads.

5.3 Numerical results and discussions

In this section, we present the results of both analysis and simulation for a variety of topologies. We will discuss the impact of link-state update interval on the blocking probabilities under different traffic loads. We will also compare the analytical results against the simulation results to study the accuracy of our analytical model.

Results of three topologies are presented here: the PacNet network, a 12-node ring, and a 16-node mesh torus. As shown in Appendix A, PacNet is a topology of inter-connected rings with 15 nodes and 21 links and has appeared in other routing and wavelength assignment studies [22, 26, 49]. The ring and the mesh torus are chosen to extend the coverage of our study to networks with both low and high connectivities.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

In addition to the assumptions in Section 5.1, we further assume that 1) every link in the network is composed of two fibers of opposite directions, with $C = 8$ wavelength channels per fiber; 2) all connections are bi-directional, i.e., when a connection reserves a wavelength in one fiber of a link, it also reserves the same wavelength in the fiber of opposite direction; 3) the traffic loads are uniformly distributed with $\lambda_R = \lambda$ and $\mu = 1$ between each pair of source-destination nodes; and 4) both fixed routing and adaptive routing use the shortest-path routing policy in their RWA computations.

5.3.1 PacNet

Both the analysis and simulation results of fixed routing in PacNet are plotted in Fig. 5.1(a). They match quite well with each other. It is shown that under light traffic loads, the blocking probability increases quickly when update interval T increases; whereas under heavy traffic loads, blocking performance is much less sensitive to the change of update interval. To figure out the reasons behind this observation, we plot routing blocking probability and setup blocking probability separately in Fig. 5.1(b). Generally, for a given traffic load, longer update interval causes more inaccurate information and larger setup blocking probability. Since long lightpaths are more likely to be blocked than short ones, there will be fewer established lightpaths and the average hop length will be shorter, which leads to lower utilization of network capacity and lower routing blocking probability. Under light traffic loads, many wavelengths in the network are free and most RWA computations find solutions without causing routing blocking. Therefore, network blocking probability reflects the dominating setup blocking with a high sensitivity to the update interval. When traffic load increases, setup blocking probability increases since link-state information is more likely to be outdated when there are more traffic arrivals on the links; however, routing blocking probability increases much faster than setup blocking probability because RWA

CHAPTER 5. PERIODIC LINK-STATE UPDATE

(a) Blocking probability vs. update interval.

(b) Routing blocking probability B^R and setup blocking probability B^S vs. update interval.

Figure 5.1: Results of analysis and simulation for PacNet with 8 wavelengths per link. Fixed routing case.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

computation have to follow the wavelength continuity constraint. Finally, when traffic load becomes heavy, routing blocking probability catches up with setup blocking probability and the overall network blocking probability would reflect the combination of inverse-proportional routing blocking probability and direct-proportional setup blocking probability as update interval increases. That explains why the sensitivity of network blocking probability to update interval becomes dramatically lower under heavy traffic loads.

In Fig. 5.2(a), we show the results of adaptive routing. We can see that the network blocking probability is more sensitive to update interval under heavy traffic loads than in fixed routing, for example, under traffic load of $\lambda = 0.2$. This is because adaptive routing has more flexibility in finding a feasible RWA solution and avoiding routing blocking than fixed routing. Consequently, when traffic load increases, routing blocking probability increases at a slower speed, making the sensitivity of the network blocking to update interval drop more slowly than in fixed routing, as we can see in Fig. 5.2(b).

When applied to adaptive routing, our analytical model demonstrates high accuracy under light traffic loads. Under heavy traffic loads with small update periods, however, the accuracy is not satisfactory because routing blocking probability can no longer be ignored. Considering the low-blocking requirement of operational wavelength-routed networks in real world, we expect that most of them would normally work under light traffic loads. Therefore, the proposed model may still be applicable in many cases without significant error.

The analytical model also achieves high computational efficiency. For PacNet with 8 wavelengths per fiber, our implementation in ANSI C takes only 24.718 seconds on a Pentium IV 2.4GHz PC to get all the 48 data points for fixed routing, and it takes 16.125 seconds for adaptive routing.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

(a) Blocking probability vs. update interval.

(b) Routing blocking probability B^R and setup blocking probability B^S vs. update interval.

Figure 5.2: Results of analysis and simulation for PacNet with 8 wavelengths per link. Adaptive routing case.

5.3.2 12-node ring

The results of fixed routing in the 12-node ring network are shown in Fig. 5.3. The conclusions drawn for PacNet basically apply to the ring network as well. However, accuracy of the analytical model becomes lower in the ring network than in PacNet, especially in analyzing routing blocking probability. This confirms previous research works that the error of the link correlation model becomes significant in sparse networks with strong wavelength correlation because the model only takes into account the correlation between two neighboring links.

In Fig. 5.4, results of adaptive routing are plotted for the ring network. The analytical results start deviating from simulation results under a smaller traffic load than that in PacNet. We can see that adaptive routing is only able to improve the blocking performance under medium traffic load such as $\lambda = 0.1$, where routing blocking probability is already comparable to setup blocking probability but there is still enough wavelength resources for adaptive routing to exploit. Under lighter or heavier traffic load, either routing blocking probability is too low to significantly affect overall blocking or there is no more free wavelength that satisfies the wavelength continuity constraint. In both cases, adaptive routing is unable to improve performance in noticeable magnitude over fixed routing in the ring network.

5.3.3 16-node mesh torus

In Fig. 5.5, we plot the results of fixed routing for the 16-node mesh torus network. Both routing blocking probability and setup blocking probability is smaller compared with PacNet and the ring network, and the accuracy of analysis also improves. This is because the higher connectivity of mesh torus network diversifies the routes between node pairs over larger number of links. Consequently, it reduces the probability of

CHAPTER 5. PERIODIC LINK-STATE UPDATE

(a) Blocking probability vs. update interval.

(b) Routing blocking probability B^R and setup blocking probability B^S vs. update interval.

Figure 5.3: Results of analysis and simulation for 12-node ring network with 8 wavelengths per link. Fixed routing case.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

(a) Blocking probability vs. update interval.

(b) Routing blocking probability B^R and setup blocking probability B^S vs. update interval.

Figure 5.4: Results of analysis and simulation for 12-node ring network with 8 wavelengths per link. Adaptive routing case.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

(a) Blocking probability vs. update interval.

(b) Routing blocking probability B^R and setup blocking probability B^S vs. update interval.

Figure 5.5: Results of analysis and simulation for 16-node mesh torus network with 8 wavelengths per link. Fixed routing case.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

resource contention and weakens link correlation.

Another interesting observation on the mesh torus network is in Fig. 5.6, where the results of adaptive routing is presented. We see that the analysis remains accurate for the traffic load $\lambda = 0.3$, under which the analysis would have deviated significantly from simulation in other topologies. This is because the relatively higher network capacity can be exploited by adaptive routing to reduce routing blocking probability so that the assumption of $B_R^R = 0$ we made in Section 5.2 remains as an acceptable approximation.

5.3.4 Comparison of networks with different connectivities

A comparison of blocking performance in different networks is shown in Fig. 5.7. Under light traffic loads, the blocking probability of the 12-node ring network remains sensitive to update interval and there is not much difference from the case of PacNet. Under heavy traffic loads, however, we notice that the blocking probability in the ring network is slightly less sensitive to update interval than in PacNet due to higher routing blocking probability. In the 16-node mesh torus network, on the other hand, though the blocking performance is always better than those in PacNet and ring network regardless of traffic loads and update intervals, the sensitivity remains nearly the same as that in PacNet almost everywhere. The only exception is that under light traffic loads with short update interval, the sensitivity in mesh torus network is slightly lower than that in PacNet. This is because higher network connectivity leads to slightly shorter average hop length and marginally stronger tolerance to the outdated link-state information.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

(a) Blocking probability vs. update interval.

(b) Routing blocking probability B^R and setup blocking probability B^S vs. update interval.

Figure 5.6: Results of analysis and simulation for 16-node mesh torus network with 8 wavelengths per link. Adaptive routing case.

CHAPTER 5. PERIODIC LINK-STATE UPDATE

Figure 5.7: Comparing the blocking performances of the three networks. 8 wavelengths per link, fixed routing, simulation results only.

5.4 Conclusions

In this chapter, we studied the impacts of periodic link-state update on the blocking performance of wavelength-routed networks. We showed that blocking performance is highly sensitive to inaccurate information under light traffic loads. We presented an accurate analysis that extends the link correlation model to evaluate the effects of outdated link-state information accumulated over a long duration. The model is accurate for fixed routing as evident from its close agreement with the simulation results, and it can tackle adaptive routing cases under low traffic as well. Our work provides fresh insights into the complicated relationship between periodic update and network blocking performance, and may serve as a basis for future developments of efficient periodic link-state update schemes in wavelength-routed networks.

CHAPTER 6

Triggered Link-State Update

Triggered update advertises the link state once certain preset conditions are met, typically when the accumulated changes of link state since the last update exceeds a certain threshold. Compared with periodic update, triggered update sets up a bound for link-state changes between two consequent updates, which leads to better flexibility in handling dynamic traffic loads under certain circumstances. On the other hand, its blocking performance and control overhead are less predictable and can be significantly affected by detailed designs of the update trigger, as we will show in this chapter.

Since it is hard to study the triggered update analytically as the periodic update¹,

¹The stochastic blocking model for the triggered update is no longer a Markov chain as that for periodic update. The key difficulty relies on the fact that the probability of the link-state goes outdated is dependent on the history of when and how the link-state has changed, and thus the probability of wavelength conflict is also dependent on historical link-states.

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

we evaluate four different triggers through extensive simulations. We show that triggered update has its own features that need to be considered carefully, despite sharing some common features with periodic update. Specifically, similar to the high sensitivity of network performance to update period under light traffic load, network blocking performance remains to be highly sensitive to trigger threshold as well; however, unlike the periodic update where shorter period generally leads to better performance, smaller threshold values may lead to worse performance under certain circumstances; in addition, different triggers lead to significantly different network blocking performances. Understanding these characteristics would be helpful to develop more efficient update triggers in future.

This chapter is organized as follows. In Section 6.1, we will provide a general introduction to link-state update trigger, followed by descriptions of the four specific triggers. The evaluation method will then be discussed for link-state update triggers. Extensive numerical results and discussions will be presented in Section 6.2. Finally, Section 6.3 concludes the chapter.

6.1 Link-State Update Triggers

6.1.1 A General Description

In triggered update, a link state would be advertised once the change since last update has exceeded a certain threshold. Generally, the link state refers to the set of free wavelengths on the link. By defining the latest advertised link state as the *link state image*, the condition to trigger a new update could be expressed in terms of a measurement function of the difference between link state and its image. Specifically, for a link j , denote the link state as S_j and the link-state image as $\hat{S}_j$. Then a link-state

 CHAPTER 6. TRIGGERED LINK-STATE UPDATE

update would be triggered once

$$F(S_j, \hat{S}_j) \geq th, \quad (6.1)$$

where F denotes the measurement function and th denotes the preset threshold.

There are two different types of link-state changes. When the set of free wavelengths on a link is changed but the *number* of free wavelengths remains the same, we call such type of changes the *pattern changes*. For example, if one wavelength becomes busy and later another wavelength is released, the overall effect is a pattern change. In the case where the number of free wavelengths is also changed, the link state change is said to be an *availability change*. Therefore, a pattern change happens when

$$S_j \neq \hat{S}_j \quad \text{and} \quad X_j = \hat{X}_j, \quad (6.2)$$

while an availability change happens when

$$X_j \neq \hat{X}_j, \quad (6.3)$$

where X_j and $\hat{X}_j$ denote the number of free wavelengths in S_j and $\hat{S}_j$, respectively, i.e.,

$$\begin{cases} X_j = |S_j|, \\ \hat{X}_j = |\hat{S}_j|. \end{cases} \quad (6.4)$$

In conventional IP or ATM networks, only the availability changes would affect the network performance. In wavelength-routed networks, however, due to the wavelength continuity constraint, the pattern changes can also significantly affect the network performance, as we would see later. Next we will introduce several specific triggers defined by pattern changes and availability changes, respectively.

6.1.2 Specific Link-State Update Triggers

Link-state update triggers have been widely adopted in conventional networks [50], and have been applied in optical networks [14,21] as well. Two link-state update triggers are proposed in [21] for wavelength-routed networks, namely *Absolute Threshold Trigger* (ATT) and *Relative Threshold Trigger* (RTT). Both of them are adapted from popular triggers for IP/ATM networks and hence both have the measurement function $F(S_j, \hat{S}_j)$ defined based on availability changes. Specifically, ATT advertises the link state S_j when

$$\delta_a \geq th_a, \quad (6.5)$$

where th_a is the ATT threshold and the absolute availability change δ_a is defined as

$$\delta_a = |X_j - \hat{X}_j|. \quad (6.6)$$

RTT advertises S_j when

$$\delta_r \geq th_r, \quad (6.7)$$

where th_r is the RTT threshold and the relative availability change δ_r is defined as

$$\delta_r = \begin{cases} \frac{\delta_a}{\hat{X}_j} = \frac{|X_j - \hat{X}_j|}{\hat{X}_j} & \hat{X}_j \neq 0, \\ \infty & \hat{X}_j = 0. \end{cases} \quad (6.8)$$

To capture pattern changes in wavelength-routed networks, we propose two new triggers based on the number of link-state changes N_a rather than just on the number of free wavelengths. The *Absolute Counter Threshold Trigger* (ACTT) advertises the link state when

$$N_a \geq th_{ac} \quad (6.9)$$

 CHAPTER 6. TRIGGERED LINK-STATE UPDATE

where th_{ac} is the ACTT threshold. The *Relative Counter Threshold Trigger* (RCTT) advertises the link state when

$$N_r \geq th_{rc} \quad (6.10)$$

where th_{rc} is the RCTT threshold and the *relative counter* N_r is defined as

$$N_r = \begin{cases} \frac{N_a}{\hat{X}_j} & \hat{X}_j \neq 0, \\ \infty & \hat{X}_j = 0. \end{cases} \quad (6.11)$$

Finally, to provide a benchmark for evaluating the above triggers, we define the *accurate trigger* as the trigger that guarantees $\hat{S}_j$ to be always the same as S_j . For comparison purpose, we call all the other triggers as *inaccurate triggers*.

6.1.3 Evaluating the Triggers

The link-state update method is typically evaluated by measuring its impact on the network blocking probability [20], i.e., routing blocking probability B_R and setup blocking probability B_S in particular. To determine how accurately a link-state update trigger has the real link state reflected, we need a statistical measurement of the difference between the link state and its image. Therefore, we introduce *link state distribution* $q_j(m)$ as the probability that $X_j = m$, and *link state image distribution* $\hat{q}_j(m)$ as the probability that $\hat{X}_j = m$, respectively. We have

$$\begin{cases} \sum_{m=0}^C q_j(m) = 1, \\ \sum_{m:m \in \langle \hat{X}_j \rangle} \hat{q}_j(m) = 1 \end{cases} \quad (6.12)$$

where C denotes the number of wavelengths on link j and $\langle \hat{X}_j \rangle$ denotes the set of the possible values of $\hat{X}_j$. As we will see later, the values of $\hat{X}_j$ may not cover all the

 CHAPTER 6. TRIGGERED LINK-STATE UPDATE

values of X_j in triggered update, i.e.,

$$\langle \hat{X}_j \rangle \subseteq \langle X_j \rangle, \quad (6.13)$$

where

$$\langle X_j \rangle = \{1, 2, \dots, C\}. \quad (6.14)$$

Link state image $\hat{S}_j$ becomes inaccurate once link-state S_j changes; it remains inaccurate until the next update. Any connection request arrives when $\hat{S}_j$ is inaccurate may suffer blocking. To measure the inaccuracy of the link state image, we define *update rate* as the ratio of the number of updates to the number of link-state changes. Alternatively, the update rate could be viewed as the normalized average number of link-state updates per unit time, where the update rate of the accurate trigger is set as 1.

The purpose of our evaluation is to build the relationship between link-state update policy and blocking performance through the significance of inaccurate information. The mismatching between the distributions of link state and its image, as well as the link-state update rate, are important measurements of inaccurate information that help us to investigate the network blocking performance with various trigger thresholds under different traffic loads.

6.2 Numerical Results and Discussions

We carry out simulations on the PacNet network shown in Appendix A. We assume that each link has two fibers of opposite directions with $C = 8$ wavelengths per fiber. As in most real networks, we assume that all connections are bi-directional, which means when a connection reserves a wavelength in one fiber of a link, it also reserves

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

the same wavelength in the fiber in the opposite direction. We assume that there is no wavelength conversion in the network. Between each pair of source-destination nodes, the connection requests arrive from a Poisson process at an average arrival rate of λ . Each connection request has an exponentially distributed duration with an average of 1 unit time. We assume that the RWA calculation uses adaptive shortest-path routing and random wavelength assignment.

The distributions $q_j(m)$ and $\hat{q}_j(m)$ are measured as follows. We approximate $q_j(m)$ by measuring the ratio of the duration that link j stays at state $X_j = m$ to the overall simulated duration. $\hat{q}_j(m)$ is approximated in the same way by measuring the link state image.

Finally, note that the propagation delay and the processing delay of signaling messages can cause inaccurate information during the lightpath establishment procedure and increase network blocking probability, which has been studied in [51]. In this chapter, we neglect the propagation delay and the processing delay to eliminate their influences so that the evaluations would focus solely on the impacts of link-state update policy.

Below we will study the four specific triggers one by one where the accurate trigger, which yields zero setup blocking probability and has the overall blocking $B = B_R$, will be used as a benchmark.

6.2.1 Absolute Threshold Trigger

ATT establishes an absolute threshold for the variation of X_j with respect to its image $\hat{X}_j$. If this threshold is exceeded, a link-state update is triggered. ATT guarantees that at any moment the number of free wavelengths is bounded within the range

$$\hat{X}_j - th_a < X_j < \hat{X}_j + th_a. \quad (6.15)$$

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

When $th_a = 1$, ATT is accurate. When $th_a \geq 2$, ATT is inaccurate and only updates a subset of link states, i.e.,

$$\langle \hat{X}_j \rangle \subset \langle X_j \rangle. \quad (6.16)$$

For example, when $th_a = 8$, $\langle \hat{X}_j \rangle = \{0, 8\}$. In other words, a link-state update would happen only when there are zero or eight available wavelengths on a link.

The overall network blocking probability is plotted in Fig. 6.1(a). The blocking performance of inaccurate trigger is not sensitive to the changes of the threshold. In addition, it is less sensitive to the traffic loads and remains at a much higher level even under light traffic loads, compared to the blocking performance of accurate trigger (i.e., $th_a = 1$). For example, we see a drastic degradation in network blocking performance under light traffic load of $\lambda = 0.1$ when the threshold increases from 1 to 2; and the blocking probability stays constantly high when the threshold further increases.

The overall network blocking probability is decomposed into routing blocking probability B_R and setup blocking probability B_S in Fig. 6.1(b). It is shown that the setup blocking probability is fairly constant for all the inaccurate triggers (with $th_a \geq 2$). The reason can be found in Fig. 6.2, where we plot the update rate: when the threshold changes ATT from accurate to inaccurate between $th_a = 1$ and $th_a = 2$, the update rate drops sharply to a low level. Such a slump in link-state update rate effectively raises setup blocking probability close to its highest level, while further increase of th_a would not significantly decrease update rate any more. Under all traffic loads, we see that the normalized update rate follows nearly the same trend, leading to a similar trend of setup blocking probability.

Generally, under a given traffic load, routing blocking decreases when setup block-

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

(a) High load-dependence: under heavy loads the network blocking remains fairly constant over a wide range of trigger thresholds, including the accurate trigger, while under light loads the difference between the accurate trigger and others is significant.

(b) Setup blocking probability B_S is less sensitive to traffic load than routing blocking probability B_R . B_S increases to nearly its highest level once the trigger is no longer accurate. Routing blocking probability is small and out of the figure's range under light traffic load, thus is not shown for $\lambda = 0.1$ and $\lambda = 0.05$.

Figure 6.1: Blocking performance of Absolute Threshold Triggers (ATT), PacNet, 8 Wavelengths per link.

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

Figure 6.2: Link-state update rates of Absolute Threshold Triggers (ATT), PacNet, 8 wavelengths per link.

ing increases, as we can see in Fig. 6.1(b). This is because larger setup blocking leads to fewer established lightpaths; meanwhile, it tends to block long lightpaths rather than short ones, which results in shorter average hop length. Combined together, fewer lightpaths and shorter average hop length lead to lower utilization of the network capacity, or equivalently, larger average X_j on each link. Larger average X_j in turn leads to a lower routing blocking probability.

Nevertheless, routing blocking probability is *not* always decreasing with an increasing setup blocking. There is another factor influencing routing blocking probability, which is the difference between the link state and the link state image. Specifically, if $\hat{X}_j < X_j$, RWA computation takes some free wavelengths as occupied, thus routing blocking probability tends to be higher. If overall the probability of $\hat{X}_j < X_j$ becomes significant enough to override the effects of a larger X_j (which we described in last paragraph), routing blocking probability will increase. This kind of situation

 CHAPTER 6. TRIGGERED LINK-STATE UPDATE

is likely to happen when (1) the load is high enough, such that the routing blocking probability could be significantly affected by inaccurate link-state information², and (2) setup blocking is already high and would not change much with trigger threshold, thus the average X_j is less likely to be significantly increased by higher setup blocking. An example of this situation can be observed in Fig. 6.1(b): under heavy load $\lambda = 0.5$, routing blocking probability increases when $th_a \geq 5$ before it finally drops at $th_a = 9$ (where no link-state update is advertised and routing blocking happens only on single-hop routes). In Table 6.1, the distributions of $q_j(m)$ and $\hat{q}_j(m)$ of a heavily-loaded link are listed for the cases where $th_a = 5$ and $th_a = 8$. Some simple calculations tell us that the probability where $\hat{X}_j < X_j$ with $th_a = 8$ is significantly larger than that with $th_a = 5$, which explains the higher routing blocking probability with $th_a = 8$.

6.2.2 Relative Threshold Trigger

RTT establishes a relative threshold for the variations of X_j with respect to its image $\hat{X}_j$. RTT guarantees that at any moment the number of free wavelengths is bounded within the range

$$\hat{X}_j \cdot (1 - th_r) < X_j < \hat{X}_j \cdot (1 + th_r). \quad (6.17)$$

When $th_r \leq \frac{1}{C}$, RTT is accurate; otherwise, it is inaccurate. An inaccurate RTT requires smaller δ_a to trigger an update when $\hat{X}_j$ is smaller. For example, given $th_r = \frac{1}{6}$, $\delta_a = 2$ is required to trigger an update when $\hat{X}_j = 7$, whereas $\delta_a = 1$ is sufficient to trigger an update when $\hat{X}_j = 6$. Since $\hat{X}_j$ tends to be smaller on a heavily-loaded link, RTT is likely to capture more changes and trigger more updates on such links. Therefore, RTT is adaptive to dynamic link state and discriminates

²Under very low traffic loads, even with inaccurate link-state information, an RWA computation may still be able to find a feasible solution, since most wavelengths on the links are available.

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

Table 6.1: Example $q_j(m)$ and $\hat{q}_j(m)$ for Absolute Threshold Trigger, PacNet

Link			Between nodes 9,12	
No. paths passing through			48	
λ	th_a	m	$q_j(m)$	$\hat{q}_j(m)$
0.5	8	0	0.035626	0.960546
		1	0.1146	0.000
		2	0.186051	0.000
		3	0.232546	0.000
		4	0.218279	0.000
		5	0.137511	0.000
		6	0.057965	0.000
		7	0.015339	0.000
		8	0.002083	0.039454
		$Pr(\hat{X}_j > X_j)$	0.039372	
		$Pr(\hat{X}_j < X_j)$	0.926326	
	5	0	0.033565	0.000
		1	0.12034	0.000
		2	0.194357	0.000
		3	0.241763	0.990522
		4	0.208078	0.000
		5	0.133509	0.000
		6	0.055259	0.000
		7	0.012121	0.000
		8	0.001007	0.009478
		$Pr(\hat{X}_j > X_j)$	0.354430	
		$Pr(\hat{X}_j < X_j)$	0.406088	

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

links with different loads.

As we can see in Fig. 6.4, the average update rate of RTT decreases more gradually under heavy traffic loads than under light traffic loads. This is because more links have a small X_j under heavy traffic loads, and hence more link-state updates are triggered when a same number of link-state changes occur. Consequently, as shown in Fig. 6.3(b), setup blocking probability and routing blocking probability change gradually with the RTT threshold under heavy traffic loads. Another interesting observation from Fig. 6.3(b) is that the higher update rates sometimes make the overall network blocking probability under high traffic loads even lower than that under low traffic loads, especially when the threshold has a small value. An example is available in Fig. 6.3(a): the overall blocking probability is lower when $\lambda = 0.2$ than when $\lambda = 0.05$ at the thresholds $th_r = \frac{1}{7}$ and $th_r = \frac{1}{6}$.

6.2.3 Absolute and Relative Counter Threshold Triggers

By triggering on the number of link-state changes, the two counter threshold triggers we proposed can capture pattern changes in addition to availability changes. As a result, with the same threshold, a counter threshold trigger increases the number of updates and reduces setup blocking probability. As we observe in Fig. 6.5 and Fig. 6.6, counter triggers ACTT and RCTT each has improved blocking performance over ATT and RTT respectively, especially under light traffic loads. Under heavy traffic loads, even the difference between the performances of accurate trigger and inaccurate triggers is marginal. Therefore it is not surprising that ACTT and RCTT only slightly outperform ATT and RTT respectively.

It is interesting to note that the blocking performance of RCTT is fluctuating significantly under light traffic loads. This mainly comes from the fact that changes to or from a certain link state would never trigger any update under some threshold values,

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

(a) High load-dependence: under heavy loads the network blocking remains fairly constant over a wide range of trigger thresholds, including the accurate trigger, while under light loads the difference between the accurate trigger and others is significant.

(b) Setup blocking probability B_S is lower under heavy loads than under light loads for small thresholds.

Figure 6.3: Blocking performance of Relative Threshold Triggers (RTT), PacNet, 8 wavelengths per link.

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

Figure 6.4: Link-state update rates of Relative Threshold Triggers (RTT), PacNet, 8 wavelengths per link.

Figure 6.5: Comparisons between Absolute Counter Threshold Triggers (ACTT) and Absolute Threshold Triggers (ATT), PacNet, 8 wavelengths per link.

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

Figure 6.6: Comparisons between Relative Counter Threshold Trigger (RCTT) and Relative Threshold Triggers (RTT), PacNet, 8 wavelengths per link.

despite changes in the link state may happen frequently in the network. As a result, a lot of changes around this link state are not advertised, which degrades network blocking performance. Take the case of $th_{rc} = \frac{1}{5}$ as an example, there will never be a link-state update when $\hat{X}_j = 7$. Specifically, if $\hat{X}_j = 8$ or $\hat{X}_j = 6$, when the link state changes to $X_j = 7$ with $N_a = 1$, no update would be triggered since both the relative counters $N_r = \frac{1}{8}$ and $N_r = \frac{1}{6}$ are smaller than the threshold $th_{rc} = \frac{1}{5}$; while a sequential change from $X_j = 7$ will trigger an update with $N_a = 2$. For any other link-state image of $\hat{X}_j \leq 5$, on the other hand, a single change with $N_a = 1$ would have triggered an update before the link state could ever reach $X_j = 7$. Therefore, RCTT will never advertise link state $X_j = 7$, which happens to be one of the major states under light traffic loads. For other thresholds such as $th_{rc} = \frac{1}{6}$ or $th_{rc} = \frac{1}{4}$, on the contrary, the state $X_j = 7$ can always be updated with no more than two sequential changes, which results in better setup blocking performance as shown in Fig. 6.6.

CHAPTER 6. TRIGGERED LINK-STATE UPDATE

Similar analysis can be applied to other thresholds to explain the fluctuation of setup blocking under light traffic loads.

Under heavy traffic loads, since most links are heavily-loaded with small X_j values. Most link state changes, with a small X_j as the denominator, would be sufficient to trigger an update for all the thresholds. Therefore there is no fluctuation in blocking performance under heavy traffic loads.

6.3 Conclusions

In this chapter, we studied the impacts of link-state update triggers on the blocking performance of wavelength-routed networks. By evaluating four specific link-state update triggers in details, we showed that the blocking performance is highly sensitive to inaccurate information under light traffic loads and provided some explanations to such high sensitivity. Our work provides a framework of evaluating link-state update triggers in wavelength-routed networks, as well as some insights into the complicated relationship between the trigger thresholds, inaccurate information, and the network blocking performance.

CHAPTER 7

A Novel Method of Link-State Update

In previous two chapters, we have evaluated the performances of periodic update and triggered update. The timer-based periodic update advertises link state at preset intervals regardless of when the link state may change. Meanwhile, the change-based triggered update advertises link state without considering the duration between link-state updates. In our evaluations, it has been shown that these methods have drawbacks despite their advantages. The timer-based methods have predictable control overhead; while their major drawback is the amount of stale information introduced by the delay between link-state changes and updates. Even if the update interval is dynamically adjusted as in the so-called *fuzzy origination approach* (FOA) [23], the amount of stale information introduced by the delay between link-state changes and updates is inevitable. On the other hand, the blocking performance and control over-

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

head of change-based update methods are usually unpredictable as we have shown in Chapter 6.

Since any control packet that exceeds the capacity of the control channel will be dropped as that in other packet-switching networks, the link-state update methods must set their parameters properly following the traffic constraints assumed by the control network. A popular traffic model for packet-switching networks is the (ρ, σ) model [52], where ρ is the average rate and σ is the maximum burst size. Since the control network is shared by the routing protocol with many other protocols in GMPLS, the flow of link-state update messages usually has an average rate ρ much smaller than physical bandwidth of the control channel [21]. Meanwhile, because different protocols performs different functions and usually do not generate control traffic simultaneously, the control network can tolerate some burstiness of link-state updates, i.e., a relatively large σ . To the best of our knowledge, however, all the existing link-state update methods regulate only the average update rate but not the burstiness of the generated control traffic flow. Therefore, these methods tend to set their parameter conservatively to ensure no violation is committed. In particular, the only way for timer-based methods to tolerate more burstiness is to set smaller update intervals (either statically or dynamically), which is likely to increase the average rate at the same time. The change-based methods have better tolerance of burstiness, but they must sacrifice such tolerance significantly when depending on the hold-down timer to enforce the average rate. As a result, the blocking performance is compromised.

In this chapter, we propose a novel change-based link-state update method. This method regulates both the average rate and the burstiness of the update rate at the same time. More significantly, under limited control bandwidth, our method puts the stale link-state information of more negative impact at a higher priority to be re-

moved, and therefore further improves the blocking performance. As will be shown by comprehensive performance evaluations later in this chapter, the proposed method successfully enforces the control bandwidth quota while offering much lower blocking probability than existing link-state update methods.

The chapter is organized as follows. In Section 7.1, we will present our proposed method and discuss the important observations that inspired the invention of the method. Extensive numerical results of performance evaluations will be discussed in Section 7.2. Finally, Section 7.3 concludes the chapter.

7.1 A novel link-state update method

We first propose the *Generic Update Rate Algorithm* (GURA) to regulate link-state update rate explicitly according to the (ρ, σ) model of control bandwidth. The GURA is a straightforward application of the well-known Generic Cell Rate Algorithm [53] for B-ISDN networks. Specifically, it has two parameters: an *increment* denoted as T , and a *limit* denoted as τ . Both T and τ are in units of time and the algorithm is denoted as $GURA(T, \tau)$. Intuitively, the link-state should be advertised with T time units spacing, but the link-state shall be advertised if a change happens at least $T - \tau$ time units after the last link-state update. The flow chart of the GURA is shown in Fig. 7.1. The algorithm updates a Theoretical Arrival Time (TAT), which is the expected arrival time of the next link-state change assuming the link state changes at a constant interval of T . If the actual arrival time of the change is not “too early” relative to the TAT , in particular if the actual arrival time is after $TAT - \tau$, then the link state change is advertised and TAT is updated accordingly; otherwise the link-state change will not be advertised and TAT remains unchanged. The average rate ρ and the maximum burst size σ can be calculated from the parameters (T, τ) of

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.1: The generic update rate algorithm (GURA)

GURA algorithm:

$$\rho = \frac{k}{T}, \quad \text{and} \quad \sigma = k \cdot \left(\left\lfloor \frac{\tau}{T-t} \right\rfloor + 1 \right), \quad (7.1)$$

where k is the number of bits in a link-state update packet and t the transmission time of a single packet.

When the link state is not advertised, the wavelength becomes pseudo-surplus if the change is a reservation, or it becomes pseudo-deficient if the change is a release. It is important to understand whether pseudo-surplus and pseudo-deficient state information would degrade the network blocking performance with the same significance. If they would not, we would be able to *selectively* update link states to reduce the

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

amount of the stale information of greater significance and hence further improve the blocking performance of the GURA. For this purpose, we study four extreme cases of link-state update:

1. immediate update, where the link state is advertised immediately upon both wavelength reservations and releases, and the databases of network nodes contain neither pseudo-surplus nor pseudo-deficient information;
2. update upon reservation only, where the link state is advertised immediately upon wavelength reservations, therefore the databases of network nodes contain pseudo-deficient information only but no pseudo-surplus information;
3. update upon release only, where the link state is advertised immediately upon wavelength release, therefore the databases of network nodes contain pseudo-surplus information only but no pseudo-deficient information; and
4. no update, where link state is never advertised, and the link-state databases contain significant pseudo-surplus information but no pseudo-deficient information.

In Fig. 7.2, we plot illustrative simulation results of all these four cases¹. The blocking probability and its two components, the routing blocking probability B_R and setup blocking probability B_S are plotted. It is shown that Case 2 has a blocking probability close to the lowest value of Case 1, while Case 3 has a much higher blocking probability approaching the highest value of Case 4. This observation shows that pseudo-deficient information damages the blocking performance in a much smaller scale than pseudo-surplus information, especial under light traffic loads. We explain the reasons as follows: since routing blocking probability B_R is more sensitive to

¹Details of the simulation configuration will be given in Section 7.2. The results presented in this figure are for the PacNet network with SWC.

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.2: Different sensitivities of the blocking performance to pseudo-surplus information and pseudo-deficient information. Results are for PacNet with SWC.

traffic load than setup blocking probability B_S [54], B_R is much smaller than B_S under light traffic loads and contributes a less significant part of the overall blocking probability. Therefore, although pseudo-surplus information decreases B_R at the same time when it introduces B_S , B_S is much larger than B_R and the overall blocking probability increases significantly. On the other hand, pseudo-deficient information increases B_R and reduces the actual load on the links, thus the increment of B_R is offset by a decrement of B_S and the overall impact on blocking performance is much less significant than pseudo-surplus information.

Therefore, when we cannot update all link-state changes when they happen, we should encourage advertising more reservations than releases to reduce pseudo-surplus

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

information. In other words, we limit the number of updates upon wavelength releases when updates on wavelength reservations suffer loss. We call this the *event-discriminating* strategy and propose the Event-discriminating Update Rate Algorithm (EURA) for wavelength-routed networks. The basic idea is to add a token bucket in parallel to the $GURA(T, \tau)$ regulation. The tokens in the bucket can only be generated by updates upon wavelength reservations, yet can be consumed either by blocked updates upon wavelength reservations (due to limited control bandwidth) or by updates upon wavelength releases. Therefore, when the control bandwidth is abundant, link state is advertised upon most wavelength reservations, which generates enough tokens for advertising most wavelength releases. When the control bandwidth is limited, on the other hand, advertisements upon wavelength reservations get blocked, which consumes a large portion of the tokens generated. As a result, link state is less likely to be advertised upon wavelength releases and more control bandwidth is left for advertising upon wavelength reservations. Specifically, the method is as follows:

The number of tokens in the bucket is kept by the variable N , which is initially set as zero and is allowed to be negative. When a link-state change happens, the same (T, τ) regulation as that in the GURA is performed to make sure there is no violation to the control bandwidth quota. Moreover, if a wavelength reservation is coming later than TAT , it is advertised and one token is generated and added to the bucket by setting $N = N + 1$; if a wavelength reservation is coming earlier than $TAT - \tau$, it is not advertised and one token is removed from the bucket by setting $N = N - 1$. If the link-state change is a wavelength release, then in addition to passing the (T, τ) regulation, the link-state change can be advertised if and only if there is a positive number of tokens in the bucket, i.e., $N > 0$. Each advertisement of wavelength release consumes a token.

Since the generation and consumption of tokens in the EURA are controlled by

internal process rather than external parameters, the EURA has the same parameters of (T, τ) as those of the GURA. Hence we denote it as $EUR A(T, \tau)$. The flow chart of the $EUR A(T, \tau)$ is presented in Fig. 7.3. In the next section, we will evaluate the blocking performances of networks adopting the GURA and the EURA, respectively.

7.2 Performance evaluation

We carry out simulations on three topologies with various degrees of connectivity as shown in Appendix A: the PacNet, a 12-node ring, and a 13-node mesh. In sparse wavelength conversion case, network nodes with highest degrees are chosen to be equipped with wavelength conversion capability [41]. Each node capable of wavelength conversion has enough converters to convert all incoming wavelengths to outgoing wavelengths without blocking. Each link in the network has two fibers of opposite directions with 8 wavelengths per fiber. As in most real networks, we assume that all connections are bi-directional, which means when a connection reserves a wavelength in one fiber of a link, it also reserves the same wavelength in the fiber in the opposite direction. Between each pair of source-destination nodes, the connection requests arrive from a Poisson process at an average arrival rate of λ . Each connection request has an exponentially distributed duration with an average of 1 unit time. As aforementioned, we assume that the RWA computation uses the Shortest-First Least-Congested-Path (SF-LCP) algorithm [13, 33] for adaptive routing and random selection for wavelength assignment.

In addition to the GURA and the EURA, we study the cases of immediate update, periodic update, hold-down timer, the FOA, and no update. In the case of hold-down timer, link-state is updated immediately upon a link-state change if the duration since

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.3: The event-discriminating update algorithm (EURA)

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

the last update is longer than the hold-down timer.

The FOA employs a fuzzy inference system that consists of input membership functions, output membership functions, and a set of fuzzy rules. The input membership functions calculate a fuzzy value from the link state, i.e., the current number of available wavelengths. Then the fuzzy value is mapped by the fuzzy rules to an output fuzzy value, which in turn is calculated by the output membership functions into a waiting factor between two consecutive link-state updates. The FOA uses the Gaussian function $G(c, \sigma, x)$ as the output membership function, where c is the mean and σ is the standard deviation. The waiting factor from the Gaussian output membership function is then multiplied by the average inter-arrival time of lightpaths to calculate the interval between updates. We consider two configurations of the FOA proposed in [23], namely the $FOA(10)$ and the $FOA(0.1)$. The $FOA(10)$ is exactly the same as the FOA demonstrated in [23], using $G(0, 1.5, x)$ and $G(10, 1.5, x)$. The $FOA(0.1)$ is same as the $FOA(10)$ except that it uses $G(0, 0.015, x)$ and $G(0.1, 0.015, x)$ instead. Given the utilization of wavelength channels, the $FOA(0.1)$ outputs a waiting factor 100 times smaller than the $FOA(10)$. The details of the FOA can be found in [23].

Below we present extensive performance evaluations of the proposed link-state update methods. We will demonstrate the benefit of using the GURA to regulate link-state updates, as well as how the event-discriminating strategy helps EURA to improve the blocking performance further. Finally, we will also study the influence of various factors including (1) control bandwidth and tolerance of burstiness, (2) wavelength conversion, and (3) network connectivity on the blocking performance of the proposed algorithms.

7.2.1 Performance improvement over existing methods

In Fig. 7.4, we plot the blocking probabilities and the average update rates of different link-state update methods against varying traffic loads in the PacNet network with sparse wavelength conversion. It is shown that immediate update has the lowest blocking probability. Nevertheless, immediate update may not be practical in real world without using a hold-down timer to limit the update rate under heavy traffic load. The hold-down timer provides the basic function of rate regulating on link-state updates, and it is able to keep the update rate under a designed limit. However, because of its intolerance of burstiness, the hold-down timer cannot balance its performance under different traffic loads. Consequently, it has to bear the cost of getting very high blocking probability under light traffic load in order to limit the update rate under heavy traffic load, as illustrated by the hold-down timer of 0.1 unit time. Alternatively, it has to decrease the timer and allow the update rate to go very high under heavy traffic load in order to reduce the blocking under light traffic load, as illustrated by the hold-down timer of 0.01 unit time.

The result of periodic update with an interval of 0.1 is also plotted in the figure. Periodic update has constant update rate, but cannot achieve competitive blocking performance. As a variant of periodic update, the FOA improves blocking performance by adjusting the interval dynamically and reduces control bandwidth consumptions. As shown in Fig. 7.4, the $FOA(10)$ does not advertise the link state sufficiently and has very high blocking probability. In order to achieve lower blocking probability, the parameters of the FOA must be adjusted for more frequent updates. As demonstrated by the $FOA(0.1)$, however, the improvement of blocking performance comes at the cost of excessive updates under heavy traffic load. The aforementioned delay between link-state change and link-state update plays a critical role in preventing the FOA from achieving low blocking probability under light traffic loads.

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.4: Performance comparisons against existing link-state update methods. Results are for PacNet with SWC.

 CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

In Fig. 7.4, the $GURA(0.1, 0.4)$ significantly improves the blocking performance under light traffic load, while keeping the update rate within preset limit under heavy traffic load. It outperforms the hold-down timer 0.01 by nearly 2 orders lower in the blocking probability. Moreover, our proposed EURL improves the blocking performance even further over the GURA. In Fig. 7.4, the blocking probability of the $EURL(0.1, 0.4)$ is quite close to immediate update. When traffic load increases and update rate approaches the limit posed by control bandwidth, the EURL is also able to suppress update rate without exceeding the preset limit. Meanwhile, the EURL offers another advantage by always consuming less control bandwidth than the GURA.

In summary, the GURA has much lower blocking probability than the closest rival hold-down timer (0.01), especially under light traffic load. The EURL outperforms the GURA and is even more close to immediate update. When traffic load increases and update rate approaches the limit posed by control bandwidth, both algorithms are able to suppress update rate without exceeding the limit.

7.2.2 The impacts of control bandwidth and tolerance of burstiness

We evaluate the impact of control bandwidth on the performances of the EURL and the GURA. As shown for the PacNet with SWC in Fig. 7.5, the EURL performs essentially the same as the GURA when control bandwidth is abundant, e.g., when $T = 0.01$. In this case, seldom is any wavelength reservation not updated and the event-discriminating strategy is not active. When bandwidth becomes limited, however, the EURL puts reducing pseudo-surplus information at a higher priority and significantly outperforms the GURA.

When the control network only tolerates very small burstiness, it is shown in Fig. 7.6 that the GURA and the EURL have similar performances as the hold-down

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.5: The impact of control bandwidth. Results are for PacNet with SWC.

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.6: The impact of burstiness. Results are for PacNet with SWC.

timer. Indeed they degenerate to the hold-down timer when $\tau = 0$. When the control network tolerates some burstiness, which is usually the case in practice as aforementioned, both algorithms improve the blocking performance significantly.

7.2.3 The impact of wavelength conversion availability and network connectivity

When wavelength conversion is available in the network, wavelength continuity constraint is relaxed. Different wavelengths can be assigned to the lightpath on different segments delimited by intermediate nodes capable of wavelength conversion. Therefore, stale link-state information may cause more than one wavelength conflict on

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.7: The impact of wavelength conversion. Results are for PacNet.

multiple segments shared by two lightpaths. In other words, setup blocking probability becomes even more dominant². Consequently, as shown for the PacNet in Fig. 7.7, the reduction in setup blocking probability by the EURA over the GCRA is more significant with FWC and SWC than with NWC. In all cases of wavelength conversion, the EURA offers less consumption of control bandwidth, which remains a desirable advantage over the GURA.

In Fig. 7.8, we compare the blocking performance in networks with different con-

²Note that here we assume that the wavelength assignment on each segment is strictly decided by the source node. For the cases where some intermediate nodes can make their own wavelength assignment decisions, network blocking probability could be significantly lowered at a cost of more complicated network control and management (e.g. [55]). Such cases, however, are out of the scope of this thesis.

CHAPTER 7. A NOVEL METHOD OF LINK-STATE UPDATE

Figure 7.8: The impact of network connectivity. Results are for networks with SWC.

nectivities. The networks are capable of SWC. In the ring network, because lightpaths between any two nodes have only two options of routing, network nodes can gather a large amount of link-state information from the lightpaths passing through. Therefore, the ring network is less sensitive to stale information than the PacNet and the mesh network under the same traffic load [56]. We can see that the EURA and the GURA have similar blocking performances in the ring network as immediate update, while the EURA's improvements over the GURA become more significant in networks with higher connectivities. Again, the EURA consumes much smaller control bandwidth than the GURA and immediate update, giving itself an advantage as in all the other cases we have studied.

7.3 Conclusions

In this chapter, we proposed a novel method to regulate link-state update rate in wavelength-routed networks. We proposed the generic update rate algorithm (GURA) to actively regulate the link-state update rate to avoid overloading control channel with excessive update messages, while tolerating the inherent burstiness of link-state changes at the same time. Based on the GURA, we proposed the event-discriminating update rate algorithm (EURA). By giving pseudo-surplus information a higher priority to be removed than pseudo-deficient information, it further improves the network blocking performance. Comprehensive performance evaluations were conducted for the proposed algorithms, with discussions on the impacts of several different factors including control bandwidth and tolerance of burstiness, wavelength conversion, and network connectivity. It is shown that the proposed algorithms successfully enforce the control bandwidth quota while offering much lower blocking probability than existing link-state update methods.

CHAPTER 8

Conclusion

Continuously driving down the operational costs is critical to telecommunication service providers to keep competitive. In the past, WDM technology had helped to improve the utilization of the fibers significantly. Current optical networks fall short, however, in the face of the growing dynamic services that impose additional requirements on lightpath establishment. To take the challenge, routing protocols are introduced to provide dynamic, distributed, and on-demand wavelength services. The link-state based routing protocol offers the advantages of faster convergence, lower blocking probability, and better support of traffic engineering. Nevertheless, these benefits do not come without drawbacks. The link-state based routing protocol has to make trade-off so that the link-state update messages will not overload the control channels and cause control plane failure.

CHAPTER 8. CONCLUSION

This thesis investigates the performance-overhead trade-offs of link-state based routing in wavelength-routed networks. After characterizing the link-state based lightpath establishment with representative analytical models and extensive simulation-based evaluations, this thesis follows with a novel method for improving the efficiency and reliability of link-state update. The remainder of this chapter discusses the contributions of this thesis in more detail and presents some related problems that are subject to future research.

8.1 Research Contributions

This thesis contributes a number of new insights into the behavior of link-state based lightpath establishment. In Chapter 3, we present a general model to study the important parameters affecting the performance and overhead of link-state based lightpath establishment. The model clearly formulates the components of the routing and signaling model and allows further investigation of the fundamental trade-off between link-state update overhead and the network blocking performance. A significant component of our research, also described in Chapter 3, is the realization of the routing and signaling model in the simulation environment we developed to handle efficiently a variety of network and routing configurations. The simulator is object-oriented, fully extensible, and able to provide complete support within a unified architecture for various RWA algorithms, topologies, and statistics collections.

Recognizing that a solid basis justifying the follow-up studies must be provided first, the performance study in Chapter 4 constitutes the first in-depth investigation on the benefits of advertising wavelength availability in a distributed lightpath establishment environment. Extensive simulation results show that advertising wavelength availability information can provide significant performance gain, however, only un-

CHAPTER 8. CONCLUSION

der light traffic load. Moreover, in the presence of outdated information, especially when the link-state update interval is large, neither the density of wavelength conversion nor the routing methods could make significant difference to network performance; even a higher degree of network connectivity leads to only marginally larger performance gain. The study clarifies that future research on link-state based light-path establishment shall focus on enhancing network performance under light traffic load where the major objective is to reduce the effects of outdated information and keep the control overhead as low as possible.

Then that follows the study on the performance of previously known methods of link-state update in wavelength-routed networks. Chapter 5 and Chapter 6, respectively, presents the first evaluation of the blocking performances of periodic update and triggered update on a comprehensive range of configurations with different routing methods, wavelength conversions and network connectivities. In Chapter 5, an analytical model is proposed for the first time to evaluate how the stale information introduced by periodic update may affect the blocking performance. Extensive simulation results validate the accuracy of the model and show that the stale link-state information significantly degrades the blocking performance under light traffic loads. In Chapter 6, four different triggers are evaluated through extensive simulations. Similar to the high sensitivity to update period under light traffic load, network blocking performance remains to be highly sensitive to trigger threshold as well. Unlike the periodic update where shorter period generally leads to better performance, however, smaller threshold values may lead to worse performance under certain circumstances. Despite the flexibility in handling dynamic traffic loads, it is shown that triggered update does not provide sufficient resolution and reliable control of the trade-off between performance and overhead. Although the study of previously known methods of link-state update in these two chapters does provide us in-depth understanding of

CHAPTER 8. CONCLUSION

how the stale information may affect the blocking performance, the problem of how to fine-tune the routing protocol to minimize such impact remains open.

Chapter 7 provides the answer in the form of a novel link-state update method. Two algorithms are proposed to regulate link-state update rate in wavelength-routed networks. The generic update rate algorithm (GURA) actively regulates the link-state update rate to avoid overloading control channel with excessive update messages, while tolerating the inherent burstiness of link-state changes at the same time. Moreover, it is shown for the first time that pseudo-surplus information degrades the blocking performance more than pseudo-deficient information. Therefore, the event-discriminating update rate algorithm (EURA) is proposed to improve further the network blocking performance by giving pseudo-surplus information a higher priority to be removed. Comprehensive performance evaluations were conducted for the proposed algorithms, with discussions on the impacts of several different factors including control bandwidth and tolerance of burstiness, wavelength conversion, and network connectivity. It is shown that the proposed algorithms successfully enforce the control bandwidth quota while offering much lower blocking probability than previously known link-state update methods.

8.2 Future Research Directions

This thesis contributes new insights and solutions for improving the efficiency of link-state update in wavelength-routed networks. We have addressed the part of link-state based lightpath establishment that has the most impact on the network performance. Nevertheless, to improve the performance further, and probably to provide new functionality, a number of interesting research problems remain and require investigation. First, it may be possible to improve further the network performance by fine-tuning

CHAPTER 8. CONCLUSION

the RWA algorithms. Although the RWA algorithms may have smaller impact on the blocking performance compared with link-state update methods, as shown in Chapter 4, different RWA algorithms do have different capability in exploiting the link-state information and have different blocking performances. A more sophisticated wavelength assignment method, such as wavelength reassignment at intermediate nodes capable of wavelength conversion, may have the potential to reduce the probability of wavelength contention in distributed environment. Secondly, the flooding procedure of link-state update messages currently used in OSPF has not been optimized for wavelength-routed networks yet. A possible approach of improvement could be division of flooding areas separated by network nodes that are capable of wavelength conversion and reassignment. Careful study and design are required for such scheme to work properly, since the network nodes capable of wavelength-conversion and reassignment may not always be able to form a cut set to isolate the flooding area. Finally, distributed implementation of constraint-based routing in wavelength-routed networks needs to be studied carefully to determine its feasibility and scalability in terms of RWA computation complexity and the requirement for dynamic dissemination of optical impairment parameters.

Author's Publications

1. S. Shen, G. Xiao, and T. H. Cheng, "Evaluating the impact of the link-state update period on the blocking performance of wavelength-routed networks," in *Proc. Optical Fiber Communication Conference (OFC)*, vol. 2, Feb. 2004, pp. 131–133.
2. ———, "Evaluating link-state update triggers in wavelength-routed networks," in *Proc. Asia-Pacific Optical Communications Conference and Exhibition (APOC)*, Nov. 2004.
3. ———, "Benefits of advertising wavelength availability in distributed lightpath establishment," in *Proc. IEEE International Conference on Communications (ICC)*, May 16–20, 2005.
4. ———, "A novel method of link-state update in wavelength-routed networks," *J. Lightwave Technol.*, vol. 24, no. 3, pp. 1112–1120, Mar. 2006.
5. ———, "Benefits of advertising wavelength availability in distributed lightpath establishment," to appear in *Computer Networks*.

Bibliography

- [1] ITU-T Recommendation G.707, “Network node interface for the synchronous digital hierarchy (SDH),” 2003.
- [2] ANSI Standard T1.105-2001, “Synchronous optical network (SONET) - basic description including multiplex structure, rates, and formats,” 2001.
- [3] R. Ballart and Y. Ching, “SONET: Now it’s the standard optical network,” *IEEE Commun. Mag.*, vol. 27, no. 3, pp. 8–15, Mar. 1989.
- [4] P. E. Green, *Fiber-Optic Networks*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [5] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, 2nd ed. Morgan Kaufmann, 2001.
- [6] B. Mukherjee, “WDM-based local lightwave networks — Part I: Single-hop systems,” *IEEE Network*, vol. 6, no. 3, pp. 12–27, May 1992.
- [7] ———, “WDM-based local lightwave networks — Part II: Multihop systems,” *IEEE Network*, vol. 6, no. 4, pp. 20–32, July 1992.
- [8] P. E. Green, “Progress in optical networking,” *IEEE Commun. Mag.*, vol. 39, no. 1, pp. 54–61, Jan. 2001.
- [9] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath communications: an approach to high bandwidth optical WAN’s,” *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1171–1182, July 1992.
- [10] R. Ramaswami and K. N. Sivarajan, “Routing and wavelength assignment in all-optical networks,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 489–500, Oct. 1995.

BIBLIOGRAPHY

- [11] M. Kovacevic and A. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 868–880, June 1996.
- [12] H. Zang, L. Sahasrabuddhe, J. P. Jue, S. Ramamurthy, and B. Mukherjee, "Dynamic lightpath establishment in wavelength-routed WDM networks," *IEEE Commun. Mag.*, pp. 100–108, Sept. 2001.
- [13] 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, Jan. 2000.
- [14] R. Ramaswami and A. Segall, "Distributed network control for optical networks," *IEEE/ACM Trans. Networking*, vol. 5, pp. 936–942, Dec. 1997.
- [15] J. Zheng and H. T. Mouftah, "Distributed lightpath control based on destination routing for wavelength-routed WDM networks," in *Proc. IEEE GLOBECOM '01*, vol. 5, 2001, pp. 1526–1530.
- [16] E. Mannie *et al.*, "Generalized multi-protocol label switching (GMPLS) architecture," IETF RFC 3945, Oct. 2004.
- [17] J. Moy, "OSPF version 2," RFC 2328, Apr. 1998.
- [18] K. Kompella and Y. Rekhter, "OSPF extensions in support of generalized multi-protocol label switching," IETF Internet-Draft, work in progress, Oct. 2003.
- [19] G. Li, J. Yates, D. Wang, and C. Kalmanek, "Control plane design for reliable optical networks," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 90–96, Feb. 2002.
- [20] A. Shaikh, J. Rexford, and K. G. Shin, "Evaluating the impact of stale link state on quality-of-service routing," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 162–176, Apr. 2001.
- [21] S. Sengupta, D. Saha, and S. Chaudhuri, "Analysis of enhanced OSPF for routing lightpaths in optical mesh networks," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 5, 2002, pp. 2865–2869.
- [22] H. Zang, L. Sahasrabuddhe, J. P. Jue, S. Ramamurthy, and B. Mukherjee, "Connection management for wavelength-routed WDM networks," in *Proc. IEEE GLOBECOM '99*, vol. 2, 1999, pp. 1428–1432.

BIBLIOGRAPHY

- [23] A. I. Al-Fuqaha, G. M. Chaudhry, M. Guizani, and M. A. Labrador, "Routing framework for all-optical DWDM metro and long-haul transport networks with sparse wavelength conversion capabilities," *IEEE J. Select. Areas Commun.*, vol. 22, no. 8, pp. 1443–1459, Oct. 2004.
- [24] X. Yuan, R. Melhem, R. Gupta, Y. Mei, and C. Qiao, "Distributed control protocols for wavelength reservation and their performance evaluation," *Photonic Network Communications*, vol. 1, no. 3, pp. 207–218, 1999.
- [25] J. P. Jue and G. Xiao, "An adaptive routing algorithm for wavelength-routed optical networks with a distributed control scheme," in *Proc. IEEE IC3N*, 2000, pp. 192–197.
- [26] K. Lu, G. Xiao, and I. Chlamtac, "Analysis of blocking probability for distributed lightpath establishment in WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 1, pp. 187–197, Feb. 2005.
- [27] J. J. Garcia-Luna-Aceves, "Distributed routing with labeled distances," in *Proc. IEEE INFOCOM '92*, vol. 2, May 1992, pp. 633–643.
- [28] G. S. Malkin, "RIP version 2," RFC 2453, Nov. 1998.
- [29] C. Xin, Y. Ye, T. Wang, S. Dixit, C. Qiao, and M. Yoo, "On an ip-centric optical control plane," *IEEE Commun. Mag.*, vol. 39, no. 9, pp. 88–93, Sept. 2001.
- [30] J. Strand, A. L. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 81–87, Feb. 2001.
- [31] Y. Mei and C. Qiao, "Efficient distributed control protocols for WDM all-optical networks," in *Proc. IEEE IC3N '97*, Sept. 1997, pp. 150–153.
- [32] C. Xin, Y. Ye, S. Dixit, and C. Qiao, "An emulation-based comparative study of centralized and distributed control schemes for optical networks," *Optical Networks Magazine*, vol. 3, no. 5, pp. 44–51, 2002.
- [33] K. Chan and T. P. Yum, "Analysis of least congested path routing in WDM light-wave networks," in *Proc. IEEE INFOCOM '94*, vol. 2, Toronto, Ont., Canada, June 1994, pp. 962–969.

BIBLIOGRAPHY

- [34] S. Pitchumani, I. Cerutti, and A. Fumagalli, "Destination-initiated wavelength-weighted reservation protocols: Scalable solutions for wdm rings," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 3, June 20–24, 2004, pp. 1776–1780.
- [35] X. Wang, L. Pong, H. Morikawa, and T. Aoyama, "Dynamic lightpath establishment using prioritized wavelength assignment," in *IEICE General Conference*, Sept. 2002, pp. B–10–30.
- [36] J. Zhou and X. Yuan, "A study of dynamic routing and wavelength assignment with imprecise network state information," in *Proc. International Conference on Parallel Processing Workshops*, 2002, pp. 207–213.
- [37] R. Coltun, "The OSPF opaque LSA option," RFC 2370, July 1998.
- [38] D. Katz, K. Kompella, and D. Yeung, "Traffic engineering (TE) extensions to OSPF version 2," RFC 3630, Sept. 2003.
- [39] ISO Standard ISO/IEC 19501:2005, "Unified modeling language specification," 2005.
- [40] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-optical networks with sparse wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, pp. 544–557, Aug. 1996.
- [41] J. Iness and B. Mukherjee, "Sparse wavelength conversion in wavelength-routed WDM optical networks," *Photonic Network Communications*, vol. 1, no. 2, pp. 183–205, 1999.
- [42] G. Xiao, K. Lu, and I. Chlamtac, "An evaluation of distributed wavelength provisioning in WDM optical networks with sparse wavelength conversion," *J. Light-wave Technol.*, vol. 22, no. 7, pp. 1668–1678, July 2004.
- [43] J. Banks *et al.*, *Discrete-event System Simulation*, 2nd ed. Prentice Hall, 1999.
- [44] A. Ariza, E. Casilari, and F. Sandoval, "Strategies for updating link states in QoS routers," *IEE Electronic Letters*, vol. 36, no. 20, pp. 1749–1750, Sept. 2000.
- [45] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of service based routing: a performance perspective," in *Proc. ACM SIGCOMM '98*, vol. 18, no. 4, Oct. 1998, pp. 17–28.

BIBLIOGRAPHY

- [46] S.-P. Chung, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss network with state-dependent routing," *IEEE/ACM Trans. Networking*, vol. 1, pp. 105–115, Feb. 1993.
- [47] A. Birman, "Computing approximate blocking probability for a class of all-optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 15, pp. 852–857, June 1996.
- [48] A. Sridharan and K. N. Sivarajan, "Blocking in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 384–397, Apr. 2004.
- [49] R. Ramamurthy and B. Mukherjee, "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks," *IEEE/ACM Trans. Networking*, vol. 10, no. 3, pp. 351–367, June 2002.
- [50] J. Moy, "OSPF version 2," RFC 2328, Apr. 1998.
- [51] K. Lu, G. Xiao, and I. Chlamtac, "Blocking analysis of dynamic lightpath establishment in wavelength-routed networks," in *Proc. IEEE ICC '02*, vol. 5, Apr. 2002, pp. 2912–2916.
- [52] R. L. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.
- [53] ITU-T Recommendation I.371, "Traffic control and congestion control in B-ISDN," 2004.
- [54] S. Shen, G. Xiao, and T. H. Cheng, "Evaluating the impact of the link-state update period on the blocking performance of wavelength-routed networks," in *Proc. Optical Fiber Communication Conference (OFC)*, vol. 2, Feb. 2004, pp. 131–133.
- [55] K. Lu, J. P. Jue, G. Xiao, I. Chlamtac, and T. Ozugur, "Intermediate-node initiated reservation (IIR): a new signaling scheme for wavelength-routed networks," *IEEE J. Select. Areas Commun.*, vol. 21, no. 8, pp. 1285–1294, Oct. 2003.
- [56] S. Shen, G. Xiao, and T. H. Cheng, "Benefits of advertising wavelength availability in distributed lightpath establishment," in *Proc. IEEE International Conference on Communications (ICC)*, May 16–20, 2005.

APPENDIX A

Network Topologies

APPENDIX A. NETWORK TOPOLOGIES

Figure A.1: PacNet topology. It has 15 nodes and 21 links. Each link in the network is composed of two fibers of opposite directions. The numbers next to the links denote the physical lengths in 10 kilometers, which are used as the metric of calculating the shortest path. In SWC case, nodes 3, 4, 9, and 11 are capable of wavelength conversion.

Figure A.2: A 12-node ring network. In SWC case, nodes 0, 4, and 8 are capable of wavelength conversion.

APPENDIX A. NETWORK TOPOLOGIES

Figure A.3: A 13-node mesh network. In SWC case, nodes 5, 6, and 9 are capable of wavelength conversion.

Figure A.4: A 16-node mesh torus network.