

# Onlinezuweisung von Ressourcen in Dynamischen Optischen Netzen

Dissertation

zur Erlangung des akademischen Grades  
Doktor-Ingenieur

Vorgelegt  
der Fakultät für Elektrotechnik und Informationstechnik  
der Technischen Universität Chemnitz

von **M.Sc., Ronald Romero Reyes**

Chemnitz, den 24.04.2019



# Online Resource Allocation in Dynamic Optical Networks

## Dissertation

zur Erlangung des akademischen Grades  
Doktor-Ingenieur

Vorgelegt  
der Fakultät für Elektrotechnik und Informationstechnik  
der Technischen Universität Chemnitz

von **M.Sc., Ronald Romero Reyes**

Chemnitz, den 24.04.2019



## Bibliographische Beschreibung

Ronald Romero Reyes

Onlinezuweisung von Ressourcen in Dynamischen Optischen Netzen

Dissertation (in englischer Sprache)

162 Seiten, 61 Abbildungen, 31 Tabellen, 118 Literaturverweise

## Referat

Konventionelle, optische Transportnetze haben die Bereitstellung von High-Speed-Konnektivität in Form von langfristig installierten Verbindungen konstanter Bitrate ermöglicht. Die Einrichtungszeiten solcher Verbindungen liegen in der Größenordnung von Wochen, da in den meisten Fällen manuelle Eingriffe erforderlich sind. Nach der Installation bleiben die Verbindungen für Monate oder Jahre aktiv. Das Aufkommen von Grid Computing und Cloud-basierten Diensten bringt neue Anforderungen mit sich, die von heutigen optischen Transportnetzen nicht mehr erfüllt werden können. Dies begründet die Notwendigkeit einer Umstellung auf dynamische, optische Netze, welche die kurzfristige Bereitstellung von Bandbreite auf Nachfrage (Bandwidth on Demand - BoD) ermöglichen. Diese Netze müssen Verbindungen mit unterschiedlichen Bitratenanforderungen, mit zufälligen Ankunfts- und Haltezeiten und stringenten Einrichtungszeiten realisieren können. Grid Computing und Cloud-basierte Dienste führen in manchen Fällen zu Verbindungsanforderungen mit Haltezeiten im Bereich von Sekunden, wobei die Einrichtungszeiten im Extremfall in der Größenordnung von Millisekunden liegen können.

Bei optischen Netzen für BoD muss der Verbindungsaufbau und -abbau, sowie das Netzmanagement ohne manuelle Eingriffe vonstattengehen. Die dafür notwendigen Technologien sind Flex-Grid-Wellenlängenmultiplexing, rekonfigurierbare optische Add / Drop-Multiplexer (ROADMs) und bandbreitenvariable, abstimmbare Transponder. Weiterhin sind Online-Ressourcenzuweisungsmechanismen erforderlich, um für jede eintreffende Verbindungsanforderung abhängig vom aktuellen Netzzustand entscheiden zu können, ob diese akzeptiert werden kann und welche Netzressourcen hierfür reserviert werden. Dies bedeutet, dass die Ressourcenzuteilung als Online-Optimierungsproblem behandelt werden muss. Die Entscheidungen sollen so getroffen werden, dass auf lange Sicht ein vorgegebenes Optimierungsziel erreicht wird. Die Ressourcenzuweisung bei dynamischen optischen Netzen lässt sich in die Teilfunktionen Routing- und Spektrumszuteilung (RSA), Verbindungsannahmekontrolle (CAC) und Dienstgütesteuerung (GoS Control) untergliedern.

In dieser Dissertation wird das Problem der Online-Ressourcenzuteilung in dynamischen optischen Netzen behandelt. Es wird die Theorie der Markov-Entscheidungsprozesse (MDP) angewendet, um die Ressourcenzuweisung als Online-Optimierungsproblem zu formulieren. Die MDP-basierte Formulierung hat zwei Vorteile. Zum einen lassen sich verschiedene Optimierungszielfunktionen realisieren (z.B. die Minimierung der Blockierungswahrscheinlichkeiten oder die Maximierung der wirtschaftlichen Erlöse). Zum anderen lässt sich die Dienstgüte von Gruppen von Verbindungen mit spezifischen Verkehrsparametern gezielt beeinflussen (und damit eine gewisse GoS-Steuerung realisieren). Um das Optimierungsproblem zu lösen, wird in der Dissertation ein schnelles, adaptives und zustandsabhängiges Verfahren vorgestellt, dass im realen Netzbetrieb rekursiv ausgeführt wird und die Teilfunktionen RSA und CAC umfasst. Damit ist das Netz in der Lage, für jede eintreffende Verbindungsanforderung eine optimale Ressourcenzuweisung zu bestimmen. Weiterhin wird in der Dissertation die Implementierung des Verfahrens unter Verwendung eines 3-Way-Handshake-Protokolls für den Verbindungsaufbau betrachtet und ein analytisches Modell vorgestellt, um die Verbindungsaufbauzeit abzuschätzen. Die Arbeit wird abgerundet durch eine Bewertung der Investitionskosten (CAPEX) von dynamischen optischen Netzen. Es werden die wichtigsten Kostenfaktoren und die Beziehung zwischen den Kosten und der Performanz des Netzes analysiert. Die Leistungsfähigkeit aller in der Arbeit vorgeschlagenen Verfahren sowie die Genauigkeit des analytischen Modells zur Bestimmung der Verbindungsaufbauzeit wird durch umfangreiche Simulationen nachgewiesen.

## Schlagwörter

Dynamische Optische Netze, Flex-grid Wavelength Division Multiplexing (WDM), Online Routing and Spectrum Allocation (RSA), Connection Admission Control (CAC), Ressourcenzuweisung, Dynamische Programmierung, Markov Entscheidungsprozesse, Capital Expenditures (CAPEX)

## Bibliographic Description

Ronald Romero Reyes  
Online Resource Allocation in Dynamic Optical Networks  
Dissertation  
162 Pages, 61 Figures, 31 Tables, 118 References

### Abstract

Conventional optical transport networks have leveraged the provisioning of high-speed connectivity in the form of long-term installed, constant bit-rate connections. The setup times of such connections are in the order of weeks, given that in most cases manual installation is required. Once installed, connections remain active for months or years. The advent of grid computing and cloud-based services brings new connectivity requirements which cannot be met by the present-day optical transport network. This has raised awareness on the need for a changeover to dynamic optical networks that enable the provisioning of bandwidth on demand (BoD) in the optical domain. These networks will have to serve connections with different bit-rate requirements, with random interarrival times and durations, and with stringent setup latencies. Ongoing research has shown that grid computing and cloud-based services may in some cases request connections with holding times ranging from seconds to hours, and with setup latencies that must be in the order of milliseconds.

To provide BoD, dynamic optical networks must perform connection setup, maintenance and teardown without manual labour. For that, software-configurable networks are needed that are deployed with enough capacity to automatically establish connections. Recently, network architectures have been proposed for that purpose that embrace flex-grid wavelength division multiplexing, reconfigurable optical add/drop multiplexers, and bandwidth variable and tunable transponders as the main technology drivers. To exploit the benefits of these technologies, online resource allocation methods are necessary to ensure that during network operation the installed capacity is efficiently assigned to connections. As connections may arrive and depart randomly, the traffic matrix is unknown, and hence, each connection request submitted to the network has to be processed independently. This implies that resource allocation must be tackled as an online optimization problem which for each connection request, depending on the network state, decides whether the request is admitted or rejected. If admitted, a further decision is made on which resources are assigned to the connection. The decisions are so calculated that, in the long-run, a desired performance objective is optimized. To achieve its goal, resource allocation implements control functions for routing and spectrum allocation (RSA), connection admission control (CAC), and grade of service (GoS) control.

In this dissertation we tackle the problem of online resource allocation in dynamic optical networks. For that, the theory of Markov decision processes (MDP) is applied to formulate resource allocation as an online optimization problem. An MDP-based formulation has two relevant advantages. First, the problem can be solved to optimize an arbitrarily defined performance objective (e.g. minimization of blocking probability or maximization of economic revenue). Secondly, it can provide GoS control for groups of connections with different statistical properties. To solve the optimization problem, a fast, adaptive and state-dependent online algorithm is proposed to calculate a resource allocation policy. The calculation is performed recursively during network operation, and uses algorithms for RSA and CAC. The resulting policy is a course of action that instructs the network how to process each connection request. Furthermore, an implementation of the method is proposed that uses a 3-way handshake protocol for connection setup, and an analytical performance evaluation model is derived to estimate the connection setup latency. Our study is complemented by an evaluation of the capital expenditures of dynamic optical networks. The main cost drivers are identified.

The performance of the methods proposed in this thesis, including the accuracy of the analytical evaluation of the connection setup latency, were evaluated by simulations. The contributions from the thesis provide a novel approach that meets the requirements envisioned for resource allocation in dynamic optical networks.

### Keywords

Dynamic Optical Networks, Flex-grid Wavelength Division Multiplexing (WDM), Online Routing and Spectrum Assignment (RSA), Connection Admission Control (CAC), Resource Allocation, Dynamic Programming, Markov Decision Processes (MDP), Capital Expenditures (CAPEX)



# Acknowledgements

This dissertation was completed at the Chair for Communication Networks at the Technische Universität Chemnitz, Germany. Part of this work was supported by the German Federal Ministry of Education (BMBF), project VINO. I thank my doctoral supervisor Prof. Dr.-Ing. Thomas Bauschert not only for giving me the opportunity to participate in this project, but for his valuable guidance, support and constant dedication. His ideas and suggestions were key to the development of the models proposed in this thesis.

I would also like to express my gratitude to my students Vishwanath Pai and Omair Ali for their support and dedication in the outstanding implementation work they performed. Also I thank my colleagues at the Chair for Communication Networks. They provided me with a pleasant working environment and were always open to exchange their knowledge and to provide constructive ideas. Last but not least I thank and dedicate this work to my parents Manuel and Elizabeth, and my brothers Alexander and Rosemberg. They are the most important persons in my life, without their love and emotional support I would have never been able to accomplish this work.





# Contents

<b>Chapter 1 Introduction and Motivation .....</b>	<b>1</b>
1.1. Problem Statement .....	2
1.2. Research Contributions .....	3
1.3. Publications .....	4
1.4. Thesis Outline .....	4
<b>Chapter 2 Dynamic Optical Networking .....</b>	<b>6</b>
2.1. Dynamic Optical Network Architecture .....	6
2.2. Technology Drivers .....	8
2.3. Network Planning Challenges .....	10
2.4. Modelling Dynamic Optical Networks as Large-Scale Stochastic Systems .....	14
2.5. Chapter Summary .....	15
<b>Chapter 3 Reward-based Resource Allocation in Dynamic Optical Networks .....</b>	<b>17</b>
3.1. Stochastic Network Model .....	17
3.2. Policy-based Online Resource Allocation .....	21
3.3. Network Reward Maximization Problem .....	24
3.4. Network Reward Maximization as a Markov Decision Process .....	33
3.5. Chapter Summary .....	39
<b>Chapter 4 Approximate Approach to Online Resource Allocation .....</b>	<b>41</b>
4.1. Approximate Reward Maximization Problem .....	41
4.2. Macrostate-Dependent Network Reward Model .....	47
4.3. Online Estimation of Link Arrival Rates .....	55
4.4. Approximate Policy Iteration Algorithm .....	57
4.5. Chapter Summary .....	64
<b>Chapter 5 Performance Evaluation Results .....</b>	<b>66</b>
5.1. General Remarks .....	66
5.2. Performance Comparison of the Exact and the Approximate Reward Models .....	68
5.3. Performance Analysis of the Approximate PIA .....	77
5.4. Chapter Summary .....	85

<b>Chapter 6 Connection Establishment and Online Resource Allocation .....</b>	<b>87</b>
6.1. Implementation of the Approximate PIA .....	87
6.2. Performance Evaluation of Communication Protocols by Stochastic Task Graphs .....	90
6.3. Connection Setup Latency in Dynamic Optical Networks .....	93
6.4. Evaluation of the Analytical Performance Model for Connection Setup Latency .....	97
6.5. Chapter Summary .....	106
<b>Chapter 7 Modelling Infrastructure Costs in Dynamic Optical Networks.....</b>	<b>108</b>
7.1. Bottom-Up Calculation of the Total Cost of the Network Ownership.....	108
7.2. Calculation of Infrastructure Costs in Dynamic Optical Networks .....	110
7.3. Evaluation of Network Infrastructure Costs .....	115
7.4. Chapter Summary .....	121
<b>Chapter 8 Conclusion.....</b>	<b>123</b>
<b>List of Acronyms.....</b>	<b>127</b>
<b>List of Figures .....</b>	<b>129</b>
<b>List of Tables.....</b>	<b>131</b>
<b>Bibliography.....</b>	<b>132</b>
<b>Annex.....</b>	<b>141</b>



# Chapter 1

## Introduction and Motivation

Current optical transport networks are circuit-switched systems that carry customer demands on optical connections. For that, fixed-grid Wavelength Division Multiplexing (WDM) is implemented in the optical layer. A connection is a lightpath that consists of an optical channel established on a route between two network nodes. Nowadays, setup times for optical connections are in the order of weeks as in most cases manual labour is required for the provisioning of resources. Once established, connections remain active for months or years. Most operators today deploy fixed-grid WDM networks that multiplex up to 80 channels per fibre at a 50 GHz spacing. Each channel may carry a single connection with typical fixed bit-rates of 10, 40 and 100 Gbps. This inflexible and static design of the optical layer was conceived for the provisioning of constant bit-rate connections without stringent setup requirements. A typical use case of this type of connectivity is the transport of Internet protocol (IP) traffic. Operators overprovision capacity in the optical layer (by leasing/deploying high-speed, constant-bit rate connections) to cope with the burstiness of IP traffic. According to [Rob05, SS11], in 2005 it was observed that the lightpaths used to serve IP traffic had around 25 percent average utilization in the US Internet. This trend continues today, which evinces that optical capacity is not being used efficiently, as it is underutilized at the expense of high network deployment costs.

The constant IP traffic growth together with the emergence of grid computing and cloud-based services are imposing new connectivity requirements on the optical layer. This is raising awareness on the need for a changeover to dynamic optical networks. As argued in [Sal06, Sal07, Sim14], besides capacity benefits, dynamic optical networking enables the provisioning of bandwidth on demand (BoD) in the optical layer. Hence, instead of supplying long-term installed, constant bit-rate connectivity, dynamic optical networks are expected to serve connections with different bit-rate requirements, and with random interarrival and holding times. Besides, connections may need to be set up and torn down rapidly. (According to [Sal06, Sal07], to cope with the connectivity needs of grid computing and cloud-based services, the optical network will have to serve connections with interarrival and holding times ranging from seconds to hours.) The scientific community is currently working on the definition of a network architecture that meets these connectivity requirements. In order to provide fast BoD connectivity, this architecture must be software-controlled so that connections are automatically configured without manual intervention. Ongoing efforts in this direction have embraced flex-grid WDM [ITU12], reconfigurable optical add/drop multiplexers (ROADMs) [PFRC11, PP12], and bandwidth variable & tunable transponders [SS12, TAS<sup>+</sup>14] as the key technology drivers. In particular, the Core Optical Networks program (CORONET) [Sal06] - one of the most visible initiatives that promoted the definition of a dynamic optical network architecture - has adopted these drivers to develop two contributions: a directionless/colorless ROADM architecture [WFJA10, SCG<sup>+</sup>12] that interworks with a shared pool of transponders [SW10, CCC<sup>+</sup>12, SCG<sup>+</sup>12], and a 3-way handshake (3WHS) protocol that enables setup times in the range of milliseconds to seconds [CCC<sup>+</sup>12, SCG<sup>+</sup>12, SN09, SGK<sup>+</sup>14]. The idea behind CORONET's proposal is that the network must be installed with enough capacity to cope with an expected demand growth without manual labour. For that, the nodes are designed with ROADMs equipped with pools of shared transponders. Hence, instead of installing a dedicated pair of transponders for each point to point connection (like conventional networks do), the dimensioning, i.e. the estimation of the number of transponders, and the installation of pools is performed before the network gets operational. Then during operation, the customer signals are assigned - on demand - to transponders in the pool, thereby avoiding manual configurations. This implies that a connection request may get blocked if transponders are unavailable in the shared pools of the source/destination ROADMs. To minimize this effect, in [SW10, SCG<sup>+</sup>12] a pool dimensioning method was proposed that obtains low transponder-related blocking probabilities (e.g. less than  $10^{-4}$ ). By having sufficient installed capacity in the nodes, the 3WHS protocol was then proposed as the most efficient strategy to perform fast connection setup. For a connection request, the 3WHS protocol - unlike the generalized multiprotocol

label switching (GMPLS) [Ber03, Man04] mechanisms for connection setup - simultaneously probes resources on different paths, and therefore, it provides the network with real time information for more efficient resource allocation. Another advantage, as seen in [SCG<sup>+</sup>12], is that the protocol performs well at minimizing backward blocking (i.e. the blocking arising from concurrent connection requests trying to seize the same resources on a given path). These capabilities render the 3WHS protocol more efficient than GMPLS. A thorough discussion of the advantages of the 3WHS protocol over GMPLS can be found in [CCC<sup>+</sup>12, SCG<sup>+</sup>12].

The aforementioned innovations represent a first step towards the implementation of dynamic optical networks. However, further work is still needed for the definition of a control plane architecture, along with appropriate signalling mechanisms, that enable automatic multilayer/inter-domain interoperability. This is a prerequisite to automatically deliver BoD to higher technology layers such as IP. In this respect, ongoing research suggests that software defined networking (SDN) [ONF13, ONF14] can be the control plane solution for that purpose. For instance, in [Das12] an SDN-based framework was proposed for the convergence of packet and optical circuit-switched networks. This work forms the basis to start with the definition of SDN-based control mechanisms for dynamic optical networks. Such mechanisms include (to name a few): signalling procedures whereby higher technology layers (like IP) request BoD from the optical layer, signalling procedures for connection setup (for instance, those procedures should define how to realize protocols like the 3WHS in an SDN framework), and the definition of a flow control mechanism that supports the spectrum granularity of flex-grid WDM. (Note that a typical fixed-grid system supports spectrum grids with 50 GHz channels, whereas flex-grid defines a finer granularity with 6.25 GHz channel slots [ITU12].) Besides these open research challenges, another problem to tackle is the development of strategies for optimum resource allocation. In dynamic optical networks, connections may arrive and depart randomly. Therefore, the network has to implement online control functions to efficiently assign the installed capacity on demand. This means that every connection request submitted to the network has to be processed independently. This problem is the main focus of this thesis and will be described in more detail in the following section.

## 1.1. Problem Statement

Offline network planning and online resource allocation play a relevant role in the design of efficient network infrastructures. In dynamic optical networks, offline planning relies on demand forecasting to perform the dimensioning of links, ROADMs and transponders (i.e. it guarantees that sufficient infrastructure is deployed to provide BoD without manual intervention). On the other hand, online resource allocation ensures that the installed capacity is efficiently assigned to the connections served in the network. This is achieved by implementing network control functions for routing and spectrum allocation (RSA), connection admission control (CAC) and grade of service (GoS) control. These functions aim at providing quality of service (QoS) guarantees while the resources are optimally exploited.

Assuming an operational network with ROADMs equipped with pools of transponders, blocking of a connection request may occur either if transponders are unavailable in the shared pools of the source and destination ROADMs, or if a suitable lightpath cannot be found in the network. Low transponder-related blocking probabilities, as mentioned in the previous section, are achieved by a proper dimensioning of the transponder pools. On the other hand, the control of the blocking probability (i.e. the GoS) due to the unavailability of lightpaths is a task of online resource allocation. For a committed GoS, upon arrival of a connection request, resource allocation uses an online RSA algorithm to calculate candidate lightpaths. Then, based on admission decision rules, CAC determines the lightpath appropriate for the connection. Besides performing GoS control, resource allocation needs to fulfil at least three additional requirements to efficiently supply BoD. First, RSA and CAC algorithms must be fast, adaptive and state-dependent. Fast as resource allocation is performed on the time scale of connection interarrival times. Adaptive to varying traffic conditions, i.e. the criteria used to calculate lightpaths and to define the admission decision rules must dynamically change according to the statistical properties of the traffic. Also, admission control must take decisions that depend on the state, i.e. actual resource occupancy, of the network. Secondly, resource allocation algorithms must be designed to interwork with connection setup protocols. In reality, connection setup involves resource allocation. Thus, the design and implementation of network control functions for RSA and CAC needs to be aware of the signalling mechanisms whereby connections are established in the network. And thirdly, instead of solely minimizing blocking probabilities, the algorithms should allow optimization of other desired performance objectives, e.g. economic revenue maximization or cost minimization.

The design of online resource allocation algorithms that satisfy these requisites is an open problem. Today research on this field has focused on the design of online RSA algorithms. Existing approaches, e.g. [WWL10, CTV11b, PJJW11a, WWH<sup>+</sup>11, WZK<sup>+</sup>11, CVR<sup>+</sup>12, WZZ<sup>+</sup>12, ADB<sup>+</sup>13, CTV13], solely aim at minimizing the overall blocking probability, and do not allow GoS control of individual connection classes. (We adopt the term class to refer to a group of connections which have similar properties such as

the bandwidth requirements and holding times.) In reality a dynamic optical network must be able to serve different connection classes, each one with a committed GoS. Thus, online approaches that solely minimize overall blocking need to be extended to a multi-rate environment, and have to further satisfy the requisites envisioned for resource allocation in dynamic optical networks.

## 1.2. Research Contributions

In this thesis we tackle the problem of online resource allocation in dynamic optical networks. For that, we use the theory of Markov decision processes (MDP) [How60] to formulate resource allocation as a reward-based optimization problem. The idea behind the concept of reward is very simple: connections are categorized into classes, for each class, a reward parameter (whose meaning and actual value is defined by the network operator) is assigned. The reward parameter quantifies the benefit that a connection brings to the network if it is admitted. The goal is to find the resource allocation policy that maximizes the rate at which reward is earned from carried connections. (The term policy refers to the course of action used by the network to assign resources to connections depending on the network state.) The advantage of this approach is that the reward parameters can be set either to optimize any desired objective (e.g. blocking minimization, maximization of carried traffic or economic revenue) or to equalize or prioritize the GoS offered to different connection classes. MDP theory also provides the mathematical method to calculate the optimum policy. To apply this theory to the design of control algorithms for a specific type of network, it is mandatory to define first a stochastic model for the network. This includes a proper definition of the network state and the constraints of the optimization problem. With the model defined, the time evolution of the network state is represented as a stochastic process. Then MDP theory is used to control this process according to the desired optimization objective. This methodology was successfully applied to the design of adaptive and state-dependent routing in telephone and multiservice networks [DPKW88, DM89, DM92, DM94, Kri91, HKT00, Hwa93, Dzi97, Nor02]. However, those algorithms are not applicable to optical systems. The reason is that they describe stochastic networks with states and constraints that do not model the actual properties of the optical layer. In this thesis we therefore extend the applicability of MDP theory to the design of resource allocation methods for dynamic optical networks. The main contributions made to achieve this goal are:

1. **Formulation of an exact reward-based resource allocation algorithm:** a method is proposed whereby dynamic flex-grid optical networks are modelled as large-scale stochastic loss systems. The network state is defined as the configuration of the optical spectrum over all network links. This configuration is determined by the connections carried in the network, and hence, the state is given by the spectrum configurations of lightpaths that fulfil the contiguity and continuity constraints. (In flex-grid systems, a lightpath consists of contiguous spectrum slots centered at the same frequency on each link used by the lightpath.) Given this definition of state, the time evolution of the network is modelled as a continuous-time stochastic process. By applying MDP theory to control this process, resource allocation is formulated as a reward-based optimization problem. The solution is calculated by an online iteration algorithm that determines the optimum resource allocation policy. This policy is a set of state-dependent decisions on RSA and CAC that tell the network how to process each connection request. The approach guarantees that if the policy is used to allocate resources, the network reward rate is maximized. Since the policy calculation is performed online, its decisions are adaptive to changing traffic conditions. Besides, the approach optimizes any desired objective (and if needed performs GoS control) by properly setting the reward parameters. The proposed algorithm extends the applicability of MDP theory to the control of stochastic systems subject to contiguity and continuity constraints.
2. **Formulation of an approximate reward-based resource allocation algorithm:** a drawback of state-dependent control is that the computational complexity grows with the size of the network state-space. That is also the case of the exact reward-based algorithm, which is unsolvable for large-size networks. To circumvent this problem, a link decomposition approach is used whereby the rate of reward from the network is approximated as the sum of the link reward rates. By this strategy, the optimization problem is decomposed into separate problems (one for each link) by assuming statistical independence of the link state distributions. This approach reduces the computational effort required to calculate the resource allocation policy. In order to improve the adaptability to varying traffic conditions, the approach implements a method to estimate from online measurements the offered traffic. This allows the network to adapt the state-dependent policy decisions for each connection request.
3. **Definition of an implementation scenario for the approximate reward-based algorithm with the 3WHS protocol:** online resource allocation is part of the connection setup procedure. In that

respect, the 3WHS protocol for connection setup is adapted to perform resource allocation with the approximate reward-based algorithm. In the proposed scenario, the node that receives the connection request triggers the connection setup process, whereas the destination node calculates the policy decisions on CAC and RSA, and starts the resource allocation procedure. A centralized control plane is responsible for coordinating the whole process. The signalling mechanism of the 3WHS protocol guarantees setup latencies in the range of milliseconds to seconds. The proposed approach can be realized as an extension to GMPLS or as an SDN-based implementation.

4. **Analytical approach to evaluating connection setup latency in dynamic optical networks:** an analytical method to estimate the connection setup latency of the 3WHS protocol is proposed. For that, the protocol is modelled as a task graph which represents the signalling latency during the connection establishment phase. By using reduction techniques, the graph is simplified so as to obtain a performance model that estimates the mean connection setup time. The analytical model provides precise estimates of the connection setup latencies. The advantage of the task graph approach is that it is applicable to any connection setup protocol operating with arbitrary resource allocation algorithms. Thus, the provided analysis can easily be extended to estimate latencies in different implementation scenarios for online resource allocation.
5. **Evaluation of infrastructure costs for dynamic optical networking:** a comprehensive cost model to evaluate the capital expenditures (CAPEX) for dynamic optical networks is provided. Based on a given traffic demand, the cost model uses a bottom-up approach to dimensioning the directionless/colorless ROADMs and the shared pools of transponders in the network. Then the infrastructure costs are evaluated. The approach provides a framework to identify the main cost drivers. The model can further be applied to calculate the network operational expenses (OPEX) and the total cost of ownership (TCO). The work extends the scope of existing cost evaluation studies for flex-grid WDM networks, e.g. [LYL<sup>+</sup>12].

The performance of all methods proposed in this thesis, including the accuracy of the analytical evaluation of the connection setup latency, were evaluated by simulations.

## 1.3. Publications

The contributions to this thesis from the author are published in two journal papers [RB17a, RB17b] and nine conference papers [RB15, REB15, RB16a, RB16b, RB16c, RB16d, RB17c, RAB18, RB18a]. Another publication from the author related to the topic is [GREB17]. A synopsis of the thesis that summarizes relevant findings is published in [RB18b].

## 1.4. Thesis Outline

The thesis is organized as follows. In Chapter 2 the architecture for dynamic optical networks is explained. Emphasis is given on the main technology drivers, the network planning challenges and the importance of modelling the network as a large-scale stochastic system. This chapter provides the main definitions and assumptions used in the remaining chapters of the thesis. Chapter 3 is focused on the formulation of the exact reward-based algorithm for resource allocation. For that, a stochastic loss model is defined for dynamic optical networks. Then the concept of policy is introduced, and MDP theory is applied to derive the reward-based optimization problem. An online policy iteration procedure is formulated to calculate the optimum resource allocation policy. In Chapter 4 a link independence assumption is applied to the derivation of an approximate reward-based algorithm. A simplified iteration procedure is proposed to calculate the resource allocation policy. The procedure uses an online estimation method that adapts the policy decisions to the stochastic properties of the traffic. Chapter 5 presents performance evaluation results obtained by simulations. First a performance comparison is made between the exact and the approximate reward-based algorithms. The purpose is to assess the accuracy of the approximate method. Besides, for different network topologies, a thorough performance analysis is made on the approximate algorithm. Chapter 6 describes a control plane implementation for the approximate approach with the 3WHS protocol. Furthermore, an analytical performance evaluation model is proposed to estimate the protocol latency. Simulations are used to assess the accuracy of the model. In Chapter 7 a bottom-up cost calculation framework is proposed to calculate the CAPEX in dynamic optical networks. In particular, the cost contributions from ROADMs and transponders to the network CAPEX is studied. Finally, Chapter 8 concludes the thesis and outlines future research challenges.





# Chapter 2

## Dynamic Optical Networking

This chapter describes an architecture envisioned for the design and implementation of dynamic optical networks. The architecture defines the network model used in the next chapters of this thesis. Section 2.1 outlines the transport services and technology layers that are currently driving the emergence of dynamic optical networking. In particular, a multilayer/multiservice transport network architecture is defined for the provisioning of bandwidth on demand over the photonic layer. In Section 2.2 we introduce the key technology enablers to dynamic optical networking, namely, the flexible spectrum grid (denoted as flex-grid), the bandwidth variable and tunable transponders, and the flexible optical switches. Section 2.3 discusses network design challenges that involve network control functions for online resource allocation. In Section 2.4 we argue on the importance of designing those control functions by modelling the network as a large-scale stochastic system. By this approach, it is possible to implement control mechanisms that guarantee optimum utilization of the network resources.

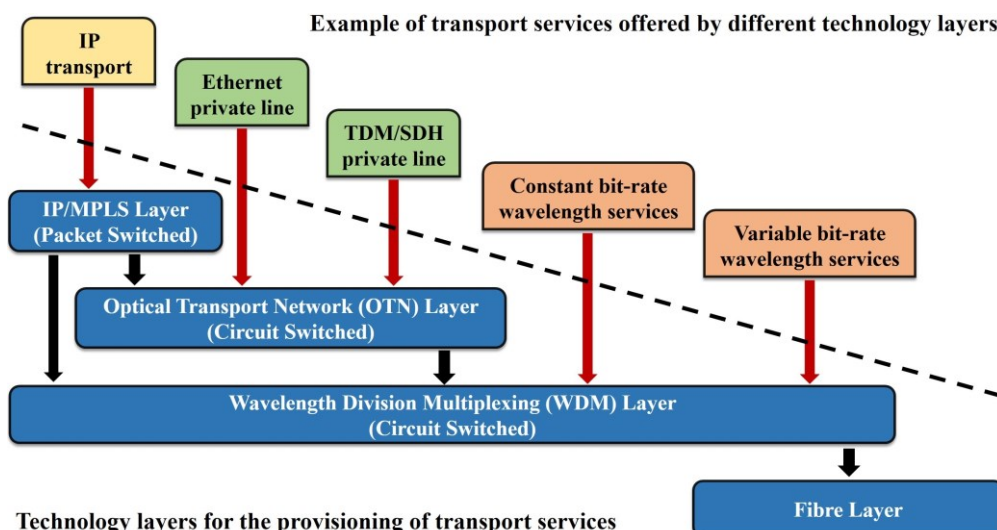
### 2.1. Dynamic Optical Network Architecture

Communication service providers (CSP) supply information, transport and passive infrastructure services. Information services provide mechanisms to generate, store, process and retrieve information [CW03]. Transport offers transparent transmission of information without modifying its content. On the other hand, passive infrastructure services provide the physical support for the provisioning of transport. To offer those services, CSPs deploy multiservice networks that integrate different technology layers [REB15, RB17b]. In particular, the photonic (or optical) layer plays a relevant role as it leverages the provisioning of high-speed connectivity over access, metro and core networks. In the following we focus on the core or transport network and outline an architecture that has been envisioned for a dynamic optical layer.

#### 2.1.1 Transport Services and Technology Layers

Core networks may provide a variety of transport services by integrating packet and circuit switched networking technologies. Figure 2.1 gives an overview of such integration which derives in a multilayer and multiservice core network. At the bottom is the fibre layer which provides the physical support for the provisioning of transport. Upon this fibre-based layout, we have the WDM layer which provides high-speed (constant bit-rate and variable bit-rate) wavelength services. On top of this layer, the optical transport network (OTN) technology can be provisioned to supply sub-wavelength services such as Ethernet and time division multiplexing (TDM) private lines. (Synchronous digital hierarchy (SDH) and synchronous optical network (SONET) services can be provisioned over OTN as well.) At the top we find the Internet protocol/multiprotocol label switching (IP/MPLS) layer which provides the transport commonly offered by internet service providers. Unlike IP/MPLS and Ethernet, which are implemented by packet-switched technologies, WDM, OTN, TDM and SDH/SONET services are realized by circuit-switched networking systems.

The service classification in Fig. 2.1 points out the importance of WDM as the key technology enabler for the provisioning of high-speed transport at all upper layers. Optical transport networks are the circuit-switched systems that implement the WDM layer. These networks carry client signals, e.g. wavelength services, IP/MPLS and OTN traffic streams, on optical connections provisioned as lightpaths, i.e. optical channels established on a route between two nodes. Today the WDM layer is static in the sense that the setup times for optical connections are in the order of weeks, and once established, connections remain



**Figure 2.1:** Transport services and technology layers that define the core network.

active for months or years. With the emergence of cloud-based services and grid computing, it is expected an evolution towards a dynamic WDM layer capable of providing high-speed bandwidth on demand [BSBS08, JTK<sup>+</sup>09, MCD<sup>+</sup>11, AWY13, Sim14, CDF<sup>+</sup>15]. This implies a changeover to highly-dynamic networks where connections can be rapidly set up and torn down. In this future scenario, optical connections are expected to have random arrival times and different bandwidth requirements.

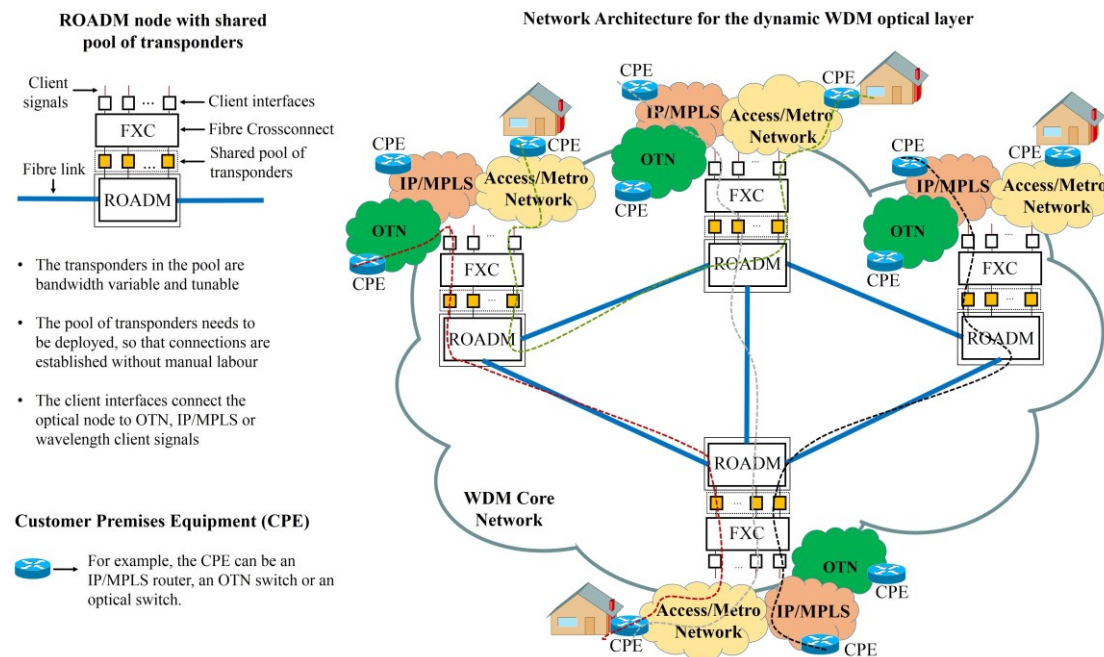
## 2.1.2 Network Architecture for the Dynamic Optical WDM Layer

In order to cope with the stringent connectivity requirements from applications such as those based on cloud services and grid computing, the WDM layer must be implemented as a dynamic optical network. We adopt the term “dynamic” to refer to an optical transport network which has the following attributes:

- The network is able to provide BoD connectivity to the customers, i.e. optical connections can be established when and where needed. This has two relevant implications. First, the network must be installed with the infrastructure required to establish connections without involving manual intervention [CCC<sup>+</sup>12, SCG<sup>+</sup>12, Sim14]. Secondly, connections (if needed) can be rapidly set up and torn down. In [Sal06, Sal07] it is shown that connections might require setup times in the order of milliseconds to seconds (e.g. for inter-data centre networks).
- The network performs resource allocation for connection requests which may arrive randomly and which may have different bandwidth requirements. Furthermore, these requests may need to be served either upon arrival or at some specific time - as defined by the customer - after the arrival (this is the case of scheduled traffic, where a request is submitted in advance for the reservation of resources [Sim14]). The connection holding times can be either random (i.e. the connections can arbitrarily be terminated by the customers) or deterministic. This implies that the network has to implement resource allocation algorithms to efficiently serve the traffic demand.

To address these requirements, in Fig. 2.2 we present an architecture for dynamic optical networks which is based on the concepts developed in [Sal06, Sal07, SN09, SW10, WFJA10, MCD<sup>+</sup>11, SS11, CCC<sup>+</sup>12, SCG<sup>+</sup>12, CDF<sup>+</sup>15]. The architecture is realized by three key technology drivers, namely, flex-grid WDM [TAS<sup>+</sup>14], bandwidth variable and tunable transponders and flexible ROADMs.

Besides enabling an efficient utilization of the optical spectrum, flex-grid WDM makes possible the provisioning of connections with different bandwidth requirements. To implement this technology in a dynamic environment, the optical network is installed with nodes that integrate three components: a fibre crossconnect (FXC), a directionless and colorless flexible ROADM, and a shared pool of transponders [SW10, WFJA10, CCC<sup>+</sup>12, SCG<sup>+</sup>12]. As seen in Fig. 2.2 the fibre crossconnect interconnects (on demand) the client interfaces to transponders in the pool. The transponders must be bandwidth variable and tunable, so that they can dynamically be configured to provision the bandwidth requested by the connections. The pool of transponders connects to a ROADM which is responsible for adding/dropping and switching (in the optical domain) connections.



**Figure 2.2:** Dynamic optical network architecture.

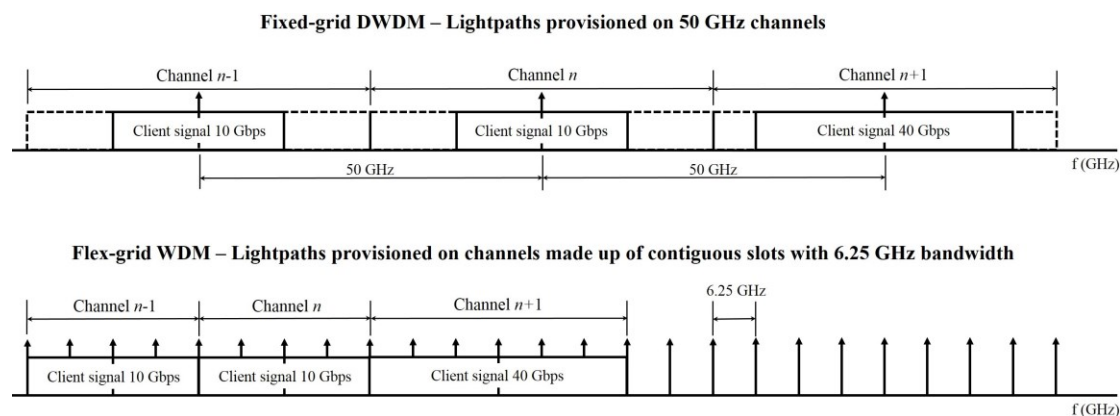
The customers of the dynamic optical network are higher technology layers such as OTN and IP/MPLS or clients that request high-speed (either constant or variable bit-rate) wavelength services. If the customers are far away from the nearest point of presence (POP) of the network provider, their customer premises equipment (CPE) may reach the POP via a dedicated channel over an access/metro network. Otherwise, if the CPE is collocated at the POP, a short-range interconnection, e.g. grey or coloured [RKD<sup>+</sup>13], can be used to reach the FXC of the optical node (see Fig. 2.2).

## 2.2. Technology Drivers

Conventional optical transport networks use fixed-grid WDM [ITU12] as the technology that multiplexes different connections onto a single optical fibre. With this technology, the optical spectrum is split into channels spaced at a fixed distance (expressed in GHz). For instance, most operators today deploy fixed-grid systems that multiplex up to 80 channels per fibre at a 50 GHz spacing. This WDM variant is known as dense WDM (DWDM). Each channel may carry connections with typical (fixed) bit-rates of 10, 40 and 100 Gbps. The major drawback of this technology is the inefficient usage of the spectrum resources. The emergence of advance multiplexing techniques, such as orthogonal frequency division multiplexing (OFDM) [ZDMM13] and Nyquist-WDM [BCC<sup>+</sup>11], have enabled the evolution of WDM towards flex-grid WDM [ITU12]. This technology is the main driving force behind the dynamic network architecture depicted in Fig. 2.2.

### 2.2.1 Flex-grid Wavelength Division Multiplexing

The International Telecommunication Union – Telecommunication standardization sector (ITU-T) has defined in the recommendation G.694.1 [ITU12] the frequency plan for flex-grid WDM. Compared to fixed-grid, the flexible standard defines a spectrum grid that consists of frequency slots with a 6.25 GHz bandwidth. This finer granularity allows the provisioning of optical channels with different bandwidth requirements, which is accomplished by allocating the number of contiguous slots that guarantee the bit-rates required by the connections. This number depends on the spectral efficiency of the modulation scheme used by the transponders. Figure 2.3 presents a comparison of the traditional fixed-grid DWDM system with flex-grid. As seen, the major advantage of a flexible grid is the capability of tailoring the optical bandwidth to the customer requirements, thereby saving spectrum resources. Fixed and flex-grid are defined in the C-band transmission window which spans the wavelength range 1530 nm-1565 nm.



**Figure 2.3:** Comparison between fixed-grid and flex-grid WDM.

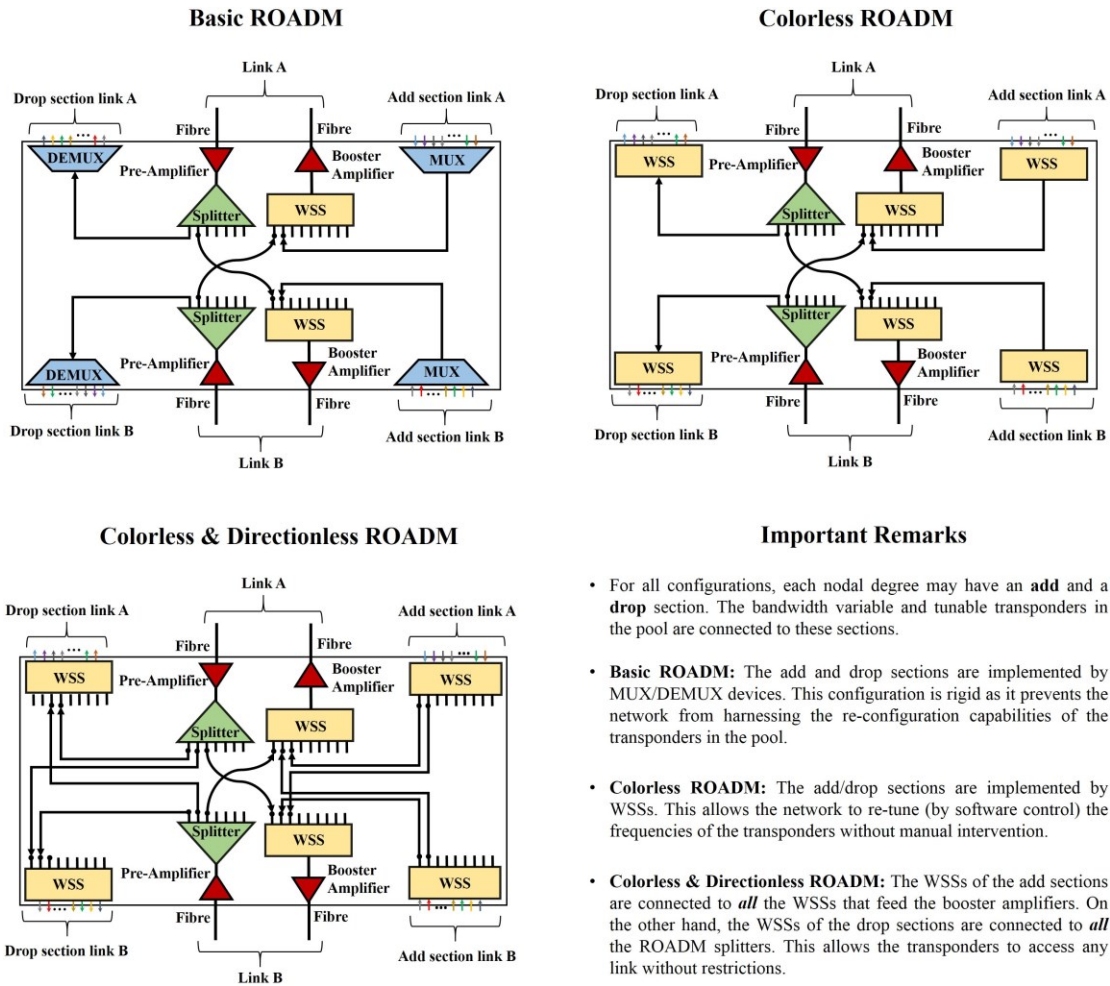
## 2.2.2 Flexible Node Architecture

Let us take a closer look at the flexible node architecture depicted in Fig. 2.2. The transponders installed in the shared pool are bandwidth variable. This is accomplished by multiplexing sub-carrier frequencies (using techniques such as OFDM or Nyquist-WDM) into a single channel that provides the bit-rate of the client signal [JTK<sup>+</sup>08, KRM<sup>+</sup>10, GJLY12]. (The sub-carrier frequencies are defined in the flexible spectrum grid.) Besides providing bit-rate adaptability, the transponders are frequency-tunable, thereby the sub-carrier frequencies that conform a client lightpath can be centred at any standard frequency within the spectrum grid. Another attribute of the transponders is the transparent reach, which is the maximum achievable transmission distance before requiring signal regeneration. This distance is shortened as the signal bit-rate increases (and is dependent on the modulation scheme applied to the channel sub-carriers). For instance, typical values published in [RKD<sup>+</sup>13] show that flexible transponders can provide bit rates from 10 Gbps (with reach 2000 km) to 400 Gbps (with reach 150 km).

An important design issue is the dimensioning of the number of transponders to install in the pool of each network node. Notice that blocking of an optical connection request may occur either if transponders are unavailable in the shared pools of the source and destination ROADMs, or if a suitable lightpath cannot be found in the network (due to the unavailability of free spectrum slots). In [SW10, CCC<sup>+</sup>12, SCG<sup>+</sup>12] a dimensioning method has been proposed that obtains low transponder-related blocking probabilities (e.g. less than  $10^{-4}$ ). In the method, a traffic forecast needs to be defined for the optical network. Then it is assumed that each node has an infinite number of transponders. Based on this assumption, simulations are run to determine the histograms of the transponder occupancies at each node. From the histograms, the number of transponders that guarantee a desired blocking is obtained for each pool. Although the method uses simulations, it has the advantage that is applicable to any arbitrarily defined traffic forecast, and thus, it is not limited by the traffic assumptions (e.g. Poisson arrivals) typically made by traditional queuing models used for capacity dimensioning.

Flexible ROADMs [PFR11] switch connections with different bandwidth requirements in the optical domain. Besides, they are the elements that add and drop connections in the network. In Fig. 2.4 we present three typical ROADM configurations, namely, basic, colorless, colorless & directionless, which are built by combining arrayed waveguide gratings (AWGs) - used to implement optical multiplexers (MUX) and demultiplexers (DEMUX) - passive splitters, wavelength selective switches (WSS) and optical amplifiers (pre-amplifiers and booster amplifiers for incoming and outgoing WDM signals, respectively) [GBS<sup>+</sup>10]. As seen in Fig. 2.4, all ROADM configurations (which are depicted for a two-degree node) have in common that an add/drop section is installed per node degree. The add/drop sections are connected to the transponders installed in the node. Before explaining the differences among the configurations, let us recall that an optical splitter is a passive device that replicates an incoming WDM signal on all its outgoing ports. On the other hand, the WSS takes an incoming WDM signal and individually switches each of its constituent output port [GBS<sup>+</sup>10].

In the basic ROADM configuration (see Fig. 2.4), each transponder attached to an add/drop section (via the passive MUX/DEMUX) can only access a single ROADM link. Besides this limitation, this node configuration does not automatically react (i.e. without manual intervention) if a transponder re-tunes its carrier frequency. (This is understood by the fact that the MUX/DEMUX ports are coloured, i.e. they operate at a fixed frequency.) This limitation is overcome by the colorless ROADM configuration which replaces the MUX/DEMUX devices by WSSs. These switches have colorless ports, and therefore, they allow the transponders to automatically re-tune their carrier frequencies without restrictions. However, as with the basic ROADM, this solution (as seen in Fig. 2.4) only allows the transponders to add/drop traffic from a fixed ROADM link. This restriction is avoided by implementing the add sections with WSSs that



**Figure 2.4:** Typical ROADM configurations (a two-degree node is depicted as an example).

can connect to all the WSSs that feed the booster amplifiers. On the other hand, the drop sections are implemented by WSSs that can connect to all the ROADM splitters. This solution is the colorless & directionless ROADM seen in Fig. 2.4. It allows any transponder to access any link connected to the node. This configuration enhances the network resilience (as traffic can automatically be re-routed in case of failures). In Chapter 7 a thorough analysis of these three configurations will be presented.

The colorless & directionless configuration together with the shared pools of transponders render the dynamic optical network in Fig. 2.2 implementable. However, this approach poses challenges for the design and operation of this type of networks. These issues will be discussed in the following section with special focus on the resource allocation problem and on selected control plane design challenges.

### 2.3. Network Planning Challenges

To supply their services at low costs in the telecommunications market, CSPs perform techno-economic analyses to decide on a suitable network deployment plan. These analyses consist of the formulation, modelling and evaluation of business cases for selected candidate network solutions. A business case is a study that projects into the future the financial consequences (i.e. costs, revenues, profits, risk, etc.) of a decision. It tells (by means of financial metrics) what is expected to happen if a candidate network solution is adopted [REB15]. It is the task of the decision makers to select and install the network solution that fits best to their business interests. After the initial deployment, the evolution of the network life-cycle is determined by techno-economic evaluations of future network upgrades/migrations. The goal is to achieve optimal technical performance while yielding financial feasibility.

Techno-economic analyses involve network planning and economic evaluation methods. To illustrate this within the context of the network life-cycle, assume that at time  $t_i$  a CSP wants to install a network to cope with an expected service demand over an interval  $T_i$ . To select a suitable network deployment plan, different candidate network solutions are defined. The financial feasibility of each solution is evaluated

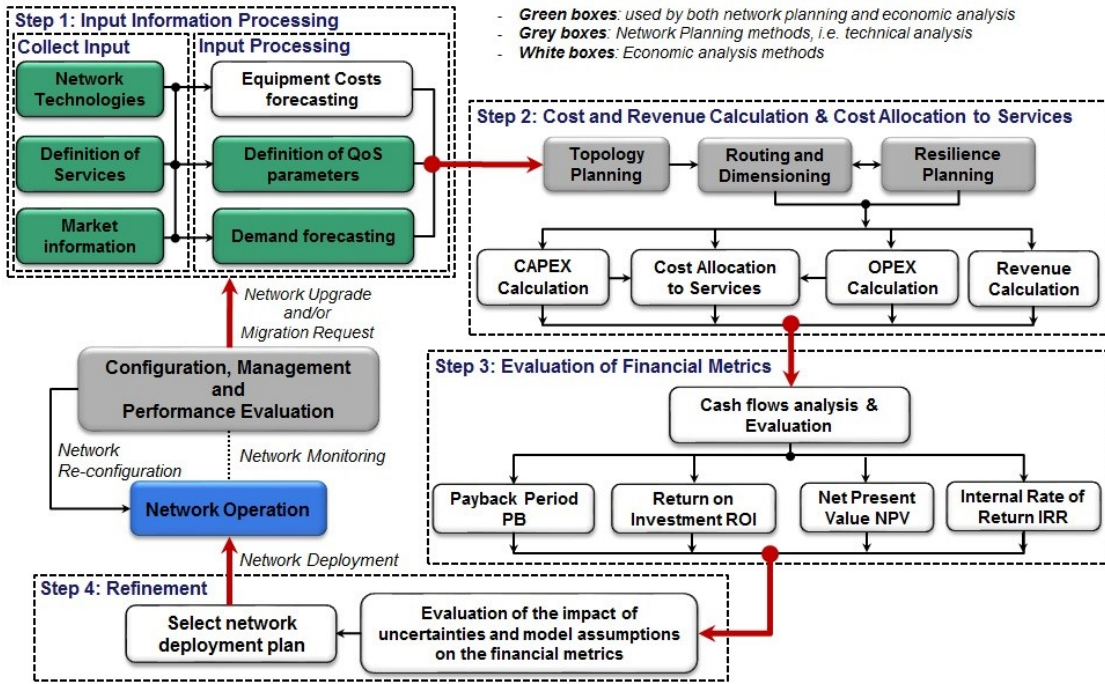


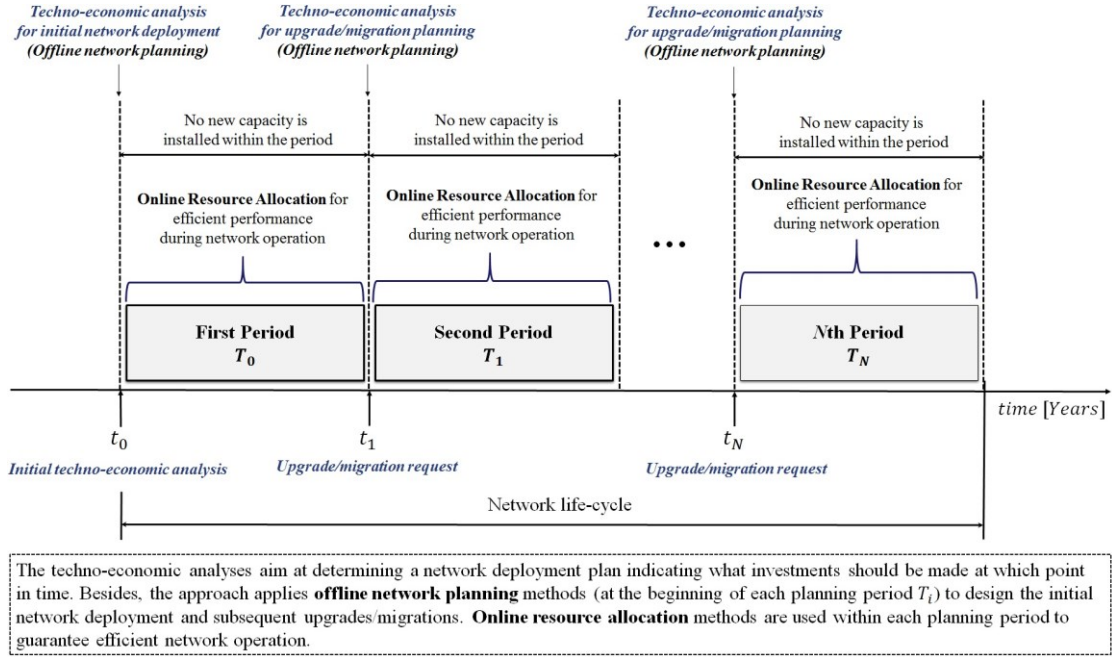
Figure 2.5: Techno-economic analysis for network deployment planning [RB17b].

with the four-step model depicted in Fig. 2.5 [VCOL08, RB17b]. The grey and white boxes represent methods to perform technical and economic analysis, respectively. The green boxes are methods common to both analyses.

At time  $t_i$  the four steps in Fig. 2.5 are executed for each candidate network solution. The first step, namely, input information processing, is responsible for surveying and defining data regarding market information, competitors, the network architecture and technologies to use, equipment pricing, the services to be offered and their QoS requirements. With this information, service demands and equipment costs are forecast over  $T_i$ . In the second step, the input information is used to design the network on the condition that the TCO is minimized while no capacity upgrades are needed in  $T_i$ . For that, offline network planning optimization algorithms are used for topology design, demand routing, infrastructure dimensioning and resilience planning. With the resulting design, the TCO is calculated as the sum the CAPEX and OPEX. Besides, the costs are allocated to the services offered by the CSP [You85, CVM<sup>+</sup>06, REB15, RB17b] and the expected revenues are calculated. Hence, the second step defines a network deployment plan and its cash flow stream that states the timing and value of costs (cash outflows) and revenues (cash inflows) over  $T_i$ . In the third step, financial metrics such as the payback period (PB), the return on investment (ROI), the net present value (NPV) and the internal rate of return (IRR) are evaluated [VCOL08]. They provide information about the time it takes for revenues to cover costs, expected investment gains, risks, etc. The reliability of that information is assessed in the last step, namely, refinement. The purpose is to evaluate the impact of the model assumptions and the uncertainties of the input information on the financial metrics. Real options valuation [TVW<sup>+</sup>13] can be used as a method to identify and quantify the risks associated to uncertainties and unexpected events.

Once the candidate network solutions are assessed, the most suitable alternative is chosen and installed at  $t_i$ . In order to guarantee optimal operation within the planning period  $T_i$ , configuration, management and performance evaluation functions are performed on the installed network (see in Fig. 2.5 the blocks that connect the fourth to the first step). These functions include online resource allocation algorithms whereby capacity is assigned to the service demands admitted over  $T_i$ . Furthermore, these functions should recommend the time  $t'_i$  at which a network upgrade/migration is necessary. At this time, the four-steps of the techno-economic analysis are repeated so as to select a suitable solution for the upgrade/migration request. This process is sequentially executed over the network life-cycle (see Fig. 2.6, where the evolution of a typical network throughout different planning periods  $T_i$  is shown - the periods need not have the same length).

As seen in Fig. 2.6, offline network planning and online resource allocation play a relevant role in the design of efficient network infrastructures. Based on demand forecasting, offline planning guarantees that enough capacity is installed to cope with the expected demand growth. On the other hand, online resource allocation ensures that the installed capacity is efficiently used by the demands actually served in the network. In particular, for dynamic optical networks, offline planning includes the dimensioning of the colorless & directionless ROADMs and the sizing of shared pools of transponders (i.e. it guarantees that



**Figure 2.6:** The network life-cycle as driven by techno-economic evaluations.

sufficient infrastructure is deployed to provide BoD without manual intervention). Online resource allocation, on the other hand, is responsible for the implementation of network control functions for CAC and RSA. These functions guarantee that lightpaths are allocated to connections in such a way that a desired performance is achieved.

### 2.3.1 Online Resource Allocation

Conventional fixed-grid WDM networks serve connections with setup times in the order of weeks (as in most cases manual labour is required) and once established, connections remain active for months or years. Given the static characteristics of this traffic, offline planning for capacity provisioning is performed by routing and wavelength assignment (RWA) algorithms. With the advent of flex-grid WDM, more efficient capacity dimensioning is attained by offline RSA schemes - see for example [CTV11a, KW11, KRV<sup>+</sup>13]. Both RWA and RSA offline algorithms have in common that the traffic matrix is known. Besides, these algorithms can be formulated as integer linear programming (ILP) or mixed ILP (MILP) problems. RWA and RSA differ in that RSA algorithms must calculate lightpaths that fulfil the contiguity and continuity constraints, i.e. the lightpath spectrum slots must be adjacent and centered at the same frequency on each link in the selected path. (In RWA algorithms there is no contiguity constraint as a single wavelength may suffice for a connection request.)

In dynamic optical networks, offline RSA algorithms are only useful (based on demand forecasting) to estimate the capacity to deploy for any given planning period. However, for network operation (i.e. to perform online resource allocation) those algorithms are inapplicable as they tend to be slow. To provide GoS guarantees to their customers, the network needs to rely on resource allocation methods that integrate algorithms for online RSA and CAC. To carry a new connection, online RSA - based on an optimization objective - is responsible for calculating a set of candidate lightpaths. Then CAC applies decision rules to determine the lightpath on which the connection has to be carried. Although parts of the same problem, RSA and CAC can be solved separately to reduce modelling and implementation complexity [Dzi97]. In this context, for flex-grid WDM, a variety of online RSA schemes are proposed in the literature [WWL10, CTV11b, PJJW11a, WWH<sup>+</sup>11, WZK<sup>+</sup>11, CVR<sup>+</sup>12, WZZ<sup>+</sup>12, ADB<sup>+</sup>13, CTV13]. They mainly aim at minimizing the overall blocking probability. A drawback of these algorithms is that they do not allow individual control of the blocking probability (i.e. GoS) of different connection classes (i.e. connections that have different bandwidth requirements). However, these approaches represent a first step towards the design of efficient resource allocation algorithms in dynamic optical networks. In general, any online resource allocation algorithm should fulfil the following requirements:

1. Since online resource allocation is performed on the time scale of connection interarrival times, the algorithms must be fast. Existing online RSA schemes tackle this issue by adopting heuristic methods to calculate lightpaths. For instance, some approaches [WWH<sup>+</sup>11, WZK<sup>+</sup>11, WZZ<sup>+</sup>12] resort to decomposition techniques that solve the routing and the spectrum allocation problems



separately. By adopting this decomposition, resource allocation can be realized as follows. For a connection request, a routing algorithm is run to calculate candidate paths. (Routing can be solved by considering the physical limitations imposed by the transparent reach of the transponders and the modulation formats.) For each candidate path, a spectrum allocation algorithm, such as first-fit (FF) or random-fit (RF) [TAK<sup>+</sup>14], calculates an optical channel that meets the contiguity and continuity constraints. The resulting lightpaths are evaluated by a CAC algorithm so as to decide whether to admit or reject the connection. If admitted, a lightpath is chosen and provisioned for the connection.

2. The algorithms must perform individual control of the blocking probability (i.e. GoS) of different connection classes. This is particularly important if BoD services are to be offered. This allows CSPs to provide service level agreements (SLA) to customers with distinct bandwidth needs.
3. Although the blocking probability is the major performance objective, the algorithms should allow the optimization of objectives arbitrarily defined, e.g. economic revenue maximization or cost minimization.
4. The algorithms must be state-dependent and adaptive to changing traffic load conditions. This means that the CAC decisions have to take into account the actual resource occupancy (i.e. the network state), and need to adapt to traffic fluctuations so as to meet the SLAs at its best.

These requirements pose new challenges to the design of online resource allocation algorithms. Their performance is subject to the deployment of sufficient capacity at the beginning of each planning period within the network life-cycle (i.e. offline network planning should guarantee that the ROADMs, transponder pools and the fibre links are properly sized to cope with the traffic growth).

Spectrum defragmentation is another challenge to online network planning. The random arrival and departure of optical connections renders the spectral resources prone to reach a so-called fragmentation state [ASMW11]. A fragmentation state means that, on each link, the spectrum contains idle and non-contiguous spectrum slots which cannot be allocated to future connection requests. The increment in the connection blocking probability is a result of a fragmented spectrum. A strategy to solve this problem is to re-optimize the configuration of the lightpaths established in the network, i.e. their spectrum allocation and/or routing. By that, a spectrum defragmentation in the network is performed, whereby large blocks of empty and adjacent spectrum slots are made available for future connection requests. Online spectrum defragmentation involves three phases [GREB17]. First, on each link, the network tracks the spectrum fragmentation state so as to trigger the defragmentation process when deemed necessary. In the second phase, an optimization algorithm recalculates the spectrum allocation (or both the routing and spectrum allocation) for all carried lightpaths. In the last step, a strategy is adopted to migrate the lightpaths to the configuration calculated by the optimization algorithm. Such a strategy has to minimize the number of lightpath disruptions during the migration process. There is an increasing research interest in spectrum defragmentation strategies for flex-grid WDM networks. For example, [SP10, PJJW11b, THS<sup>+</sup>11] focus on spectrum defragmentation without migration of lightpaths. In [CVR<sup>+</sup>12] an online RSA scheme is introduced which re-allocates spectrum to established lightpaths when a connection request is blocked. A drawback of this scheme is that it may cause frequent reconfiguration processes (e.g. in high-traffic load scenarios). ILP-based defragmentation algorithms with integrated lightpath migration are introduced in [THT<sup>+</sup>14, TTHK15, TTHK16], where the defragmentation and the lightpath migration strategies are calculated simultaneously. These algorithms can be used to improve the resource allocation process in dynamic optical networks. For example, RSA schemes can be designed together with CAC decision rules to minimize the spectrum fragmentation. It is also important to investigate the extent to which a migration of lightpaths is recommended in a dynamic scenario, as migrations may lead to service disruptions that may impact the SLAs offered to the customers.

### 2.3.2 Control Plane Challenges

To unfold the potential of dynamic optical networks, a control plane architecture is needed that enables automatic multilayer/inter-domain interoperability. Moreover, given the circuit-switched nature of the optical network, novel signalling mechanisms are required for fast connection set up.

Multilayer interoperability requires the definition of logical interfaces between higher technology layers (e.g. IP/MPLS, OTN) and the flex-grid WDM dynamic layer. These interfaces should specify the signalling procedures whereby BoD is automatically provisioned to the customers. Although efforts have been made to extend the GMPLS protocol suite to support flex-grid WDM (see for example [JTK<sup>+</sup>09, GCZ<sup>+</sup>13]), ongoing research shows that an SDN-based control plane would reduce complexity, thereby adding flexibility to the control of dynamic traffic flows [DPM12]. In particular, the work in [Das12] has

embraced SDN to propose an integration of packet (IP/MPLS) and optical circuit-switched networking technologies. As a result, SDN concepts have been extended to the optical domain [Col13, ONF13, SHS<sup>+</sup>14]. This has stimulated further research on the integration of GMPLS protocol functionalities with SDN/OpenFlow capabilities [ANE<sup>+</sup>11, CKE<sup>+</sup>12]. The experimental results reported in [CNF<sup>+</sup>12, LCT<sup>+</sup>12, YZZ<sup>+</sup>12, ZZY<sup>+</sup>13, ZZZ<sup>+</sup>13] indicate that by centralizing the network control with an SDN architecture, not only multilayer convergence, but inter-domain interoperability is attained. However, additional effort is still required for the definition of an SDN-based dynamic optical network. In particular, this effort entails the standardization of the signalling procedures for the provisioning of capacity for multirate connections. (Note that currently, as defined by the fixed-grid WDM standard [ITU12], the GMPLS protocol handles connections at the granularity of wavelengths. The extension to flex-grid WDM implies a finer spectrum granularity, which imposes stringent scalability requirements not yet handled by GMPLS [TAS<sup>+</sup>14].) Furthermore, as argued in [Sal06, Sal07] dynamic optical networks might need to serve connections with setup time requirements in the order of milliseconds to seconds. This evinces the need for faster connection establishment protocols that interwork with resource allocation algorithms.

In [CCC<sup>+</sup>12, SCG<sup>+</sup>12, SN09, SGK<sup>+</sup>14] a 3-way handshake (3WHS) protocol was proposed to provide fast connection setup in dynamic optical networks. (A modified version of this protocol for multicarrier domain networks has been published in [WGK<sup>+</sup>15].) Originally, the 3WHS protocol was designed as an alternative (or extension) to GMPLS in order to cope with the stringent setup latencies of BoD services. Unlike GMPLS, upon arrival of a connection request, the 3WHS protocol is able to simultaneously probe resources on different paths, and therefore, it provides the network nodes with the information they need to perform more efficient resource allocation. Another advantage, as shown by the simulation results in [SCG<sup>+</sup>12], is that the protocol performs well at minimizing backward blocking (i.e. the blocking arising from concurrent connection requests trying to seize the same resources on a given path). These capabilities make the 3WHS protocol faster than GMPLS. An open issue is the definition of the implementation details of the protocol in an SDN-based environment. In this respect, a major challenge is the minimization of the delays of the signalling messages exchanged between the nodes and the SDN controller. As discussed in [Sim14], a purely centralized control plane implementation may result in large setup latencies. Therefore, a question to answer is whether a hybrid (i.e. centralized and distributed) or a purely centralized control plane is more suitable for fast connection setup.

## 2.4. Modelling Dynamic Optical Networks as Large-Scale Stochastic Systems

Extrinsic and intrinsic factors determine the performance (e.g. the blocking probability, the connection setup latency) of dynamic optical networks. Among the extrinsic factors we have the statistical properties of the bandwidth requirements, the interarrival times and the durations of connections. These properties, in principle, are independent of - and unknown to - the network. On the other hand, intrinsic factors include online resource allocation algorithms whereby lightpaths are provisioned on demand. The network performance is the large-scale effect that emerges from the interaction between the extrinsic and intrinsic factors. Understanding the causal relationship responsible for the emergence of the large-scale effects is essential for the design of efficient network control. By having this knowledge, it is possible to handle (and learn from) the randomness of connections so that the network control attains a desired performance. Given the random nature of the traffic, dynamic optical networks behave as large-scale stochastic systems whose state (i.e. the network resource occupation) changes by following the decisions of the resource allocation algorithm.

Having a suitable stochastic network model is key to the implementation of optimum control. A variety of models were successfully proposed in the past for packet and circuit switched networks (see for example [KY14]). In particular, circuit switched networks are modelled as stochastic loss systems. Relevant models for this type of networks such as [Kau81, Kel88, DM89, Hun89, Kel91, Hwa93, DM94, Ros95, HKT00, KY14] have succeeded in the design of adaptive/state-dependent routing and CAC. Unfortunately, these models are inapplicable to dynamic optical networks. The reason is that they describe stochastic systems with states which are not subject to continuity and contiguity constraints. A first step towards the definition of a stochastic model for dynamic optical networks is the work in [YZZ<sup>+</sup>14]. Therein a model is proposed for a flex-grid WDM link. The optical link is described as a continuous-time Markov chain with states that fulfil the spectrum contiguity constraint. From the Markovian model, the link performance metrics (e.g. the blocking probability, the spectrum fragmentation) can be derived analytically. The simulation results in [YZZ<sup>+</sup>14] show that, for different spectrum allocation schemes (e.g. first-fit, random-fit), the analytical model provides accurate estimates of the link performance.

In order to design applicable network control algorithms, the stochastic link model in [YZZ<sup>+</sup>14] must be extended first to multi-link networks. This implies the generalization of the concept of state, which has

to consider the spectrum continuity constraint. Having defined the network model, existing methods for the control of loss networks can be applied to design online RSA and CAC algorithms. For example, the strategies developed in [Kel88, DM89, Hun89, Kel91, Kri91, Ros95, ZRP00, CNDR01, Mei13] model resource allocation as an online optimization problem and propose solutions methods that are adaptive and state-dependent. In particular, the network control models in [Kel91, DM89, Ros95] suggest that, if an accurate stochastic network model is defined, efficient resource allocation algorithms can be designed and implemented for dynamic optical networks.

## 2.5. Chapter Summary

Dynamic optical networks are envisioned as multiservice infrastructures capable of providing bandwidth when and where needed. These networks can be built as circuit-switched systems that serve connections with different bandwidth requirements, random interarrival times and durations. The key technology drivers to dynamic optical networking are flex-grid WDM, bandwidth variable and tunable transponders and flexible ROADMs. With these technologies, the network design is performed by techno-economic analyses that involve offline network planning methods and economic evaluations. Based on demand forecasting, offline planning ensures that - before operation - the network nodes and links are installed with enough capacity to cope with the demand growth. For that, the optical nodes are designed with directionless & colorless ROADMs which are interconnected (via fibre crossconnects) with pools of transponders. By this node design, client interfaces are assigned (on demand) to transponders in the pool, thereby avoiding manual labour during the process of connection setup. During network operation, online resource allocation ensures that the installed capacity is efficiently used by the demands served in the network. For that, the network has to implement control functions for RSA and CAC. These functions guarantee that lightpaths are allocated to connections in such a way that a desired performance is achieved.

The statistical characteristics of the traffic and the control logic of the resource allocation algorithms determine the performance of dynamic optical networks (e.g. the blocking probability). This performance is a large-scale effect that can be controlled by modelling the network as a stochastic loss system. In the next chapters we will elaborate on this idea with special focus on adaptive and state-dependent resource allocation strategies. For that, the concepts and models introduced in the remaining chapters make the following assumptions. First, that the network conforms to the architecture described in this chapter. Secondly, that the dimensioning of ROADMs, transponders pools and flex-grid links has been performed by offline network planning methods. Thus, the assumption is made that sufficient capacity is available to cope with an expected demand growth. Our focus will then be on how to efficiently allocate resources to connections during network operation.



# Chapter 3

## Reward-based Resource Allocation in Dynamic Optical Networks

Online resource allocation in dynamic optical networks involves network control functions for CAC and RSA. These functions aim at guaranteeing that lightpaths are allocated to connections in such a way that a desirable network performance is achieved. In this chapter we define the performance as the rate at which the network earns reward from carried connections. Within that context, resource allocation is modelled as a Markov decision process that provides a policy with decisions on admission, routing and spectrum allocation. Section 3.1 defines a stochastic model for dynamic optical networks. In Section 3.2 the network model is used to introduce the concept of policy as a mechanism for resource allocation. Section 3.3 formulates the policy dependent reward-based optimization problem. Section 3.4 discusses a policy iteration algorithm that calculates an optimum policy for network reward maximization. To facilitate the readability of the concepts and mathematical models introduced in this chapter, in Table 3.1 we summarize the most relevant variables and parameters which are defined and used throughout all chapter sections. The content in this chapter includes results published by the author in [RB16a, RB16b].

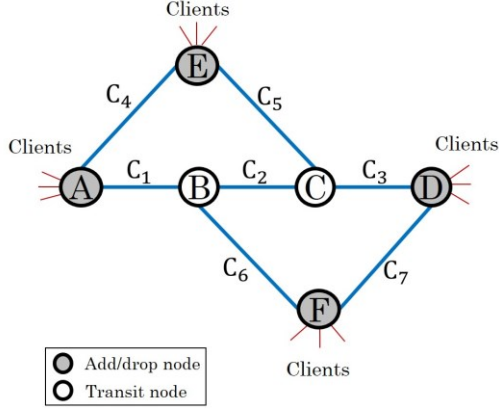
### 3.1. Stochastic Network Model

Consider a dynamic flex-grid optical network with  $N$  nodes and  $L$  links. Each link  $l$  has a capacity of  $C_l$  ordered spectrum slots and the nodes employ directionless/colorless ROADMs. Two types of nodes are distinguished: add/drop nodes and transit nodes. They differ only in that add/drop nodes have transponder pools to establish connections on demand. The connections served by the network are categorized into  $J$  distinct classes, with a class- $j$  being a group of connections fully characterized by six parameters, namely  $(o, d)_j$ ,  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$ . Specifically, class- $j$  connections request connectivity between a pair of add/drop nodes  $(o, d)_j$ , arrive at the network at a mean rate  $\lambda_j$ , and have a mean holding time  $\mu_j^{-1}$ . A carried class- $j$  connection seizes a bandwidth of  $b_j$  spectrum slots. This bandwidth is provisioned by a lightpath on a route  $\rho$  selected from a set of feasible paths  $\Gamma_j$ . (In general,  $\Gamma_j$  may contain either all feasible paths between the nodes  $(o, d)_j$ , or a sub-set of  $k$ -shortest paths.) The lightpath fulfils the contiguity and continuity constraints, i.e. in addition to being adjacent, the  $b_j$  spectrum slots are centered at the same frequency on each link in the selected path  $\rho$ . A class- $j$  connection brings to the network  $r_j$  reward units (ru). The meaning and value of the reward parameter  $r_j$  is defined by the network operator and thus,  $r_j$  does not necessarily represent an economic revenue. Class- $j$  connections offer to the pair of add/drop nodes  $(o, d)_j$  a traffic load of  $A_j = \lambda_j / \mu_j$  Erlangs. The rates  $\lambda_j$  and  $\mu_j$  have units of connections per unit of time (con/uot). Note that  $\mu_j$  is the rate at which a single carried class- $j$  connection is terminated.

**Example 3.1** Consider the network depicted in Fig. 3.1, which consists of seven links, four add/drop nodes (A, D, E, F) and two transit nodes (B, C). The add/drop nodes define six node-pairs, each one serving two connection classes, one requesting two and the other four spectrum slots. Thus, the network serves 12 classes, which are defined by the parameters shown in Fig. 3.1. For example, class-10 is the group of connections carried as lightpaths with a bandwidth of  $b_{10} = 4$  adjacent slots provisioned between the nodes  $(o, d)_{10} = (D, E)$ . They arrive at a mean rate of  $\lambda_{10} = 6$  (con/uot) and have a mean holding time  $\mu_{10}^{-1} = 1$ , thereby offering a traffic load of  $A_{10} = 6$  Erlangs. The lightpaths are provisioned on routes selected from the set of paths  $\Gamma_{10} = \{(D, C, E), (D, F, B, A, E)\}$ . From a carried class-10 connection the network earns a reward of  $r_{10} = 8$  (ru).

Symbol	Units	Description
$C_l$	slots	Capacity, in spectrum slots, of link $l$ with $l = 1, 2, \dots, L$
$J$	classes	Number of connection classes served by the network
$(o, d)_j$	-----	Pair of nodes between which class- $j$ connections are established
$\lambda_j$	con/uot	Exogenous mean rate at which class- $j$ connections arrive at the network
$\mu_j$	con/uot	Mean rate at which a single class- $j$ connection is terminated
$b_j$	slots	Number of contiguous slots requested by a class- $j$ connection
$\Gamma_j$	-----	Set of $k$ -shortest paths that connect the nodes $(o, d)_j$ . On these paths lightpaths with a bandwidth $b_j$ are provisioned to serve class- $j$ traffic
$r_j$	ru/con	Reward parameter of a class- $j$ connection
$A_j$	Erlangs	Mean class- $j$ traffic offered to the network
$Ac_j$	Erlangs	Mean class- $j$ traffic carried by the network
$J^l$	-----	$J^l = \{j: l \in \rho \wedge \rho \in \Gamma_j\}$ , classes which can be carried on link $l$
$\mathbf{x}^l$	-----	$\mathbf{x}^l = (x_1^l, x_2^l, \dots, x_{C_l}^l)$ is a $C_l$ -dimensional vector, and represents the state of link $l$ , where $x_i^l \geq 0$ is the state of the $i$ th link spectrum slot
$\mathbf{x}$	-----	$\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^l, \dots, \mathbf{x}^L)$ is the network state, $\mathbf{x}^l$ is the state of link $l$
$\Omega_{\mathbf{x}}^l, \Omega_{\mathbf{x}}$	-----	Link state-space and network state-space, respectively. These sets have cardinalities $ \Omega_{\mathbf{x}}^l $ and $ \Omega_{\mathbf{x}} $
$\mathbf{n}^l$	-----	$\mathbf{n}^l = (n_1^l, \dots, n_j^l, \dots, n_j^l)$ is the macrostate of link $l$ , with $n_j^l$ being the number of carried class- $j$ connections
$\Omega_{\mathbf{n}}^l$	-----	Link macrostate-space. This set has cardinality $ \Omega_{\mathbf{n}}^l $
$\Pi$	-----	Resource allocation policy. It is a matrix $[\Pi(\mathbf{x}, j)]$ , the element $\Pi(\mathbf{x}, j)$ is a decision that instructs the network the course of action to take when a class- $j$ connection request arrives in network state $\mathbf{x}$
$R(\Pi)$	ru/uot	Mean rate at which the network earns reward from carried connections
$B_j$	-----	Blocking probability of class- $j$ connections
$P_{\mathbf{xy}}(\Pi)$	-----	Probability that the network moves from state $\mathbf{x}$ to state $\mathbf{y}$ , $\mathbf{x} \neq \mathbf{y}$
$v(\mathbf{x}, \Pi)$	ru	Transient reward earned by having set the network in state $\mathbf{x}$ at $t_0$
$V(\mathbf{x}, \Pi, t)$	ru	Expected reward at $t \gg t_0$ , if the network were in state $\mathbf{x}$ at $t_0$ , this reward is earned at a constant rate $R(\Pi)$
$\Gamma_{\mathbf{x}}^{j+}, \Gamma_{\mathbf{x}}^{j-}$	-----	Sets of network states which are reachable due to class- $j$ connection admissions and departures in state $\mathbf{x}$ , respectively
$\Lambda_{\mathbf{x}}^{j+}$	-----	Set of decisions for a class- $j$ request arriving in state $\mathbf{x}$ . An element $\mathbf{y}$ in $\Lambda_{\mathbf{x}}^{j+}$ is a state in $\Omega_{\mathbf{x}}$ , and it represents the state reachable if the network makes the transition $\mathbf{x} \rightarrow \mathbf{y}$
$g_j(\mathbf{y}, \mathbf{x}, \Pi)$	ru	State-dependent network reward gain $g_j(\mathbf{y}, \mathbf{x}, \Pi) = v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$ . It is the reward change, w.r.t. $V(\mathbf{x}, \Pi, t)$ , incurred at $t \gg t_0$ if at $t_0$ the transition $\mathbf{x} \rightarrow \mathbf{y}$ is caused by admission of a class- $j$ connection arriving in state $\mathbf{x}$ . It is also interpreted as the long-term reward obtained from a class- $j$ connection accepted when the network is in state $\mathbf{x}$

**Table 3.1:** Notation of relevant variables and parameters defined in Chapter 3.



Definition of parameters for connection classes						
Class- $j$	$(o, d)_j$	$b_j$	$\lambda_j$	$\mu_j^{-1}$	$r_j$	$\Gamma_j$
1	(A,D)	2	4	1	4	{(A,E,C,D),(A,B,F,D)}
2	(A,E)	2	1	1	4	{(A,E),(A,B,C,E)}
3	(A,F)	2	3	1	4	{(A,B,F),(A,E,C,D,F)}
4	(D,E)	2	5	1	4	{(D,C,E),(D,F,B,A,E)}
5	(D,F)	2	2	1	4	{(D,F),(D,C,B,F)}
6	(F,E)	2	1	1	4	{(F,B,A,E),(F,D,C,E)}
7	(A,D)	4	2	1	8	{(A,E,C,D),(A,B,F,D)}
8	(A,E)	4	5	1	8	{(A,E),(A,B,C,E)}
9	(A,F)	4	1	1	8	{(A,B,F),(A,E,C,D,F)}
10	(D,E)	4	6	1	8	{(D,C,E),(D,F,B,A,E)}
11	(D,F)	4	2	1	8	{(D,F),(D,C,B,F)}
12	(F,E)	4	9	1	8	{(F,B,A,E),(F,D,C,E)}

Figure 3.1: Example of a flex-grid optical network serving 12 classes.

Slot State	Description
$x_i^l = 0$	The $i$ th slot is free
$x_i^l = j$	The $i$ th slot is occupied and is the first one in an ordered sequence of $b_j$ adjacent spectrum slots, which are all allocated to a single class- $j$ connection, with $j \in \mathcal{J}^l$
$x_i^l = \infty$	The slot is occupied and is not the first one in the group of adjacent spectrum slots allocated to the connection it belongs to

 Table 3.2: Definition of possible states for the  $i$ th spectrum slot on a network link.

### 3.1.1 Definition of the Network State

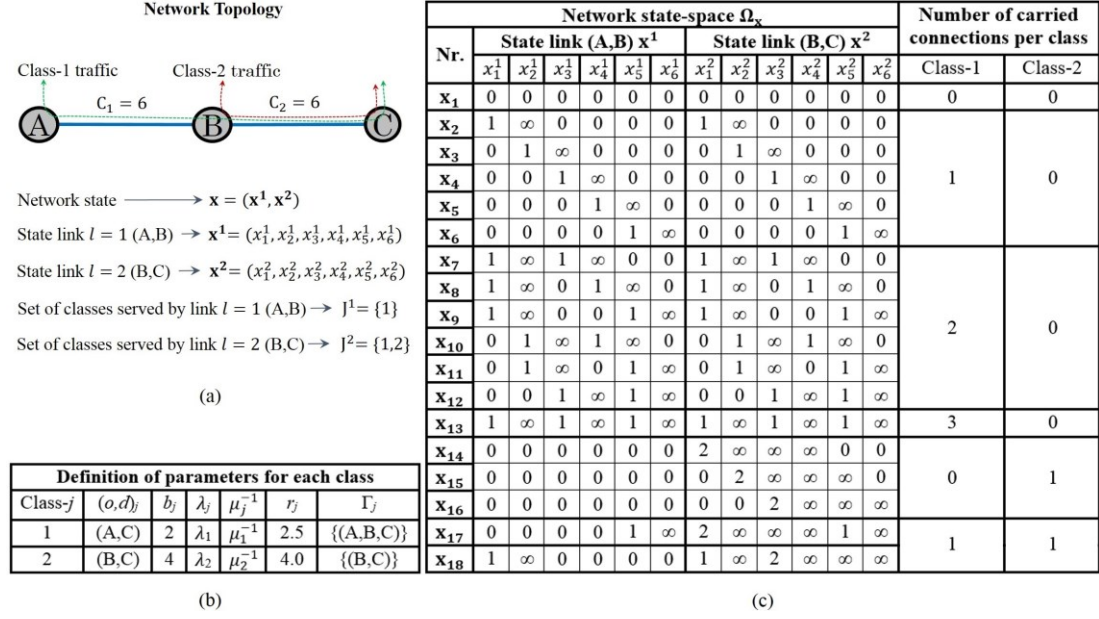
As in [YZZ<sup>+</sup>14], let  $\mathbf{x}^l = (x_1^l, x_2^l, \dots, x_{C_l}^l)$  be the state of link  $l$ , with  $x_i^l$  being the state of the  $i$ th spectrum slot ( $i = 1, 2, \dots, C_l$ ). Let us further define  $\mathcal{J}^l = \{j : l \in \rho \wedge \rho \in \Gamma_j\}$  as the set of class- $j$  connections which can be carried on link  $l$ . From this, in Table 3.2 we define the possible states  $x_i^l$  for the  $i$ th spectrum slot. Any definable  $\mathbf{x}^l$  is an allowed configuration of the optical spectrum on link  $l$  and therefore,  $\mathbf{x}^l$  has to fulfil the spectrum contiguity constraint which is defined as:

$$\text{If } x_i^l = j, j \in \mathcal{J}^l, \text{ then } i \leq C_l - b_j + 1 \text{ and } x_{i+k}^l = \infty, k = 1, 2, \dots, b_j - 1, \forall l \quad (3.1)$$

which means that when a class- $j$  connection is carried on a path  $\rho \in \Gamma_j$ , such that  $l \in \rho$  (i.e.  $j \in \mathcal{J}^l$ ), the connection seizes an optical channel with a bandwidth of  $b_j$  adjacent spectrum slots. If the first channel slot occupies the  $i$ th position in the optical grid, i.e.  $x_i^l = j$ , then the subsequent  $b_j - 1$  slots are allocated to the connection as well, i.e.  $x_{i+k}^l = \infty$  ( $k = 1, \dots, b_j - 1$ ). This implies that the  $i$ th position in the link grid is such that  $i \leq C_l - b_j + 1$ . The set of states  $\mathbf{x}^l$  that fulfil Equation (3.1) is the link state-space  $\Omega_{\mathbf{x}}^l$ .

**Example 3.2** Assume that for the network in Fig. 3.1,  $C_l = 6$  slots,  $l = 1, \dots, 7$ . The state of link  $l$  is  $\mathbf{x}^l = (x_1^l, x_2^l, x_3^l, x_4^l, x_5^l, x_6^l)$ . Consider the link  $l = 2$ , i.e. that interconnecting the nodes B-C. This link is in the path  $\rho = (A, B, C, E) \in \Gamma_2, \Gamma_8$  and in the path  $\rho = (D, C, B, F) \in \Gamma_5, \Gamma_{11}$  (see Fig. 3.1). Thus, the link may solely carry connections of classes  $\mathcal{J}^2 = \{2, 5, 8, 11\}$ . According to Table 3.2 and Equation (3.1), two possible states for this link are  $\mathbf{x}^2 = (0, 5, \infty, 0, 2, \infty)$  and  $\mathbf{x}^2 = (11, \infty, \infty, \infty, 5, \infty)$ . In both states the link carries two connections: in the former, a class-5 connection that seizes slots 2-3 and a class-2 connection which occupies slots 5-6 (i.e. each connection uses two slots, as  $b_2 = b_5 = 2$ ), whereas in the latter state, a class-11 connection seizes slots 1-4 (as  $b_{11} = 4$ ) and a class-5 connection uses slots 5-6. An example of a non-valid state for this link is  $\mathbf{x}^2 = (0, 3, \infty, 11, \infty, \infty)$ . First, because the link cannot carry class-3 connections (i.e.  $j = 3 \notin \mathcal{J}^2$ , and thus, no path in  $\Gamma_3$  contains link B-C). Secondly, because the spectrum configuration “11,  $\infty, \infty$ ” violates Equation (3.1). Any class-11 connection must be configured as “11,  $\infty, \infty, \infty$ ”, where the starting slot  $i$  for which  $x_i^2 = 11$  has to fulfil  $i \leq 3$  and  $x_{i+k}^2 = \infty$  for  $k = 1, 2, 3$ .

Let us further define  $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^l, \dots, \mathbf{x}^L)$  as the network state, with  $\mathbf{x}^l$  being the state of link  $l$  which fulfils Equation (3.1). The state  $\mathbf{x}$  is a valid configuration of the optical spectrum in the network, i.e.  $\mathbf{x}$  represents the allocation of the spectrum slots (over all network links) to carried connections. Any definable state  $\mathbf{x}$  has to fulfil the spectrum continuity constraint which is defined as:



**Figure 3.2:** Example of a flex-grid optical network that serves two connection classes.

$$\text{If } x_i^l = j, \text{ then } x_i^{l'} = j \text{ and } x_{i+k}^{l'} = \infty, k = 1, 2, \dots, b_j - 1, \forall l' \in \rho: l \in \rho, \rho \in \Gamma_j \quad (3.2)$$

which means that if a class- $j$  connection seizes  $b_j$  slots on a link  $l$  in a path  $\rho \in \Gamma_j$ , then on the remaining links  $l'$  in the path, the  $b_j$  slots are positioned as on link  $l$  (i.e. tuned at the same frequency). The set of states  $\mathbf{x}$  that fulfil the contiguity and continuity constraints, i.e. Equations (3.1) and (3.2) is the network state-space  $\Omega_{\mathbf{x}}$ .

**Example 3.3** Assume that the network in Fig. 3.1, with  $C_l = 6$  slots,  $\forall l$ , is in a state  $\mathbf{x}$  in which a class-9 connection is carried on the path  $\rho = (A, B, F) \in \Gamma_9$ . In this state, the two links in  $\rho$ , namely, link (A,B) denoted as  $l = 1$ , and link (B,F) denoted as  $l = 6$ , can be configured as  $\mathbf{x}^1 = (x_1^1, x_2^1, 9, \infty, \infty, \infty)$  and  $\mathbf{x}^6 = (x_6^1, x_2^6, 9, \infty, \infty, \infty)$ , thereby the connection fulfils Equations (3.1) and (3.2) as it seizes slots 3-6 in both links. The slots  $x_1^1, x_2^1$  and  $x_1^6, x_2^6$  can be in any state that does not violate the constraints.

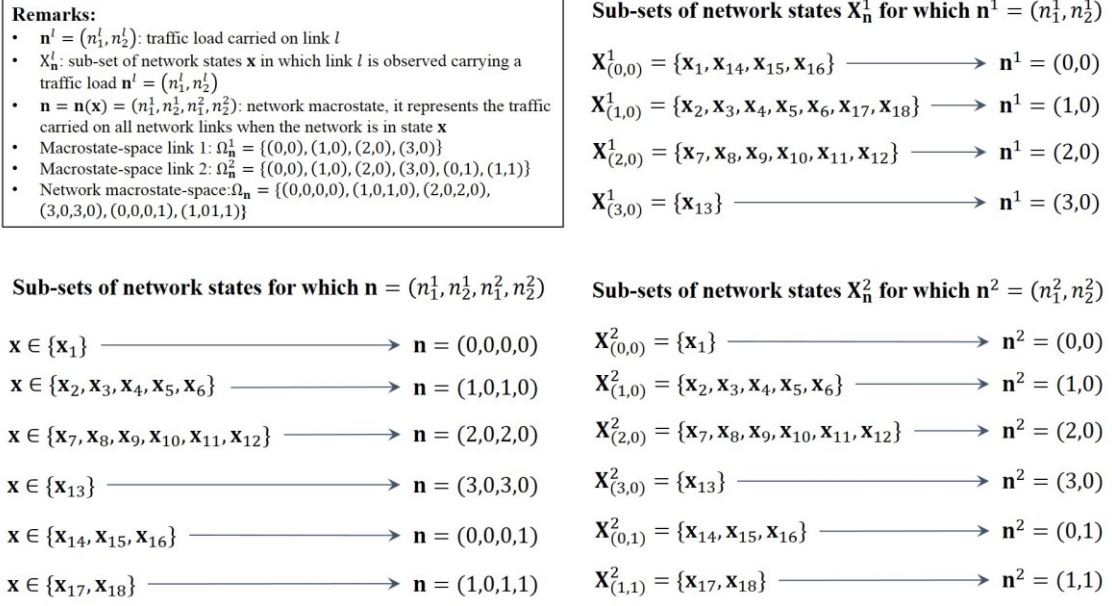
**Example 3.4** Consider the network in Fig. 3.2a, with three add/drop nodes (A,B,C) and two links: link (A,B) denoted as  $l = 1$ , and link (B,C) denoted as  $l = 2$ , with capacities  $C_1 = 6$  and  $C_2 = 6$  slots. The network serves two classes defined by the parameters in Fig. 3.2b. Class-1 is the group of connections between the nodes  $(o, d)_1 = (A, C)$ , with a bandwidth  $b_1 = 2$  provisioned on the path (A, B, C) in  $\Gamma_1$ . A class-1 connection yields a reward of  $r_1 = 2.5$  (ru). Class-2 connections seize  $b_2 = 4$  slots between the pair of nodes  $(o, d)_2 = (B, C)$ . These lightpaths are provisioned on the route (B, C), which is in the set  $\Gamma_2$ . A class-2 connection yields a reward of  $r_2 = 4$  (ru). Links 1 and 2 serve, respectively, classes  $J^1 = \{1\}$  and  $J^2 = \{1,2\}$ . The network state is given by  $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2)$ , with the link states defined as  $\mathbf{x}^1 = (x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1)$  and  $\mathbf{x}^2 = (x_1^2, x_2^2, x_3^2, x_4^2, x_5^2, x_6^2)$ . Figure 3.2c shows state-space  $\Omega_{\mathbf{x}}$  of the network as derived from Equations (3.1) and (3.2). The space  $\Omega_{\mathbf{x}}$  has 18 states, each one representing a feasible configuration of the optical spectrum in the network. (In Fig. 3.2c, each state  $\mathbf{x}$  is numbered as  $\mathbf{x}_i, i = 1, 2, \dots, 18$ .) For example, while in state  $\mathbf{x} = \mathbf{x}_{17} = (0, 0, 0, 0, 1, \infty, 2, \infty, \infty, \infty, 1, \infty)$ , which corresponds to the link states  $\mathbf{x}^1 = (0, 0, 0, 0, 1, \infty)$  and  $\mathbf{x}^2 = (2, \infty, \infty, \infty, 1, \infty)$ , the network carries two connections: one of class-1 that seizes slots 5-6 on links 1 and 2, i.e. on the path (A,B,C), and a class-2 connection which occupies slots 1-4 on link 2, i.e. on the path (B,C). In Fig 3.2c, every carried connection fulfils the spectrum contiguity and continuity constraints.

### 3.1.2 Definition of the Network Macrostate

Let  $\mathbf{n}^l = (n_1^l, n_2^l, \dots, n_j^l, \dots, n_J^l)$  be the link macrostate, which is the instantaneous traffic carried on link  $l$ , with  $n_j^l$  and  $J$  being the number of carried class- $j$  connections and the total number of classes served by the network, respectively. Any definable macrostate  $\mathbf{n}^l$  has to fulfil the capacity constraint:

$$\sum_{j \in J^l} b_j \cdot n_j^l \leq C_l, \forall l \quad (3.3)$$





**Figure 3.3:** Possible macrostates (carried traffics) for the network in Fig. 3.2.

where the sum is over the classes  $j$  which can be carried on link  $l$ . From this we have that  $n_j^l = 0, \forall j \notin J^l$ . The set of macrostates that meet the capacity constraint is the macrostate-space  $\Omega_n^l$ . Any macrostate  $\mathbf{n}^l$  is explicitly defined by a sub-set of network states  $\mathbf{x}$ . We denote such a sub-set as  $X_n^l$ , i.e. there can be more than one spectrum configuration  $\mathbf{x}$  that define the same macrostate (carried load)  $\mathbf{n}^l$ . Let us further define  $\mathbf{n} = \mathbf{n}(\mathbf{x}) = (\mathbf{n}^1, \mathbf{n}^2, \dots, \mathbf{n}^l, \dots, \mathbf{n}^L)$  as the network macrostate, which represents the number of carried connections per class (over all network links) when the network is in state  $\mathbf{x}$ . The set of all macrostates  $\mathbf{n}$  is the network macrostate-space  $\Omega_n$ .

**Example 3.5** For the network in Fig. 3.2, the network macrostate is  $\mathbf{n} = (\mathbf{n}^1, \mathbf{n}^2)$  with link macrostates  $\mathbf{n}^1 = (n_1^1, n_2^1)$  and  $\mathbf{n}^2 = (n_1^2, n_2^2)$ . The state-space  $\Omega_x$  in Fig. 3.2c defines the network macrostate-space  $\Omega_n = \{(0,0,0,0), (1,0,1,0), (2,0,2,0), (3,0,3,0), (0,0,0,1), (1,0,1,1)\}$  with link macrostate-spaces given by  $\Omega_n^1 = \{(0,0), (1,0), (2,0), (3,0)\}$  and  $\Omega_n^2 = \{(0,0), (1,0), (2,0), (3,0), (0,1), (1,1)\}$ . Figure 3.3 shows the sets of states  $X_n^l$  which define each of these macrostates. For example, while in state  $\mathbf{x} = \mathbf{x}_{17}$ , the network is in the macrostate  $\mathbf{n} = \mathbf{n}(\mathbf{x}_{17}) = (1,0,1,1)$ , where  $\mathbf{n}^1 = (1,0)$  and  $\mathbf{n}^2 = (1,1)$ , i.e. the network carries a class-1 and a class-2 connection, where the traffic load is distributed between the links as defined in  $\mathbf{n}^1$  and  $\mathbf{n}^2$ . However, as seen in Fig 3.3, the link macrostate  $\mathbf{n}^1 = (1,0)$  is also observable when the network is in any state  $\mathbf{x} \in X_{(1,0)}^1 = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_{17}, \mathbf{x}_{18}\}$ , whereas  $\mathbf{n}^2 = (1,1)$  when the state is  $\mathbf{x} \in X_{(1,1)}^2 = \{\mathbf{x}_{17}, \mathbf{x}_{18}\}$  and  $\mathbf{n} = (1,0,1,1)$  when  $\mathbf{x} \in \{\mathbf{x}_{17}, \mathbf{x}_{18}\}$ . This evinces the fact that different spectrum configurations  $\mathbf{x}$  may yield the same macrostates.

In summary, an optical network serving  $J$  connections classes, with class- $j$  defined by the parameters  $(o, d)_j, \lambda_j, \mu_j^{-1}, b_j, \Gamma_j$ , and  $r_j$ , can be represented by the states  $\mathbf{x} \in \Omega_x$ , where  $\mathbf{x}$  is a valid configuration of the optical spectrum in the network. Each  $\mathbf{x}$  in  $\Omega_x$  fulfils the continuity and contiguity constrains, and defines a carried load  $\mathbf{n} = \mathbf{n}(\mathbf{x})$ . In what follows we use this stochastic network model to study online resource allocation as a reward maximization problem.

## 3.2. Policy-based Online Resource Allocation

Dynamic optical networks allow class- $j$  connections to be requested between a pair of nodes  $(o, d)_j$  on demand. For that, flex-grid WDM, directionless/colorless ROADMs and tunable transponders are used. Blocking of a class- $j$  connection request occurs either if transponders are unavailable in the shared pools at the source and destination ROADMs, or if a suitable lightpath cannot be found within the network. Low transponder-related blocking probabilities (e.g. less than  $10^{-4}$ ) can easily be achieved by a proper dimensioning of the transponder pools; see for example the method in [SCG+12, SW10]. On the other hand, to reduce the blocking probability due to the unavailability of lightpaths in the network, a suitable method for online resource allocation is needed. We refer to online resource allocation as the mechanism whereby the network assigns, on demand, spectrum resources to connections (besides link spectrum slots,

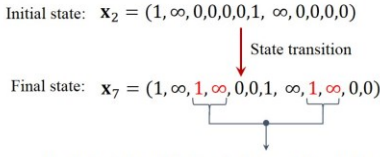
Policy $\Pi = \Pi_1$						
State $\mathbf{x}$	Class- $j$					
	1			2		
	Admit	Path	Slots	Admit	Path	Slots
$\mathbf{x}_1$	Yes	(A,B,C)	1-2	Yes	(B,C)	1-4
$\mathbf{x}_2$	Yes	(A,B,C)	3-4	Yes	(B,C)	3-6
$\mathbf{x}_3$	Yes	(A,B,C)	4-5	No	-----	-----
$\mathbf{x}_4$	Yes	(A,B,C)	1-2	No	-----	-----
$\mathbf{x}_5$	Yes	(A,B,C)	1-2	No	-----	-----
$\mathbf{x}_6$	Yes	(A,B,C)	1-2	Yes	(B,C)	1-4
$\mathbf{x}_7$	Yes	(A,B,C)	5-6	No	-----	-----
$\mathbf{x}_8$	No	-----	-----	No	-----	-----
$\mathbf{x}_9$	Yes	(A,B,C)	3-4	No	-----	-----
$\mathbf{x}_{10}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{11}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{12}$	Yes	(A,B,C)	1-2	No	-----	-----
$\mathbf{x}_{13}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{14}$	Yes	(A,B,C)	5-6	No	-----	-----
$\mathbf{x}_{15}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{16}$	Yes	(A,B,C)	1-2	No	-----	-----
$\mathbf{x}_{17}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{18}$	No	-----	-----	No	-----	-----

Policy $\Pi = \Pi_2$						
State $\mathbf{x}$	Class- $j$					
	1			2		
	Admit	Path	Slots	Admit	Path	Slots
$\mathbf{x}_1$	Yes	(A,B,C)	3-4	Yes	(B,C)	2-5
$\mathbf{x}_2$	Yes	(A,B,C)	4-5	No	-----	-----
$\mathbf{x}_3$	Yes	(A,B,C)	5-6	No	-----	-----
$\mathbf{x}_4$	Yes	(A,B,C)	5-6	No	-----	-----
$\mathbf{x}_5$	Yes	(A,B,C)	2-3	No	-----	-----
$\mathbf{x}_6$	Yes	(A,B,C)	2-3	No	-----	-----
$\mathbf{x}_7$	Yes	(A,B,C)	5-6	No	-----	-----
$\mathbf{x}_8$	No	-----	-----	No	-----	-----
$\mathbf{x}_9$	Yes	(A,B,C)	3-4	No	-----	-----
$\mathbf{x}_{10}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{11}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{12}$	Yes	(A,B,C)	1-2	No	-----	-----
$\mathbf{x}_{13}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{14}$	Yes	(A,B,C)	5-6	No	-----	-----
$\mathbf{x}_{15}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{16}$	Yes	(A,B,C)	1-2	No	-----	-----
$\mathbf{x}_{17}$	No	-----	-----	No	-----	-----
$\mathbf{x}_{18}$	No	-----	-----	No	-----	-----

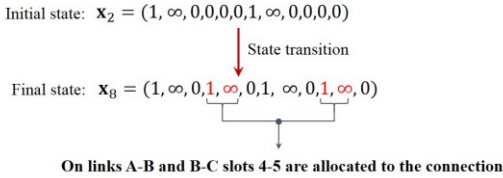
**Figure 3.4:** Example of two possible resource allocation policies for the network in Fig. 3.2.

Class-1 connection request arrives in state  $\mathbf{x}_2$ :

- If the network uses policy  $\Pi_1$ , it executes decision  $\Pi_1(\mathbf{x}_2, 1)$ :

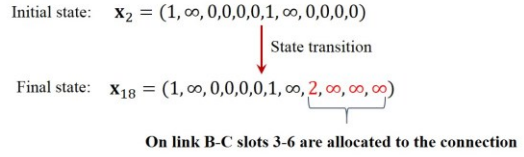


- If the network uses policy  $\Pi_2$ , it executes decision  $\Pi_2(\mathbf{x}_2, 1)$ :

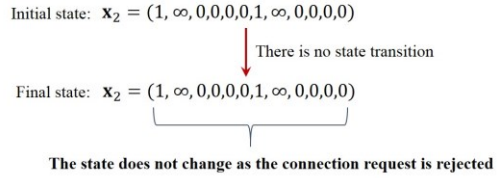


Class-2 connection request arrives in state  $\mathbf{x}_2$ :

- If the network uses policy  $\Pi_1$ , it executes decision  $\Pi_1(\mathbf{x}_2, 2)$ :



- If the network uses policy  $\Pi_2$ , it executes decision  $\Pi_2(\mathbf{x}_2, 2)$ :



**Figure 3.5:** Resource allocation for class-1 and class-2 arrivals when two distinct policies are used.

this includes transponders and ROADM resources as well). Such a mechanism is a decision making process whereby for every connection request, a decision is provided on admission, routing and spectrum assignment. The decision depends on two factors, namely, the class of the connection request and the network state at the time of the connection arrival. Based on this, let  $\Pi$  be a resource allocation policy that instructs the network the course of action to take when a class- $j$  connection request arrives in network state  $\mathbf{x}$ . We can thus visualize a policy  $\Pi$  as a two-dimensional matrix. The number of rows and columns equals the size of the network state-space  $\Omega_{\mathbf{x}}$  and the total number  $J$  of connection classes, respectively. A matrix entry represents a decision. Upon arrival of a connection request, the network retrieves from the matrix the entry that matches the current network state (row) and the connection class (column). That entry indicates whether to accept the request or not. In case of acceptance, the entry further specifies the configuration of the lightpath allocated to the connection (i.e. the route and the spectrum slots that make up the optical channel). Thus, any definable policy  $\Pi$  determines how to handle connection requests.

**Example 3.6** In Fig. 3.4, two possible policies  $\Pi$  are shown for the network in Fig. 3.2. Each policy consists of 36 decisions (18 per connection class). A policy decision  $\Pi(\mathbf{x}, j)$  tells the network what to do if a class- $j$  connection request arrives in state  $\mathbf{x}$ . For these two policies, consider the example in Fig 3.5. Assume that in state  $\mathbf{x} = \mathbf{x}_2$  a class-1 arrival occurs. If policy  $\Pi_1$  is used, then the matrix entry  $\Pi_1(\mathbf{x}_2, 1)$  in Fig. 3.4 instructs the network to admit the connection, to establish it on the path  $\rho = (A, B, C) \in \Gamma_1$ , with an optical channel defined by the slots 3-4 (i.e. on links A-B and B-C these slots are allocated to the

connection). As a result of this decision, the state transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_7$  occurs. On the contrary, if policy  $\Pi_2$  is used to allocate resources, then the connection is admitted on the same path, but using on both links the slots 4-5. In this case, the state transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_8$  occurs. Moreover, with  $\Pi_1$ , if a class-2 arrival occurs in network state  $\mathbf{x} = \mathbf{x}_2$ , the connection seizes slots 3-6 in the path (B, C), thus eliciting a transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$ . With  $\Pi_2$  this request is rejected and thus, there is no state transition (see Fig. 3.5).

### 3.2.1 Properties of a Resource Allocation Policy

In an operational network two kinds of events may cause a state transition (i.e. a change on the spectrum configuration  $\mathbf{x}$ ), these are: the departure of a carried connection or the admission of a connection request. In the former event resources are released, whereas in the latter, resources are provisioned in accordance with a decision made by a policy  $\Pi$ . Regardless of the policy used by the network, we consider the case in which all decisions  $\Pi(\mathbf{x}, j)$  in a policy  $\Pi$  have the following properties:

1. Every decision  $\Pi(\mathbf{x}, j)$  is time-independent and deterministic, i.e. given a class- $j$  arrival in state  $\mathbf{x}$ , the decision  $\Pi(\mathbf{x}, j)$  is always the same and does not depend on the time instant at which the arrival occurs.
2. Every decision  $\Pi(\mathbf{x}, j)$  which grants admission to a class- $j$  request arriving in state  $\mathbf{x}$ , causes a state transition  $\mathbf{x} \rightarrow \mathbf{y}$  such that  $\mathbf{y} \in \Omega_{\mathbf{x}}$ . Therefore, any decision that grants admission has to produce a new spectrum configuration  $\mathbf{y}$  which meets the contiguity, continuity and capacity constraints, i.e. Equations (3.1)-(3.3).
3. There can be policy decisions  $\Pi(\mathbf{x}, j)$  that deny admission despite the availability of resources to provision a lightpath for a connection request.

Any definable policy  $\Pi$  has  $J \times |\Omega_{\mathbf{x}}|$  decisions that fulfil the aforementioned properties, where  $|\Omega_{\mathbf{x}}|$  is the cardinality of the network state-space. Once a network selects a policy  $\Pi$  to allocate resources, all future state transitions due to incoming requests are given by the decisions defined in the policy matrix  $\Pi$ . Those decisions determine the network performance.

**Example 3.7** For the two policies in Fig. 3.4, their decisions produce state transitions  $\mathbf{x} \rightarrow \mathbf{y}$  for which  $\mathbf{y} \in \Omega_{\mathbf{x}}$ . On the other hand, for class-2 requests, there is a remarkable difference between both policies. As seen in Fig. 3.2, these requests could be admitted when they arrive in states  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  or  $\mathbf{x}_6$  (as in those states lightpaths can be found without violating the spectrum constraints), however, unlike  $\Pi_1$ , policy  $\Pi_2$  only admits class-2 requests when they arrive in state  $\mathbf{x} = \mathbf{x}_1$ . This illustrates that policies can be defined that deny admission despite the availability of resources in the network.

### 3.2.2 Policy Calculation as a Decision Making Process

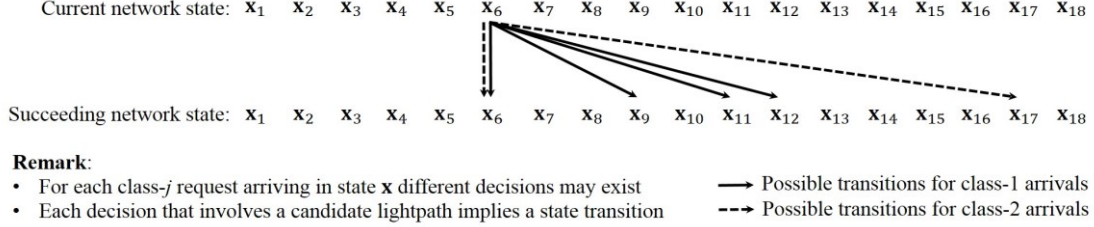
So far we have assumed that a network uses a policy  $\Pi$  to perform resource allocation. Yet an important issue remains to be clarified: the mechanism by which the policy  $\Pi$  is calculated and selected from the set of definable policies. By policy calculation we refer to the definition of all decisions  $\Pi(\mathbf{x}, j)$  that make up the policy matrix  $\Pi$ . The solution to this problem depends on the performance objective fixed by the network operator (e.g. minimization of total blocking probability, maximization of carried traffic or economic revenue). Given an objective, policy  $\Pi$  is then the resource allocation strategy that yields the best network performance. We can think of the calculation of that policy as a decision making process whereby for each pair  $(\mathbf{x}, j)$  an optimum decision  $\Pi(\mathbf{x}, j)$  is determined. The reason is that, for a class- $j$  request arriving in state  $\mathbf{x}$ , different courses of action can be available in the network. To evaluate and select one of those possibilities, online RSA and CAC algorithms are needed. The goal of RSA is to define all possible decisions (i.e. courses of action) available for a class- $j$  request in state  $\mathbf{x}$ . With this, admission control selects the optimum decision  $\Pi(\mathbf{x}, j)$ . Thus, for a class- $j$  request arriving in state  $\mathbf{x}$ , the optimum decision is calculated as follows:

1. An RSA algorithm calculates the set of candidate lightpaths available on all routes  $\rho \in \Gamma_j$ . Each lightpath is a potential decision which, if selected, elicits a transition  $\mathbf{x} \rightarrow \mathbf{y}$  such that  $\mathbf{y} \in \Omega_{\mathbf{x}}$ . If no candidate lightpaths are found, then resources are not available and the decision is made to reject the connection request (i.e.  $\Pi(\mathbf{x}, j) \rightarrow$  Do not admit).

Decision 1				Decision 2				Decision 3				Decision 4			
State $x$	Class-1			State $x$	Class-1			State $x$	Class-1			State $x$	Class-1		
	Admit	Path	Slots		Admit	Path	Slots		Admit	Path	Slots		Admit	Path	Slots
$x_2$	Yes	(A,B,C)	3-4	$x_2$	Yes	(A,B,C)	4-5	$x_2$	Yes	(A,B,C)	5-6	$x_2$	No	-----	-----

State transition:  $x_2 \rightarrow x_7$       State transition:  $x_2 \rightarrow x_8$       State transition:  $x_2 \rightarrow x_9$       There is no state transition

**Figure 3.6:** Possible decisions for a class-1 request arriving in state  $x_2$ .



**Figure 3.7:** Possible decisions when either a class-1 or a class-2 arrival occurs in state  $x_6$ .

2. Given the set of candidate lightpaths, admission control decides whether to accept or reject the connection. To make this decision, admission decision rules are used that aim at optimizing the network performance (based on the optimization objective). If the connection is accepted, then a candidate lightpath is selected. (Note that despite the availability of resources a possible course of action is to not accept the connection request.) The resulting decision is  $\Pi(x, j)$ .

The policy calculation approach evinces that the optimization objective is attainable through the rules that make decisions. The definition of these rules will be discussed in Chapter 4.

**Example 3.8** For the network in Fig. 3.2, consider a class-1 arrival in state  $x_2$ . The decision making process must select an optimum decision from the four possibilities shown in Fig. 3.6. The first three correspond to the candidate lightpaths calculated by solving the RSA problem (all of them are routed on the path  $\rho = (A, B, C) \in \Gamma_1$ ); the last possibility is connection rejection. In Fig. 3.6, it is depicted the state transition that each decision would trigger, if selected. The optimization objective defines which option is the optimum.

**Example 3.9** The possible decisions available for a class- $j$  request arriving in state  $x$  can be represented by a state-transition diagram in which every transition defines a decision. For the network in Fig. 3.2, the diagram in Fig. 3.7 depicts all possible decisions when either a class-1 or a class-2 request arrives in state  $x_6$ . Class-1 and class-2 have four and two courses of action, respectively (represented by the arrows in the diagram). For both classes the decision “reject connection” (depicted by the arrows connecting the current to the succeeding state  $x_6$ , i.e. no state transition) is a possibility, as having resources available need not imply that the selection of a lightpath yields optimum performance. The succeeding states which are reachable when a request is accepted implicitly define a lightpath. To illustrate this, consider the transition  $x_6 \rightarrow x_{11}$ , which occurs when a class-1 request arrives in  $x_6$ . In this case, the connection seizes slots 2-3 on the path (A, B, C), and thus, the network undergoes a transition from  $x_6 = (0, 0, 0, 0, 1, \infty, 0, 0, 0, 0, 1, \infty)$  to  $x_{11} = (0, 1, \infty, 0, 1, \infty, 0, 1, \infty, 0, 1, \infty)$ .

For a given performance objective, an ideal approach to calculating the optimum policy is to determine all its decisions before the network gets operational and store them in memory. However, in real networks, this approach is not appropriate as the policy decisions cannot adapt to changes in the parameters  $(o, d)_j$ ,  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$ . Thus, to make the policy calculation adaptable, we consider an approach in which all policy decisions are calculated during network operation, with the optimization objective defined as the rate at which the network earns reward from carried connections.

### 3.3. Network Reward Maximization Problem

In this section we outline a reward-based approach to online resource allocation which aims maximizing the average reward carried by the network. For that, let us consider the case in which online resource allocation is performed in the network via a policy  $\Pi$ . The class- $j$  blocking probability  $B_j$  depends on the decisions defined in the policy. Therefore, if class- $j$  offers to the network a traffic load of  $A_j$  Erlangs, the policy dependent blocking  $B_j$  yields a carried traffic of  $Ac_j = A_j \cdot (1 - B_j)$  Erlangs. The reward earned by the network stems from this carried traffic. To illustrate this, consider a class- $j$  connection that gets admission. During its holding time  $\mu_j^{-1}$ , it yields a reward of  $r_j$  (ru), which means that from a carried

class- $j$  connection the network earns reward at a mean rate of  $r_j/\mu_j^{-1} = r_j \cdot \mu_j$  reward units per unit of time (ru/uot). Since a carried traffic load of  $Ac_j$  Erlangs means that on average  $Ac_j$  class- $j$  connections are carried simultaneously, then class- $j$  brings reward at a rate of  $r_j \cdot \mu_j \cdot Ac_j$  (ru/uot). Thus, a network that serves  $J$  classes earns reward at a mean rate  $R$  given by:

$$R = \sum_{j=1}^J r_j \cdot \mu_j \cdot Ac_j = \sum_{j=1}^J r_j \cdot \lambda_j \cdot (1 - B_j) \quad (3.4)$$

Besides the policy  $\Pi$  and the link capacities  $C_l$ , the network reward rate  $R$  depends on the parameters  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $r_j$  and  $\Gamma_j$ . (This dependence is further justified by the fact that, in addition to  $\Pi$ , the blocking probabilities  $B_j$  depend on the capacities  $C_l$  and on all the parameters defining the connection classes.) In an operational network, those parameters and the link capacities  $C_l$  are known. With that, we have that  $R = R(\Pi)$ . In particular,  $\lambda_j$  and  $\mu_j^{-1}$  stem from the statistical properties of class- $j$  traffic, and thus, they can be estimated from online measurements. Moreover,  $b_j$ ,  $r_j$  and  $\Gamma_j$  are defined by, and therefore known to, the network operator. Based on this knowledge, the online resource allocation problem is formulated as follows. Given a dynamic optical network serving  $J$  classes defined by the parameters  $(o, d)_j$ ,  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$ , calculate the policy  $\Pi$  which maximizes the network reward rate:

$$R^* = \max_{\Pi} R(\Pi) \quad (3.5)$$

The problem is formulated as an online optimization problem as the decisions  $\Pi(\mathbf{x}, j)$  that make up the policy matrix  $\Pi$  are calculated during network operation.

### 3.3.1 Definition of Optimization Objectives

Research on resource allocation for dynamic optical networks has primarily focused on the design of online RSA algorithms. In [TAK<sup>+</sup>14], a complete survey is presented on existing online RSA methods. The approaches therein surveyed, e.g. [WWH<sup>+</sup>11], [CVR<sup>+</sup>12], are limited in the sense that they solely aim at minimizing the overall blocking probability. This limitation is overcome by tackling resource allocation as reward-based problem. The advantage of this approach is that the reward parameters  $r_j$  can be tuned to optimize an objective arbitrarily defined by the network operator. Furthermore, it allows GoS control for individual connection classes.

If the reward parameters are defined as  $r_j = 1, \forall j$ , the network reward rate  $R$  is interpreted as the rate at which the network accepts connections. (In this case  $R$  is expressed in connections per time unit.) Thus, by maximizing  $R$ , the policy is found that minimizes the overall blocking. The objective function is:

$$R(\Pi) = \sum_{j=1}^J \lambda_j \cdot (1 - B_j) \quad (3.6)$$

Alternatively, if the reward parameters are defined as  $r_j = b_j/\mu_j$ , the reward rate  $R$  represents the overall throughput. In this case the objective is:

$$R(\Pi) = \sum_{j=1}^J b_j \cdot Ac_j \quad (3.7)$$

where the product  $b_j \cdot Ac_j$  denotes the mean number of spectrum slots seized by class- $j$  traffic, i.e. the average class- $j$  throughput. Another interesting case is revenue maximization. In a telecom market with a high degree of competition, the optimum price of a connection is that which is proportional to its bandwidth [CW03]. Thus, the network operator can set the price for a class- $j$  connection as  $r_j = \alpha_j \cdot b_j$ , where  $\alpha_j$  is a proportionality constant. Economic revenue is then maximized by defining the objective:

$$R(\Pi) = \sum_{j=1}^J \alpha_j \cdot b_j \cdot \lambda_j \cdot (1 - B_j) \quad (3.8)$$

If  $\alpha_j = 1/\mu_j$ , it is verified that Equation (3.7) is a special case of Equation (3.8). The aforementioned objective functions point out the flexibility of a reward-based approach. In general, the objective function defined by Equation (3.4) can be tailored to different optimization objectives.

Reward-based approaches have been proposed in the literature to tackle the routing problem for telephone and multiservice packet-switched networks. Relevant algorithms and performance evaluation results are expounded in [DPKW88, DM89, Kri91, DM92, Hwa93, DM94, Dzi97, HKT00, Nor02]. These approaches are not applicable to solve routing in dynamic optical networks as they do not consider the

spectrum continuity and contiguity constraints. In our approach, we include these constraints and apply the reward concept to calculate policies that solve Equation (3.5). The constraints are enforced by the definition of state-dependent policies which provide decisions  $\Pi(\mathbf{x}, j)$  that cause state transitions  $\mathbf{x} \rightarrow \mathbf{y}$  such that  $\mathbf{y} \in \Omega_{\mathbf{x}}$ . To solve this optimization problem, we study the case in which the network exhibits the properties of a Markov stochastic process, thereby the theory of Markov decision processes [How60] is applied to calculate the policy that solves Equation (3.5).

### 3.3.2 State-Dependent Network Reward Rate

Let  $\{\mathbf{X}_t^\pi, t \geq 0\}$  be a continuous-time stochastic process that models the time evolution of the network state when a policy  $\Pi$  is used. Each random variable  $\mathbf{X}_t^\pi$  takes its values from the state-space  $\Omega_{\mathbf{x}}$  and thereby,  $\mathbf{X}_t^\pi$  is the network state at time  $t \in [0, \infty)$ . Consider the case where class- $j$  connections arrive at a rate  $\lambda_j$  following a Poisson process, and have negative exponentially distributed holding times with mean  $\mu_j^{-1}$ . Thus, the stochastic process  $\{\mathbf{X}_t^\pi, t \geq 0\}$  is Markovian as for all  $t_0 < t_1 < \dots < t_n < t_{n+1}$ , and  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_{n+1} \in \Omega_{\mathbf{x}}$ , it satisfies:

$$P\{\mathbf{X}_{t_{n+1}}^\pi = \mathbf{x}_{n+1} | \mathbf{X}_{t_n}^\pi = \mathbf{x}_n, \mathbf{X}_{t_{n-1}}^\pi = \mathbf{x}_{n-1}, \dots, \mathbf{X}_{t_0}^\pi = \mathbf{x}_0\} = P\{\mathbf{X}_{t_{n+1}}^\pi = \mathbf{x}_{n+1} | \mathbf{X}_{t_n}^\pi = \mathbf{x}_n\} \quad (3.9)$$

i.e. the probability of observing the network in state  $\mathbf{x}_{n+1}$  at time  $t_{n+1}$  depends only on the present state  $\mathbf{x}_n$  at time  $t_n$ . For this process, the sojourn time in state  $\mathbf{x}$  is a random variable exponentially distributed since the interarrival and holding times have the same distribution type. Therefore, due to the memoryless property of the exponential distribution, the sojourn time in state  $\mathbf{x}$  is independent from the states visited before entering the state  $\mathbf{x}$ . We denote the mean sojourn time in state  $\mathbf{x}$  as  $\tau_{\mathbf{x}}$ .

Let us define  $P_{\mathbf{xy}}(\Pi) \geq 0$  as the probability that the process  $\{\mathbf{X}_t^\pi, t \geq 0\}$  moves from state  $\mathbf{x}$  to state  $\mathbf{y}$ , with  $\mathbf{x}, \mathbf{y} \in \Omega_{\mathbf{x}}$ , and  $\mathbf{x} \neq \mathbf{y}$ . These transition probabilities fulfil the condition:

$$\sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} P_{\mathbf{xy}}(\Pi) = 1, \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.10)$$

Note that  $P_{\mathbf{xy}}(\Pi)$  is policy-dependent since in the case of transitions caused by admissions, the network moves to states  $\mathbf{y}$  defined by  $\Pi$ . When a transition  $\mathbf{x} \rightarrow \mathbf{y}$  occurs, the network earns  $r_{\mathbf{xy}}(\text{ru})$ , the actual value of this reward depends on the type of transition. For instance, a transition that results from the admission of a class- $j$  request arriving in state  $\mathbf{x}$ , has a reward  $r_{\mathbf{xy}} = r_j$ .

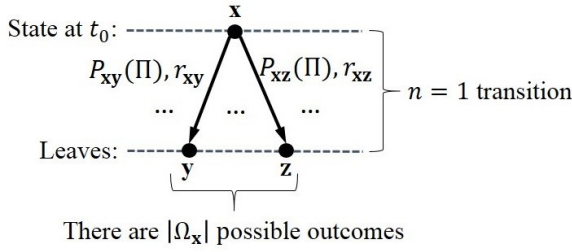
To calculate the reward rate  $R = R(\Pi)$ , consider a network using a policy  $\Pi$  which at an arbitrary time  $t_0$  is in state  $\mathbf{X}_{t_0}^\pi = \mathbf{x}$ . Let  $V(\mathbf{x}, \Pi, n)$ ,  $n \in \mathbb{Z}^+$ , be the average reward earned by the network after  $n$  subsequent transitions from state  $\mathbf{x}$  (each transition is caused either by a connection arrival or departure). To estimate  $V(\mathbf{x}, \Pi, n)$ , consider the probabilistic trees in Fig. 3.8, where, for a given  $n$ , a tree depicts all possible state transitions starting from state  $\mathbf{x}$ . In a tree, each node is a state in  $\Omega_{\mathbf{x}}$ , and the root node is the state  $\mathbf{x}$  at  $t_0$ . A tree branch (or edge) depicts a transition between the two states it interconnects. A leaf in a tree represents a destination state reachable after  $n$  transitions, and the leaf is connected to the root node by a path made up of a sequence of branches. The sum of the rewards from the branches in a path is the reward earned when the network follows the state transitions that define the path. Thus,  $V(\mathbf{x}, \Pi, n)$  is an average calculated over all the rewards from the paths that connect the root to the leaf states (i.e. to the possible destination states after  $n$  transitions). To illustrate this better, consider the tree for  $n = 1$ . This tree has  $|\Omega_{\mathbf{x}}|$  leaves, each one connected to the root node  $\mathbf{x}$  by a path with a single branch. The path  $\mathbf{x} \rightarrow \mathbf{y}$  is the one-step transition between states  $\mathbf{x}$  and  $\mathbf{y}$ . This event occurs with a probability  $P_{\mathbf{xy}}(\Pi)$ , and when it does, a reward  $r_{\mathbf{xy}}$  is earned. Thus, if the network is in state  $\mathbf{x}$  at  $t_0$ , the expected reward in one transition is:

$$V(\mathbf{x}, \Pi, 1) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} r_{\mathbf{xy}} \cdot P_{\mathbf{xy}}(\Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.11)$$

which is a weighted average calculated over all rewards  $r_{\mathbf{xy}}$ . For  $n = 2$ , we have that in the first transition the network may move from state  $\mathbf{x}$  to a state  $\mathbf{y}$  with a probability  $P_{\mathbf{xy}}(\Pi)$ . If this event occurs, a reward  $r_{\mathbf{xy}}$  is earned, and the network is then set in state  $\mathbf{y}$ . In the second transition, the reward contributions is that obtained when the network follows a one-step transition from state  $\mathbf{y}$ , namely  $V(\mathbf{y}, \Pi, 1)$ . Therefore, for  $n = 2$ , if in the first move the transition  $\mathbf{x} \rightarrow \mathbf{y}$  occurs, then a reward  $r_{\mathbf{xy}} + V(\mathbf{y}, \Pi, 1)$  is expected at the end of the second move. This reward is earned with probability  $P_{\mathbf{xy}}(\Pi)$ . Then, in two transitions, the reward is the weighted average over all transitions  $\mathbf{x} \rightarrow \mathbf{y}$  which may occur in the first move (see Fig. 3.8):

$$V(\mathbf{x}, \Pi, 2) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} [r_{\mathbf{xy}} + V(\mathbf{y}, \Pi, 1)] \cdot P_{\mathbf{xy}}(\Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.12)$$

**Average reward earned in  $n = 1$  transition**

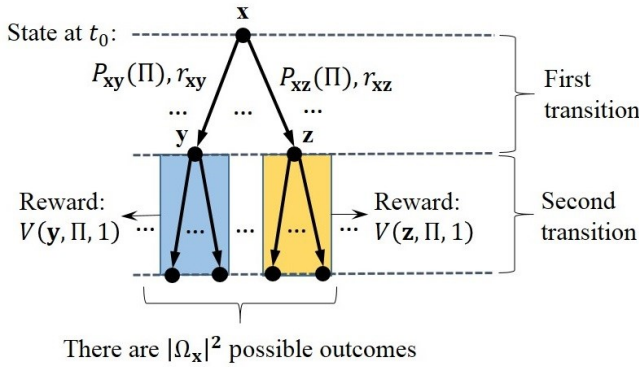


A reward of  $r_{xy}$  ( $ru$ ) is earned if in one transition the network moves from state  $x$  to state  $y$ . This event occurs with a probability  $P_{xy}(\Pi)$ . In general, from state  $x$  there are as many possible one-step transitions as states in the state-space  $\Omega_x$ . Therefore, the expected reward earned in one transition is calculated as the weighted average over all state transitions from  $x$ .

Expected reward in one transition:

$$V(x, \Pi, 1) = \sum_{y: y \neq x} r_{xy} \cdot P_{xy}(\Pi)$$

**Average reward earned in  $n = 2$  transitions**

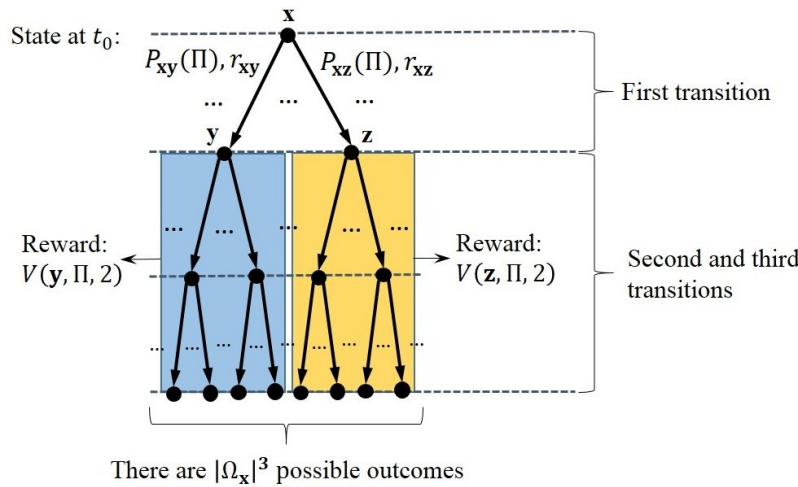


If in the first transition the network moves from state  $x$  to state  $y$ , a reward of  $r_{xy}$  ( $ru$ ) is earned. Since the network is now in state  $y$ , in the second transition the expected reward is  $V(y, \Pi, 1)$ , i.e. the reward expected in one transition when the network is in state  $y$ . Therefore,  $r_{xy} + V(y, \Pi, 1)$  ( $ru$ ) are earned with probability  $P_{xy}(\Pi)$ .

Expected reward in two transitions:

$$V(x, \Pi, 2) = \sum_{y: y \neq x} [r_{xy} + V(y, \Pi, 1)] \cdot P_{xy}(\Pi)$$

**Average reward earned in  $n = 3$  transitions**



If in the first transition the network moves from state  $x$  to state  $y$ , a reward of  $r_{xy}$  ( $ru$ ) is earned. Since the network is now in state  $y$ , in the remaining two transitions  $V(y, \Pi, 2)$  ( $ru$ ) are expected, i.e. the reward in two transitions when the network is in state  $y$ . Therefore,  $r_{xy} + V(y, \Pi, 2)$  ( $ru$ ) are earned with probability  $P_{xy}(\Pi)$ .

Expected reward in three transitions:

$$V(x, \Pi, 3) = \sum_{y: y \neq x} [r_{xy} + V(y, \Pi, 2)] \cdot P_{xy}(\Pi)$$

**Figure 3.8:** Probabilistic trees to calculate the rewards  $V(x, \Pi, n)$ .

Similarly, for  $n = 3$  we have:

$$V(\mathbf{x}, \Pi, 3) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} [r_{\mathbf{xy}} + V(\mathbf{y}, \Pi, 2)] \cdot P_{\mathbf{xy}}(\Pi) \quad , \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.13)$$

i.e. if the first transition is  $\mathbf{x} \rightarrow \mathbf{y}$ , then in three moves a mean reward  $r_{\mathbf{xy}} + V(\mathbf{y}, \Pi, 2)$  is expected, where  $r_{\mathbf{xy}}$  (ru) comes from the first transition (which sets the network in state  $\mathbf{y}$ ), and  $V(\mathbf{y}, \Pi, 2)$  (ru) are earned through the remaining two moves starting in state  $\mathbf{y}$ . By recursively applying Equations (3.11)-(3.13), if the network is in state  $\mathbf{x}$  at  $t_0$ , then at time  $t \gg t_0$ , after  $n$  transitions, the expected earned reward is:

$$V(\mathbf{x}, \Pi, n) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} [r_{\mathbf{xy}} + V(\mathbf{y}, \Pi, n - 1)] \cdot P_{\mathbf{xy}}(\Pi) \quad , \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.14)$$

which is a form of the Bellman's recurrence relation for dynamic programming shown in [Bel57, How60]. A caveat needs to be pointed out about Equation (3.14): it has been derived by intuitively using the concept of state transition. However, it must be clarified that each of the  $n$  transitions occurs at instants of time which in general are not deterministic, i.e. they are random, unknown. The reason is that the process  $\{\mathbf{X}_t^\pi, t \geq 0\}$  is continuous-time. The implications of this are considered in the definition of the transition probabilities  $P_{\mathbf{xy}}(\Pi)$ .

If in  $n$  transitions a reward  $V(\mathbf{x}, \Pi, n)$  is earned, the average reward per transition is therefore given by  $V(\mathbf{x}, \Pi, n)/n$ . As the number of transitions becomes very large, so that the network attains statistical equilibrium, the average reward earned per transition approaches a steady-state value  $g(\Pi)$  defined as:

$$\lim_{n \rightarrow \infty} \frac{V(\mathbf{x}, \Pi, n)}{n} = g(\Pi) \quad (3.15)$$

which fulfils:

$$g(\Pi) \leq \max_{\mathbf{x}, \mathbf{y}} r_{\mathbf{xy}} \quad (3.16)$$

which follows from the fact that in one transition, the maximum actual reward is that obtained from the transition  $\mathbf{x} \rightarrow \mathbf{y}$  with the highest value  $r_{\mathbf{xy}}$ . In [How60] it is proved that, for large  $n$  (and therefore, for  $t \gg t_0$ ) a recurrence relation like Equation (3.14) can be approximated by the linear equation:

$$V(\mathbf{x}, \Pi, n) = g(\Pi) \cdot n + v(\mathbf{x}, \Pi) \quad , \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.17)$$

The product  $g(\Pi) \cdot n$  is interpreted as the steady-state reward earned over  $n$  transitions (this reward is obtained when  $n$  is very large, where per move,  $g(\Pi)$  (ru) are earned). The term  $v(\mathbf{x}, \Pi)$  is the transient reward earned by having set the network in state  $\mathbf{x}$  at  $t_0$ . Equation (3.17) states that the network state  $\mathbf{x}$  at  $t_0$  determines the total reward earned at  $t \gg t_0$  (i.e. after  $n$  transitions). Thus, different states  $\mathbf{x}$  may lead to different rewards  $V(\mathbf{x}, \Pi, n)$ . For each  $\mathbf{x} \in \Omega_{\mathbf{x}}$ , Equation (3.17) describes a straight line with slope  $g(\Pi)$  and intercept  $v(\mathbf{x}, \Pi)$ . By plotting the lines describable by all states in  $\Omega_{\mathbf{x}}$ , as depicted in Fig. 3.9, we obtain a family of parallel lines, which means that regardless of the state  $\mathbf{x}$  at  $t_0$ , the network always earns reward at the same mean rate  $g(\Pi)$  per transition. For  $t \gg t_0$ , the reward  $V(\mathbf{x}, \Pi, n)$  only differs from other states in the transient value  $v(\mathbf{x}, \Pi)$ . Therefore, any pair of states  $\mathbf{x}$  and  $\mathbf{y}$  can be compared by calculating from Equation (3.17) the difference:

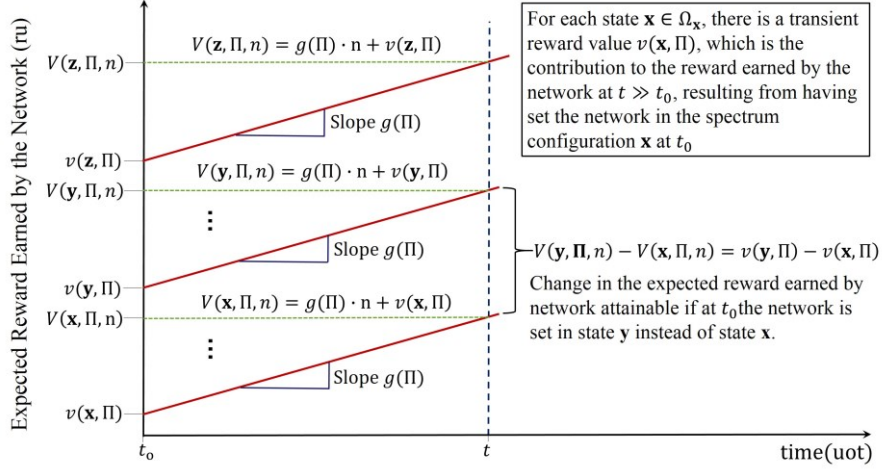
$$V(\mathbf{y}, \Pi, n) - V(\mathbf{x}, \Pi, n) = v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi) \quad (3.18)$$

which is interpreted as: if the network is in state  $\mathbf{x}$  at  $t_0$ , a reward  $V(\mathbf{x}, \Pi, n)$  would be earned at  $t \gg t_0$ . But, if the network had been in state  $\mathbf{y}$  at  $t_0$  instead, the reward  $V(\mathbf{y}, \Pi, n)$  at  $t \gg t_0$  is expected to differ in  $v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$  (ru) with respect to  $V(\mathbf{x}, \Pi, n)$ . The striking property of this result, as shown by the right hand side of Equation (3.18), is that the reward difference for  $t \gg t_0$  is independent of  $n$ . This evinces that, for a given policy  $\Pi$ , there can be states more valuable than others. For example, in Fig. 3.9, we have that  $v(\mathbf{z}, \Pi) > v(\mathbf{y}, \Pi) > v(\mathbf{x}, \Pi)$ . As a result of this, if at  $t_0$  the network is in state  $\mathbf{z}$ , at  $t \gg t_0$ , the reward  $V(\mathbf{z}, \Pi, n)$  is higher than that obtained if the network had been in state  $\mathbf{y}$  or  $\mathbf{x}$  at  $t_0$ . This implies that  $V(\mathbf{z}, \Pi, n) > V(\mathbf{y}, \Pi, n) > V(\mathbf{x}, \Pi, n)$ .

From Equation (3.17), the reward  $V(\mathbf{x}, \Pi, n - 1)$  obtained in  $n - 1$  transitions is:

$$V(\mathbf{x}, \Pi, n - 1) = g(\Pi) \cdot (n - 1) + v(\mathbf{x}, \Pi) \quad , \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.19)$$





**Figure 3.9:** Expected reward  $V(\mathbf{x}, \Pi, n)$  earned by a network until time  $t \gg t_0$  after  $n$  state transitions.

By plugging Equations (3.17) and (3.19) into Equation (3.14) we have that  $V(\mathbf{x}, \Pi, n)$  is:

$$g(\Pi) \cdot n + v(\mathbf{x}, \Pi) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} [r_{\mathbf{xy}} + g(\Pi) \cdot (n-1) + v(\mathbf{y}, \Pi)] \cdot P_{\mathbf{xy}}(\Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.20)$$

which equals to:

$$g(\Pi) \cdot n + v(\mathbf{x}, \Pi) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} [r_{\mathbf{xy}} + v(\mathbf{y}, \Pi)] \cdot P_{\mathbf{xy}}(\Pi) + g(\Pi) \cdot (n-1) \cdot \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} P_{\mathbf{xy}}(\Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.21)$$

Since  $\sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} P_{\mathbf{xy}}(\Pi) = 1$ , from Equation (3.21), the reward  $g(\Pi)$  is given by:

$$g(\Pi) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} r_{\mathbf{xy}} \cdot P_{\mathbf{xy}}(\Pi) + \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} v(\mathbf{y}, \Pi) \cdot P_{\mathbf{xy}}(\Pi) - v(\mathbf{x}, \Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.22)$$

The probabilities  $P_{\mathbf{xy}}(\Pi)$  are defined by the transition matrix  $\mathbf{Q}(\Pi) = [q_{\mathbf{xy}}(\Pi)]$  of the continuous-time Markov process  $\{\mathbf{X}_t^{\Pi}, t \geq 0\}$ , where  $q_{\mathbf{xy}}(\Pi) \geq 0, \mathbf{x} \neq \mathbf{y}$ , is the rate at which the network moves from state  $\mathbf{x}$  to state  $\mathbf{y}$ . The diagonal elements  $q_{\mathbf{xx}}(\Pi)$  are given by [Kel11, KY14, Nor97]:

$$\tau_{\mathbf{x}}^{-1} = -q_{\mathbf{xx}}(\Pi) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} q_{\mathbf{xy}}(\Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.23)$$

which means that the inverse of the sojourn time in state  $\mathbf{x}$ , i.e.  $\tau_{\mathbf{x}}^{-1}$ , is the rate at which the network moves from  $\mathbf{x}$  to another state in  $\Omega_{\mathbf{x}}$ . The transition probabilities  $P_{\mathbf{xy}}(\Pi)$  are defined as [KY14, Nor97]:

$$P_{\mathbf{xy}}(\Pi) = q_{\mathbf{xy}}(\Pi) \cdot \tau_{\mathbf{x}} \quad (3.24)$$

From this, Equation (3.22) is expressed as:

$$g(\Pi) = \tau_{\mathbf{x}} \cdot \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} r_{\mathbf{xy}} \cdot q_{\mathbf{xy}}(\Pi) + \tau_{\mathbf{x}} \cdot \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} v(\mathbf{y}, \Pi) \cdot q_{\mathbf{xy}}(\Pi) - v(\mathbf{x}, \Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.25)$$

by dividing Equation (3.25) by  $\tau_{\mathbf{x}}$ :

$$\frac{g(\Pi)}{\tau_{\mathbf{x}}} = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} r_{\mathbf{xy}} \cdot q_{\mathbf{xy}}(\Pi) + \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} v(\mathbf{y}, \Pi) \cdot q_{\mathbf{xy}}(\Pi) - \frac{v(\mathbf{x}, \Pi)}{\tau_{\mathbf{x}}}, \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.26)$$

Observe that by using Equation (3.23) we have that  $v(\mathbf{x}, \Pi)/\tau_{\mathbf{x}} = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} q_{\mathbf{xy}}(\Pi) \cdot v(\mathbf{x}, \Pi)$ , and therefore:

$$\frac{g(\Pi)}{\tau_{\mathbf{x}}} = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} r_{\mathbf{xy}} \cdot q_{\mathbf{xy}}(\Pi) + \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} [v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)] \cdot q_{\mathbf{xy}}(\Pi), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.27)$$

Notice that the ratio  $g(\Pi)/\tau_{\mathbf{x}}$  is the average reward earned by the network per unit of time, which is the network reward rate  $R(\Pi)$ . To further simplify Equation (3.27), let us analyse the transition rates  $q_{\mathbf{xy}}(\Pi)$  for a network in state  $\mathbf{x}$ . Recall that while in this state, the network moves to a state  $\mathbf{y} \in \Omega_{\mathbf{x}}$ , if and only if either an arrival or departure occurs. Based on this, in Equation (3.27) the rates  $q_{\mathbf{xy}}(\Pi)$  are defined as:

$$q_{\mathbf{x}\mathbf{y}}(\Pi) = \begin{cases} \lambda_j(\mathbf{x}, \Pi), & \text{if } \mathbf{x} \rightarrow \mathbf{y} \text{ is caused by a class } j \text{ arrival accepted by policy } \Pi \\ \mu_j, & \text{if } \mathbf{x} \rightarrow \mathbf{y} \text{ is caused by a class } j \text{ departure} \\ 0, & \text{if the transition } \mathbf{x} \rightarrow \mathbf{y} \text{ is not feasible} \end{cases} \quad (3.28)$$

Thus, for class- $j$  connections accepted by policy  $\Pi$  in state  $\mathbf{x}$ , we have that  $q_{\mathbf{x}\mathbf{y}}(\Pi) = \lambda_j(\mathbf{x}, \Pi)$  (con/uot), i.e.  $q_{\mathbf{x}\mathbf{y}}(\Pi)$  is the mean number of class- $j$  request arrivals per unit of time when the network is in state  $\mathbf{x}$ . For class- $j$  departures, we have that  $q_{\mathbf{x}\mathbf{y}}(\Pi) = \mu_j$  (con/uot), i.e.  $q_{\mathbf{x}\mathbf{y}}(\Pi)$  is the mean rate at which a class- $j$  connection is terminated in state  $\mathbf{x}$ . Furthermore, we have  $q_{\mathbf{x}\mathbf{y}}(\Pi) = 0$ , for all transitions  $\mathbf{x} \rightarrow \mathbf{y}$  which either violate the contiguity/continuity constraints or represent moves not allowed by  $\Pi$ . Let  $\Gamma_{\mathbf{x}}^{j+}$  and  $\Gamma_{\mathbf{x}}^{j-}$  be the sets of network states which are reachable due to class- $j$  connection admissions and departures in state  $\mathbf{x}$ , respectively. Based on these definitions, Equation (3.27) is expressed as:

$$R(\Pi) = q(\mathbf{x}) + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j+}} \lambda_j(\mathbf{x}, \Pi) \cdot [v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)] + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j-}} \mu_j \cdot [v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)] \quad , \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.29a)$$

with:

$$q(\mathbf{x}) = \sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}} r_{\mathbf{xy}} \cdot q_{\mathbf{xy}}(\Pi) = \sum_{j=1}^J r_j \cdot \mu_j(\mathbf{x}) \quad (3.29b)$$

which is the rate at which the network yields reward in state  $\mathbf{x}$ , with  $\mu_j(\mathbf{x})$  being the termination rate of carried class- $j$  connections in that state. It is calculated as:

$$\mu_j(\mathbf{x}) = \mu_j \cdot n_j(\mathbf{x}) \quad (3.29c)$$

where  $n_j(\mathbf{x}) \geq 0$  is the number of class- $j$  connections carried in state  $\mathbf{x}$ . The first double summation in Equation (3.29a) is the network reward rate due to connection arrivals, whereas the second summation is the network reward rate due to departures. Having defined  $R(\Pi)$  as a function of  $\mathbf{x}$ , Equations (3.17) and (3.18) can respectively be re-written as:

$$V(\mathbf{x}, \Pi, t) = R(\Pi) \cdot t + v(\mathbf{x}, \Pi) \quad , \forall \mathbf{x} \in \Omega_{\mathbf{x}} \quad (3.29d)$$

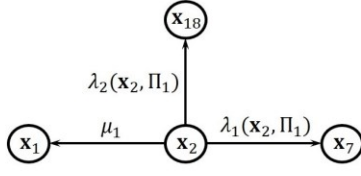
$$V(\mathbf{y}, \Pi, t) - V(\mathbf{x}, \Pi, t) = v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi) \quad (3.29e)$$

with  $V(\mathbf{x}, \Pi, t)$  being the reward expected at  $t \gg t_0$ , if the network were in state  $\mathbf{x}$  at  $t_0$ . As with Equation (3.17), Equation (3.29d) states that in steady state, the network earns reward at a rate  $R(\Pi)$  regardless of the state  $\mathbf{x}$ . Likewise, Equation (3.29e) shows that the difference  $V(\mathbf{y}, \Pi, t) - V(\mathbf{x}, \Pi, t)$  is independent of  $t$ , and thus, it has the same interpretation given for Equation (3.18).

**Example 3.10** Consider the network in Fig. 3.2 in state  $\mathbf{x} = \mathbf{x}_8 = (1, \infty, 0, 1, \infty, 0, 1, \infty, 0, 1, \infty, 0)$ . The network uses the policy  $\Pi_1$  defined in Fig. 3.4. In this state neither class-1 nor class-2 arrivals are admitted, as resources are not available. Thus,  $\Gamma_{\mathbf{x}_8}^{1+} = \{\phi\}$ ,  $\Gamma_{\mathbf{x}_8}^{2+} = \{\phi\}$  which implies  $\lambda_1(\mathbf{x}_8, \Pi_1) = \lambda_2(\mathbf{x}_8, \Pi_1) = 0$ . In state  $\mathbf{x}_8$  the network is carrying two connections of class-1, the termination of either of them causes a state transition. These two transitions are defined as:  $\mathbf{x}_8 \rightarrow \mathbf{x}_2 = (1, \infty, 0, 0, 0, 0, 1, \infty, 0, 0, 0, 0)$  and  $\mathbf{x}_8 \rightarrow \mathbf{x}_5 = (0, 0, 0, 1, \infty, 0, 0, 0, 0, 1, \infty, 0)$ . Hence for class-1 departures  $\Gamma_{\mathbf{x}_8}^{1-} = \{\mathbf{x}_2, \mathbf{x}_5\}$ , whereas for class-2,  $\Gamma_{\mathbf{x}_8}^{2-} = \{\phi\}$ , as class-2 traffic is not being carried. In state  $\mathbf{x}_8$ , the network yields reward at a rate  $q(\mathbf{x}_8) = \sum_{j=1}^2 r_j \cdot \mu_j(\mathbf{x}_8) = r_1 \cdot 2 \cdot \mu_1$  (ru/uot). Moreover, the first double sum in Equation (3.29a) is given by  $\sum_{\mathbf{y} \in \Gamma_{\mathbf{x}_8}^{1+}} \lambda_1(\mathbf{x}_8, \Pi_1) \cdot [v(\mathbf{y}, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] + \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}_8}^{2+}} \lambda_2(\mathbf{x}_8, \Pi_1) \cdot [v(\mathbf{y}, \Pi) - v(\mathbf{x}_8, \Pi_1)] = 0$ . The second double summation is  $\sum_{\mathbf{y} \in \Gamma_{\mathbf{x}_8}^{1-}} \mu_1 \cdot [v(\mathbf{y}, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] + \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}_8}^{2-}} \mu_2 \cdot [v(\mathbf{y}, \Pi_1) - v(\mathbf{x}_8, \Pi_1)]$ , where the summation over the set  $\Gamma_{\mathbf{x}_8}^{2-}$  is zero (as it is an empty set), and thus,  $\sum_{\mathbf{y} \in \Gamma_{\mathbf{x}_8}^{1-}} \mu_1 \cdot [v(\mathbf{y}, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] = \mu_1 \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_5, \Pi_1) - v(\mathbf{x}_8, \Pi_1)]$ . Therefore, for state  $\mathbf{x}_8$ , Equation (3.29a) reads as  $R(\Pi_1) = r_1 \cdot 2 \cdot \mu_1 + \mu_1 \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_5, \Pi_1) - v(\mathbf{x}_8, \Pi_1)]$ .

**Example 3.11** Consider again the network in Fig. 3.2 in state  $\mathbf{x} = \mathbf{x}_2$ . For class-1 and class-2 arrivals, we have explained in Example 3.6 all feasible state transitions when policies  $\Pi_1$  and  $\Pi_2$  allocate resources in this state (see Fig. 3.5). In Fig. 3.10, we show the state transition diagrams, the transition rates and probabilities which are defined by these policies when the network is in state  $\mathbf{x}_2$ . Furthermore, Fig. 3.10

**Possible state transitions when the network uses policy  $\Pi_1$  and is in state  $\mathbf{x}_2$ :**



**Remark:**

- While in state  $\mathbf{x}_2$ , the network may only move to the states  $\mathbf{x}_1$ ,  $\mathbf{x}_7$  or  $\mathbf{x}_{18}$
- States  $\mathbf{x}_7$  and  $\mathbf{x}_{18}$  are reachable when a class-1 or a class-2 arrival occurs, respectively
- State  $\mathbf{x}_1$  is reached when the class-1 connection being carried in state  $\mathbf{x}_2$  departs
- Thus, for  $\Pi_1$ :  $\Gamma_{\mathbf{x}_2}^{1+} = \{\mathbf{x}_7\}$ ,  $\Gamma_{\mathbf{x}_2}^{2+} = \{\mathbf{x}_{18}\}$ ,  $\Gamma_{\mathbf{x}_2}^{1-} = \{\mathbf{x}_1\}$ ,  $\Gamma_{\mathbf{x}_2}^{2-} = \{\phi\}$

The transition rates are:

$$q_{\mathbf{x}_2\mathbf{x}_1}(\Pi_1) = \mu_1, q_{\mathbf{x}_2\mathbf{x}_7}(\Pi_1) = \lambda_1(\mathbf{x}_2, \Pi_1), q_{\mathbf{x}_2\mathbf{x}_{18}}(\Pi_1) = \lambda_2(\mathbf{x}_2, \Pi_1), \text{ and } q_{\mathbf{x}_2\mathbf{y}}(\Pi_1) = 0 \text{ for } \mathbf{y} \neq \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_{18}$$

The sojourn time in state  $\mathbf{x}_2$  is:

$$\tau_{\mathbf{x}_2} = \frac{1}{\lambda_1(\mathbf{x}_2, \Pi_1) + \lambda_2(\mathbf{x}_2, \Pi_1) + \mu_1}$$

The transition probabilities are given by  $P_{\mathbf{x}_2\mathbf{y}}(\Pi_1) = q_{\mathbf{x}_2\mathbf{y}}(\Pi_1) \cdot \tau_{\mathbf{x}_2}$ , i.e.

$$P_{\mathbf{x}_2\mathbf{x}_1}(\Pi_1) = \mu_1 \cdot \tau_{\mathbf{x}_2}, P_{\mathbf{x}_2\mathbf{x}_7}(\Pi_1) = \lambda_1(\mathbf{x}_2, \Pi_1) \cdot \tau_{\mathbf{x}_2}, P_{\mathbf{x}_2\mathbf{x}_{18}}(\Pi_1) = \lambda_2(\mathbf{x}_2, \Pi_1) \cdot \tau_{\mathbf{x}_2}, \text{ and } P_{\mathbf{x}_2\mathbf{y}}(\Pi_1) = 0 \text{ for } \mathbf{y} \neq \mathbf{x}_1, \mathbf{x}_7, \mathbf{x}_{18}$$

which fulfil:

$$\sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}_2} P_{\mathbf{x}_2\mathbf{y}}(\Pi_1) = P_{\mathbf{x}_2\mathbf{x}_1}(\Pi_1) + P_{\mathbf{x}_2\mathbf{x}_7}(\Pi_1) + P_{\mathbf{x}_2\mathbf{x}_{18}}(\Pi_1) = \mu_1 \cdot \tau_{\mathbf{x}_2} + \lambda_1(\mathbf{x}_2, \Pi_1) \cdot \tau_{\mathbf{x}_2} + \lambda_2(\mathbf{x}_2, \Pi_1) \cdot \tau_{\mathbf{x}_2} = 1$$

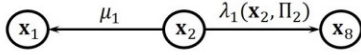
The network earns reward at a rate:

$$R(\Pi_1) = r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_2, \Pi_1) \cdot [v(\mathbf{x}_7, \Pi_1) - v(\mathbf{x}_2, \Pi_1)] + \lambda_2(\mathbf{x}_2, \Pi_1) \cdot [v(\mathbf{x}_{18}, \Pi_1) - v(\mathbf{x}_2, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_2, \Pi_1)]$$

If at  $t_0$  the network is in state  $\mathbf{x}_2$ , the expected reward earned at time  $t \gg t_0$  is:

$$V(\mathbf{x}_2, \Pi_1, t) = R(\Pi_1) \cdot t + v(\mathbf{x}_2, \Pi_1)$$

**Possible state transitions when the network uses policy  $\Pi_2$  and is in state  $\mathbf{x}_2$ :**



**Remark:**

- While in state  $\mathbf{x}_2$ , the network may only move to the states  $\mathbf{x}_1$  or  $\mathbf{x}_8$
- State  $\mathbf{x}_8$  is reachable when a class-1 arrival occurs
- State  $\mathbf{x}_1$  is reached when the class-1 connection being carried in state  $\mathbf{x}_2$  departs
- Thus, for  $\Pi_2$ :  $\Gamma_{\mathbf{x}_2}^{1+} = \{\mathbf{x}_8\}$ ,  $\Gamma_{\mathbf{x}_2}^{2+} = \{\phi\}$ ,  $\Gamma_{\mathbf{x}_2}^{1-} = \{\mathbf{x}_1\}$ ,  $\Gamma_{\mathbf{x}_2}^{2-} = \{\phi\}$

The transition rates are:

$$q_{\mathbf{x}_2\mathbf{x}_1}(\Pi_2) = \mu_1, q_{\mathbf{x}_2\mathbf{x}_8}(\Pi_2) = \lambda_1(\mathbf{x}_2, \Pi_2), \text{ and } q_{\mathbf{x}_2\mathbf{y}}(\Pi_2) = 0 \text{ for } \mathbf{y} \neq \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_8$$

The sojourn time in state  $\mathbf{x}_2$  is:

$$\tau_{\mathbf{x}_2} = \frac{1}{\lambda_1(\mathbf{x}_2, \Pi_2) + \mu_1}$$

The transition probabilities are given by  $P_{\mathbf{x}_2\mathbf{y}}(\Pi_2) = q_{\mathbf{x}_2\mathbf{y}}(\Pi_2) \cdot \tau_{\mathbf{x}_2}$ , i.e.

$$P_{\mathbf{x}_2\mathbf{x}_1}(\Pi_2) = \mu_1 \cdot \tau_{\mathbf{x}_2}, P_{\mathbf{x}_2\mathbf{x}_8}(\Pi_2) = \lambda_1(\mathbf{x}_2, \Pi_2) \cdot \tau_{\mathbf{x}_2}, \text{ and } P_{\mathbf{x}_2\mathbf{y}}(\Pi_2) = 0 \text{ for } \mathbf{y} \neq \mathbf{x}_1, \mathbf{x}_8$$

which fulfil:

$$\sum_{\mathbf{y}: \mathbf{y} \neq \mathbf{x}_2} P_{\mathbf{x}_2\mathbf{y}}(\Pi_2) = P_{\mathbf{x}_2\mathbf{x}_1}(\Pi_2) + P_{\mathbf{x}_2\mathbf{x}_8}(\Pi_2) = \mu_1 \cdot \tau_{\mathbf{x}_2} + \lambda_1(\mathbf{x}_2, \Pi_2) \cdot \tau_{\mathbf{x}_2} = 1$$

The network earns reward at a rate:

$$R(\Pi_2) = r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_2, \Pi_2) \cdot [v(\mathbf{x}_8, \Pi_2) - v(\mathbf{x}_2, \Pi_2)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_2) - v(\mathbf{x}_2, \Pi_2)]$$

If at  $t_0$  the network is in state  $\mathbf{x}_2$ , the expected reward earned at time  $t \gg t_0$  is:

$$V(\mathbf{x}_2, \Pi_2, t) = R(\Pi_2) \cdot t + v(\mathbf{x}_2, \Pi_2)$$

**Figure 3.10:** Calculation of the network reward rates  $R(\Pi_1)$  and  $R(\Pi_2)$  in Example 3.11.

defines Equation (3.29a) for  $R(\Pi_1)$  and  $R(\Pi_2)$ . When the network uses policy  $\Pi_1$ , a class-1 and a class-2 arrival cause the transitions  $\mathbf{x}_2 \rightarrow \mathbf{x}_7$  and  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$ , respectively. Therefore, for  $\mathbf{x}_2$ , we have that  $\Gamma_{\mathbf{x}_2}^{1+} = \{\mathbf{x}_7\}$  and  $\Gamma_{\mathbf{x}_2}^{2+} = \{\mathbf{x}_{18}\}$ . (Notice that these transitions are defined by the policy decisions in Fig. 3.4.) However, while in state  $\mathbf{x}_2$ , the network is carrying a single class-1 connection, thereby defining another possible transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_1$ , which takes place when the connection is terminated. Thus  $\Gamma_{\mathbf{x}_2}^{1-} = \{\mathbf{x}_1\}$  and  $\Gamma_{\mathbf{x}_2}^{2-} = \{\emptyset\}$ . Then we have that  $n_1(\mathbf{x}_2) = 1$  and  $n_2(\mathbf{x}_2) = 0$ , which implies:  $\mu_1(\mathbf{x}_2) = \mu_1$ ,  $\mu_2(\mathbf{x}_2) = 0$ , and therefore, the reward rate in state  $\mathbf{x}_2$  is  $q(\mathbf{x}_2) = r_1 \cdot \mu_1$ . As a result, from state  $\mathbf{x}_2$ , and provided that policy  $\Pi_1$  is used, only three state transitions are possible. For  $R(\Pi_1)$ , these transitions define the Equation (3.29a) as seen in Fig. 3.10. The same analysis can be followed to interpret the case for policy  $\Pi_2$  depicted in the figure. Suffice it to say that for this policy, class-2 arrivals are rejected, and thereby, only two transitions define  $R(\Pi_2)$ , namely  $\mathbf{x}_2 \rightarrow \mathbf{x}_8$ , due to a class-1 admission, and  $\mathbf{x}_2 \rightarrow \mathbf{x}_1$ , due to departure of the carried class-1 connection. This example shows that different resource allocation policies yield different network reward rates, i.e.  $R(\Pi_1) \neq R(\Pi_2)$ .

Equations (3.29) fully describe the dependence of the network reward rate  $R(\Pi)$  on each state  $\mathbf{x} \in \Omega_{\mathbf{x}}$ . For every  $\mathbf{x}$ , Equation (3.29a) has the following properties:

1. The equation is linear in the transient reward values  $v(\mathbf{x}, \Pi)$  and  $v(\mathbf{y}, \Pi)$ . These values together with  $R(\Pi)$  are the variables or unknowns of the equation. Observe that the values  $v(\mathbf{y}, \Pi)$  are the transient reward values of the network states  $\mathbf{y}$  which are reachable from transitions  $\mathbf{x} \rightarrow \mathbf{y}$ . These transitions originate from connection arrivals or departures.
2. The parameters  $q(\mathbf{x})$ ,  $\lambda_j(\mathbf{x}, \Pi)$  and  $\mu_j$  are known. By knowing  $\mathbf{x}$ , it follows that  $n_j(\mathbf{x})$ ,  $\mu_j(\mathbf{x})$  and  $q(\mathbf{x})$  are known as well. The rates  $\lambda_j(\mathbf{x}, \Pi)$  can be estimated from online measurements taken during network operation.
3. The right hand side of the equation shows that the network reward rate  $R(\Pi)$  depends on the differences  $v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$ . According to Equations (3.18) and (3.29e), these differences are interpreted as the reward change, with respect to  $V(\mathbf{x}, \Pi, t)$ , expected if instead of  $\mathbf{x}$ , the network was in state  $\mathbf{y}$  at  $t_0$ . Therefore, the reward  $V(\mathbf{y}, \Pi, t)$  at  $t \gg t_0$  is expected to differ in  $v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$  (ru) with respect to  $V(\mathbf{x}, \Pi, t)$ . As it will be shown later, this property forms the basis for the implementation of an optimization algorithm that finds the policy that maximizes the network reward rate  $R(\Pi)$ .

Since each  $\mathbf{x} \in \Omega_{\mathbf{x}}$  must fulfil Equation (3.29a), it follows that for a network that allocates resources with a policy  $\Pi$ , the reward rate  $R(\Pi)$  is a solution of the system of  $|\Omega_{\mathbf{x}}|$  linear equations defined by all states in  $\Omega_{\mathbf{x}}$ . Given that for all  $\mathbf{x}$  the parameters  $q(\mathbf{x})$ ,  $\lambda_j(\mathbf{x}, \Pi)$  and  $\mu_j$  are known, the linear system must be solved for  $R(\Pi)$  and the  $|\Omega_{\mathbf{x}}|$  values  $v(\mathbf{x}, \Pi)$ . Therefore, Equation (3.29a) implicitly defines a system of  $|\Omega_{\mathbf{x}}|$  linear equations with  $|\Omega_{\mathbf{x}}| + 1$  unknowns. A unique solution to the linear system can be calculated by arbitrarily setting one the  $|\Omega_{\mathbf{x}}|$  transient values  $v(\mathbf{x}, \Pi)$  to zero. The resulting system of  $|\Omega_{\mathbf{x}}|$  equations and  $|\Omega_{\mathbf{x}}|$  unknowns is then solved for  $R(\Pi)$  and the remaining  $|\Omega_{\mathbf{x}}| - 1$  values  $v(\mathbf{x}, \Pi)$ . This approach is valid as the third property of Equation (3.29a) emphasizes that  $R(\Pi)$  only depends on the differences  $v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$ . To prove this, let us add a constant  $k$  to all values  $v(\mathbf{x}, \Pi)$ . Then we have that the values change as:  $v'(\mathbf{y}, \Pi) = v(\mathbf{y}, \Pi) + k$  and  $v'(\mathbf{x}, \Pi) = v(\mathbf{x}, \Pi) + k$ , thereby Equation (3.29a) is now written (for a state  $\mathbf{x}$ ) as:  $R(\Pi) = q(\mathbf{x}) + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j+}} \lambda_j(\mathbf{x}, \Pi) \cdot [v'(\mathbf{y}, \Pi) - v'(\mathbf{x}, \Pi)] + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j-}} \mu_j \cdot [v'(\mathbf{y}, \Pi) - v'(\mathbf{x}, \Pi)]$ . However, note that  $v'(\mathbf{y}, \Pi) - v'(\mathbf{x}, \Pi) = v(\mathbf{y}, \Pi) + k - v(\mathbf{x}, \Pi) - k = v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$ . Therefore, adding the same constant to all values  $v(\mathbf{x}, \Pi)$  leaves Equation (3.29a) unchanged. In particular, if we define that constant as  $k = -v(\mathbf{x}_r, \Pi)$ , where  $v(\mathbf{x}_r, \Pi)$  is the transient value of an arbitrarily selected state  $\mathbf{x}_r \in \Omega_{\mathbf{x}}$ , we have that  $v'(\mathbf{y}, \Pi) = v(\mathbf{y}, \Pi) - v(\mathbf{x}_r, \Pi)$ , and for all  $\mathbf{x} \neq \mathbf{x}_r$ :  $v'(\mathbf{x}, \Pi) = v(\mathbf{x}, \Pi) - v(\mathbf{x}_r, \Pi)$ . Furthermore,  $v'(\mathbf{x}_r, \Pi) = v(\mathbf{x}_r, \Pi) - v(\mathbf{x}_r, \Pi) = 0$ . Hence, the statement “the linear system is solved by arbitrarily setting one the transient values to zero” implies that in Equation (3.29), for an arbitrarily chosen state  $\mathbf{x}_r$ , we set  $v'(\mathbf{x}_r, \Pi) = 0$ , and then, the remaining  $|\Omega_{\mathbf{x}}| - 1$  values are of the form  $v'(\mathbf{x}, \Pi) = v(\mathbf{x}, \Pi) - v(\mathbf{x}_r, \Pi)$ . This renders the linear system solvable with a solution that is unique, and the same, regardless of the chosen state  $\mathbf{x}_r$ , as the differences  $v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$  are independent of  $v(\mathbf{x}_r, \Pi)$ . This result has been proved in [How60] as well, and algorithms proposed to solve this type of linear system can be found in [Tij86].

**Example 3.12** Fig. 3.11 shows the linear system defined by Equation (3.29a) for the network in Fig. 3.2 when policy  $\Pi_1$  is used. For this network  $|\Omega_{\mathbf{x}}| = 18$ , and therefore, by arbitrarily setting one of the 18 values to zero, the system is solved for  $R(\Pi_1)$  and the remaining 17 values  $v(\mathbf{x}_i, \Pi_1)$ .

$$\begin{aligned}
\mathbf{x}_1: R(\Pi_1) &= \lambda_1(\mathbf{x}_1, \Pi_1) \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_1, \Pi_1)] + \lambda_2(\mathbf{x}_1, \Pi_1) \cdot [v(\mathbf{x}_{14}, \Pi_1) - v(\mathbf{x}_1, \Pi_1)] \\
\mathbf{x}_2: R(\Pi_1) &= r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_2, \Pi_1) \cdot [v(\mathbf{x}_7, \Pi_1) - v(\mathbf{x}_2, \Pi_1)] + \lambda_2(\mathbf{x}_2, \Pi_1) \cdot [v(\mathbf{x}_{18}, \Pi_1) - v(\mathbf{x}_2, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_2, \Pi_1)] \\
\mathbf{x}_3: R(\Pi_1) &= r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_3, \Pi_1) \cdot [v(\mathbf{x}_{10}, \Pi_1) - v(\mathbf{x}_3, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_3, \Pi_1)] \\
\mathbf{x}_4: R(\Pi_1) &= r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_4, \Pi_1) \cdot [v(\mathbf{x}_7, \Pi_1) - v(\mathbf{x}_4, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_4, \Pi_1)] \\
\mathbf{x}_5: R(\Pi_1) &= r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_5, \Pi_1) \cdot [v(\mathbf{x}_8, \Pi_1) - v(\mathbf{x}_5, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_5, \Pi_1)] \\
\mathbf{x}_6: R(\Pi_1) &= r_1 \cdot \mu_1 + \lambda_1(\mathbf{x}_6, \Pi_1) \cdot [v(\mathbf{x}_9, \Pi_1) - v(\mathbf{x}_6, \Pi_1)] + \lambda_2(\mathbf{x}_6, \Pi_1) \cdot [v(\mathbf{x}_{17}, \Pi_1) - v(\mathbf{x}_6, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_6, \Pi_1)] \\
\mathbf{x}_7: R(\Pi_1) &= r_1 \cdot 2\mu_1 + \lambda_1(\mathbf{x}_7, \Pi_1) \cdot [v(\mathbf{x}_{13}, \Pi_1) - v(\mathbf{x}_7, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_4, \Pi_1) - v(\mathbf{x}_7, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_7, \Pi_1)] \\
\mathbf{x}_8: R(\Pi_1) &= r_1 \cdot 2\mu_1 + \mu_1 \cdot [v(\mathbf{x}_5, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_8, \Pi_1)] \\
\mathbf{x}_9: R(\Pi_1) &= r_1 \cdot 2\mu_1 + \lambda_1(\mathbf{x}_9, \Pi_1) \cdot [v(\mathbf{x}_{13}, \Pi_1) - v(\mathbf{x}_9, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_6, \Pi_1) - v(\mathbf{x}_9, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_9, \Pi_1)] \\
\mathbf{x}_{10}: R(\Pi_1) &= r_1 \cdot 2\mu_1 + \mu_1 \cdot [v(\mathbf{x}_5, \Pi_1) - v(\mathbf{x}_{10}, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_3, \Pi_1) - v(\mathbf{x}_{10}, \Pi_1)] \\
\mathbf{x}_{11}: R(\Pi_1) &= r_1 \cdot 2\mu_1 + \mu_1 \cdot [v(\mathbf{x}_6, \Pi_1) - v(\mathbf{x}_{11}, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_3, \Pi_1) - v(\mathbf{x}_{11}, \Pi_1)] \\
\mathbf{x}_{12}: R(\Pi_1) &= r_1 \cdot 2\mu_1 + \lambda_1(\mathbf{x}_{12}, \Pi_1) \cdot [v(\mathbf{x}_{13}, \Pi_1) - v(\mathbf{x}_{12}, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_6, \Pi_1) - v(\mathbf{x}_{12}, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_4, \Pi_1) - v(\mathbf{x}_{12}, \Pi_1)] \\
\mathbf{x}_{13}: R(\Pi_1) &= r_1 \cdot 3\mu_1 + \mu_1 \cdot [v(\mathbf{x}_{12}, \Pi_1) - v(\mathbf{x}_{13}, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_9, \Pi_1) - v(\mathbf{x}_{13}, \Pi_1)] + \mu_1 \cdot [v(\mathbf{x}_7, \Pi_1) - v(\mathbf{x}_{13}, \Pi_1)] \\
\mathbf{x}_{14}: R(\Pi_1) &= r_2 \cdot \mu_2 + \lambda_1(\mathbf{x}_{14}, \Pi_1) \cdot [v(\mathbf{x}_{17}, \Pi_1) - v(\mathbf{x}_{14}, \Pi_1)] + \mu_2 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_{14}, \Pi_1)] \\
\mathbf{x}_{15}: R(\Pi_1) &= r_2 \cdot \mu_2 + \mu_2 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_{15}, \Pi_1)] \\
\mathbf{x}_{16}: R(\Pi_1) &= r_2 \cdot \mu_2 + \lambda_1(\mathbf{x}_{16}, \Pi_1) \cdot [v(\mathbf{x}_{18}, \Pi_1) - v(\mathbf{x}_{16}, \Pi_1)] + \mu_2 \cdot [v(\mathbf{x}_1, \Pi_1) - v(\mathbf{x}_{16}, \Pi_1)] \\
\mathbf{x}_{17}: R(\Pi_1) &= r_1 \cdot \mu_1 + r_2 \cdot \mu_2 + \mu_1 \cdot [v(\mathbf{x}_{14}, \Pi_1) - v(\mathbf{x}_{17}, \Pi_1)] + \mu_2 \cdot [v(\mathbf{x}_6, \Pi_1) - v(\mathbf{x}_{17}, \Pi_1)] \\
\mathbf{x}_{18}: R(\Pi_1) &= r_1 \cdot \mu_1 + r_2 \cdot \mu_2 + \mu_1 \cdot [v(\mathbf{x}_{16}, \Pi_1) - v(\mathbf{x}_{18}, \Pi_1)] + \mu_2 \cdot [v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_{18}, \Pi_1)]
\end{aligned}$$

**Figure 3.11:** Linear system for the network in Fig. 3.2 when policy  $\Pi_1$  is used.

### 3.4. Network Reward Maximization as a Markov Decision Process

By modelling the problem of online resource allocation as a Markov decision process, we can use the linear system defined by Equation (3.29a) to calculate an optimum policy  $\Pi = \Pi^*$  that maximizes the network reward rate  $R$ , i.e. Equation (3.5). This is achieved by the policy iteration algorithm (PIA) proposed in [How60], which is applicable to systems that behave as Markov stochastic processes. In Fig. 3.12, we present an online version of the PIA for the case of resource allocation in dynamic optical networks. By initializing the network with an arbitrarily defined policy  $\Pi_0$ , the PIA performs an iteration cycle every  $\Delta T$  time units. An iteration consists of two steps, namely, the value determination operation (VDO) and the policy improvement routine (PIR). Both steps work together to calculate a policy which has a reward rate higher than that obtained by the policy found in the previous iteration. The algorithm stops when the policies calculated in two successive iterations have the same performance, i.e. the same reward rate  $R$ . The validity of this stopping criterion and of the fact that the optimum policy is always found (regardless of the policy that initializes the PIA) is the focus of the work expounded in [How60]. Therein it is also proved that before finding the optimum policy  $\Pi^*$ , an iteration always yields a policy that outperforms all previously calculated policies. Based on these results, we propose the algorithm in Fig. 3.12 which will be explained in the following.

#### 3.4.1 The Value Determination Operation (VDO)

For a network allocating resources with a policy  $\Pi$ , the value determination operation is the calculation of the solution to the linear system defined by Equation (3.29a). The solution consists of the reward rate  $R(\Pi)$  and the values  $v(\mathbf{x}, \Pi)$ . Thus, the goal of the VDO is to calculate the performance of the policy  $\Pi$ .

Consider an optical network that initially uses an arbitrarily defined policy  $\Pi = \Pi_0$  - see Fig. 3.12. (In the flowchart depicted in the figure, we refer to an event as either the arrival or departure of a connection.) Within the first iteration cycle of length  $\Delta T$ , the network uses this policy to allocate resources for any connection request arriving in the interval  $\Delta T$ . At the end of this interval, the VDO step is executed so as to calculate  $R(\Pi_0)$  and the values  $v(\mathbf{x}, \Pi_0)$ . For this, the arrival rates  $\lambda_j(\mathbf{x}, \Pi_0)$  are estimated from measurements taken during the cycle period  $\Delta T$ . The known parameters  $q(\mathbf{x})$ ,  $\mu_j$  and  $\lambda_j(\mathbf{x}, \Pi_0)$  are used to solve the linear system. The solution is then used as input to the second step of the PIA, i.e. the policy

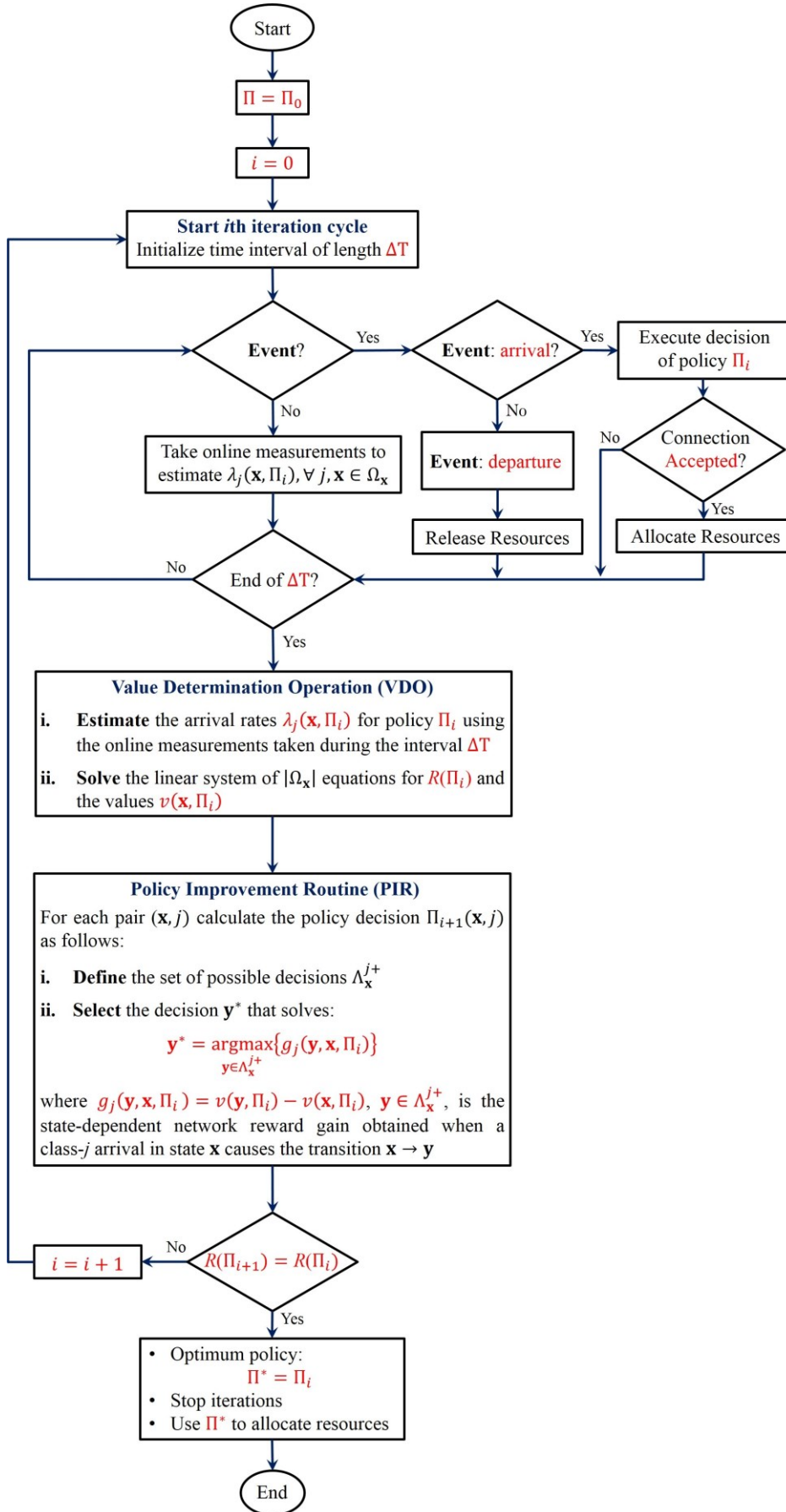
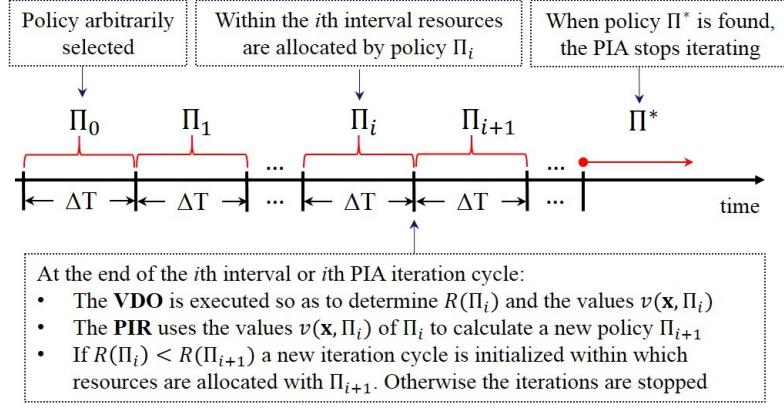


Figure 3.12: Flow chart for the policy iteration algorithm (PIA).



**Figure 3.13:** Temporal diagram for the policy iteration algorithm (PIA).

improvement routine, whose purpose is to calculate a policy  $\Pi_1$  that outperforms  $\Pi_0$ . Having determined  $\Pi_1$ , the network enters a new iteration cycle of length  $\Delta T$  within which resources are allocated based on the decisions defined in  $\Pi_1$ . In this cycle, online measurements are taken to estimate  $\lambda_j(\mathbf{x}, \Pi_1)$ . These measurements are used (at the end of the cycle) to execute the VDO step to estimate  $R(\Pi_1)$  and the values  $v(\mathbf{x}, \Pi_1)$  for  $\Pi_1$ . With this information, the PIR calculates a policy  $\Pi_2$  that improves  $\Pi_1$ , and a new iteration cycle is then entered. This process is repeated so as to calculate a sequence of policies  $\Pi_1, \Pi_2, \dots, \Pi_i, \dots, \Pi^*$ , which result, respectively, in reward rates  $R(\Pi_0) < R(\Pi_1) < \dots < R(\Pi_i) < \dots \leq R(\Pi^*)$ . The iterations are stopped when the policies obtained in two consecutive cycles attain the same network reward rate. This process is summarized in the diagram shown in Fig. 3.13.

Cycle periods with length  $\Delta T$  are used in order to measure the impact that the policy in use has on the network performance. This is accomplished by the measurements taken to estimate the rates  $\lambda_j(\mathbf{x}, \Pi)$ . The estimation of this rates has also the advantage that they reflect changes in the exogenous per-class arrival rates  $\lambda_j$ , thereby the PIA adapts the policy calculation to varying traffic loads. Recall that any policy  $\Pi$  determines the transitions  $\mathbf{x} \rightarrow \mathbf{y}$  caused by arrivals in state  $\mathbf{x}$ . Therefore, the estimation from online measurements (at the end of every cycle) of the rates  $\lambda_j(\mathbf{x}, \Pi)$ , implies the calculation of the transition rates  $\lambda_j(\mathbf{x}, \Pi) = q_{xy}(\Pi)$  that define the probabilities  $P_{xy}(\Pi)$  implicitly involved in Equation (3.29a). Without an online estimation of  $\lambda_j(\mathbf{x}, \Pi)$ , the VDO step would be unable to solve the linear system, as no information would be available about the policy performance. We defer to Chapter 4 the discussion on how the measurements are defined and how they are employed to estimate the transition rates  $\lambda_j(\mathbf{x}, \Pi)$ .

### 3.4.2 The Policy Improvement Routine (PIR)

Assume that in the  $i$ th iteration cycle the network executes a policy  $\Pi_i$ . At the end of the cycle, the VDO calculates the rate  $R(\Pi_i)$  and the values  $v(\mathbf{x}, \Pi_i)$  that characterize the policy. With that information, the PIR in Fig. 3.12 is executed so as to calculate a policy  $\Pi_{i+1}$ , such that  $R(\Pi_i) \leq R(\Pi_{i+1})$ . This calculation involves, for each pair  $(\mathbf{x}, j)$ , the determination of the matrix element  $\Pi_{i+1}(\mathbf{x}, j)$  that improves  $\Pi_i(\mathbf{x}, j)$ . This is accomplished by maximizing the right hand side of Equation (3.29a) with the transient values  $v(\mathbf{x}, \Pi_i)$  calculated in the VDO step. For a matrix element  $\Pi_{i+1}(\mathbf{x}, j)$ , we phrase the optimization problem as follows: for a class- $j$  connection request arriving in state  $\mathbf{x}$ , select from the set of possible decisions that for which  $R(\Pi_{i+1}(\mathbf{x}, j)) \geq R(\Pi_i(\mathbf{x}, j))$ . We denote the set of possible decisions as  $\Lambda_{\mathbf{x}}^{j+}$ , where an element in  $\Lambda_{\mathbf{x}}^{j+}$  is a state in  $\Omega_{\mathbf{x}}$ , and therefore,  $\Lambda_{\mathbf{x}}^{j+} \subseteq \Omega_{\mathbf{x}}$ .

Before calculating the decision  $\Pi_{i+1}(\mathbf{x}, j)$ , the PIR has to define the set  $\Lambda_{\mathbf{x}}^{j+}$ . This definition follows the procedure outlined in Section 3.2.2: an RSA algorithm is used to calculate the set of candidate lightpaths for a class- $j$  arrival in state  $\mathbf{x}$ , each lightpath represents a decision. In  $\Lambda_{\mathbf{x}}^{j+}$  a decision is then represented by the state that a candidate lightpath would configure if it is assigned to the request that arrives in state  $\mathbf{x}$ . Besides the candidate lightpaths,  $\Lambda_{\mathbf{x}}^{j+}$  contains the state  $\mathbf{x}$  as well. The reason is that the decision can be made that the class- $j$  request is rejected, and thus, no transition occurs.

**Example 3.13** For the network in Fig. 3.2, in Example 3.8 we have shown that class-1 arrivals in state  $\mathbf{x}_2$  have the four possible decisions summarized in Fig. 3.6. They define the set  $\Lambda_{\mathbf{x}_2}^{1+} = \{\mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$ , where if the decision  $\mathbf{x}_2$  is selected, then there is no state transition, meaning that the connection request is rejected. The remaining three states in  $\Lambda_{\mathbf{x}_2}^{1+}$  represent the states reachable when a lightpath is assigned to the connection request. For instance if  $\mathbf{x}_9$  is selected, then the transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_9$  occurs, which implies that the connection is admitted on a lightpath that seizes slots 5-6 on the path  $\rho = (A, B, C)$  (see “Decision 3” in Fig 3.6). Moreover, for class-2, we have that  $\Lambda_{\mathbf{x}_2}^{2+} = \{\mathbf{x}_2, \mathbf{x}_{18}\}$ , i.e. a class-2 arrival in state  $\mathbf{x}_2$  can

be either rejected (if  $\mathbf{x}_2$  is selected) or admitted on a lightpath that seizes slots 3-6 on the path  $\rho = (B, C)$ . In this case the transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$  takes place (see Fig. 3.5).

To solve the maximization problem for the pair  $(\mathbf{x}, j)$ , note that any definable policy provides decisions that solely cause state transitions which stem from connection arrivals, but not departures. This means that in Equation (3.29a), any possible decision in  $\Lambda_{\mathbf{x}}^{j+}$  solely influences the network reward rate through the double summation that represents changes due to arrivals. For a class- $j$  connection request arriving in state  $\mathbf{x}$ , the maximization problem is then defined from Equation (3.29a) as:

$$R(\Pi_{i+1}(\mathbf{x}, j)) = \max_{\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}} \{q(\mathbf{x}) + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j+}} \lambda_j(\mathbf{x}, \Pi_i) \cdot [v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i)] + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j-}} \mu_j \cdot [v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i)]\} \quad (3.30)$$

which means that from the set  $\Lambda_{\mathbf{x}}^{j+}$  we are interested in the decision  $\mathbf{y}^*$  that satisfies:

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}} \{\lambda_j(\mathbf{x}, \Pi_i) \cdot [v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i)]\} \quad (3.31)$$

i.e. Equation (3.30) is maximized by selecting the state (or equivalently the decision)  $\mathbf{y}^* \in \Lambda_{\mathbf{x}}^{j+}$  that maximizes the  $j$ th term in the double summation that represents reward changes due to arrivals. Since  $\lambda_j(\mathbf{x}, \Pi_i)$  does not depend on  $\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}$ , then we can further simplify Equation (3.31) as:

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}} \{g_j(\mathbf{y}, \mathbf{x}, \Pi_i)\} \quad (3.32)$$

where

$$g_j(\mathbf{y}, \mathbf{x}, \Pi_i) = v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i) \quad (3.33)$$

with

$$-\infty < g_j(\mathbf{y}, \mathbf{x}, \Pi_i) < +\infty \quad (3.34)$$

The difference  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i) = v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i)$  corresponds to Equation (3.29e), and we denote it as the state-dependent network reward gain. It is the long-term reward that a connection of class- $j$  brings if it is admitted in state  $\mathbf{x}$  causing the transition  $\mathbf{x} \rightarrow \mathbf{y}$ . To understand this, assume that at  $t_0$  the network is in state  $\mathbf{x}$  and a class- $j$  connection request arrives, gets admission, and seizes a lightpath that causes the transition  $\mathbf{x} \rightarrow \mathbf{y}$ . The admission of this connection has a short-term and a long-term effect on the reward earned by the network. The short-term effect is that during the connection holding time a reward of  $r_j$  (ru) is obtained. However, since at  $t_0$  the connection sets the network in state  $\mathbf{y}$ , then at  $t \gg t_0$ , i.e. as a long-term effect, the network is expected to earn a reward of  $V(\mathbf{y}, \Pi_i, t)$  (ru). From Equation (3.29e), this reward is given by  $V(\mathbf{y}, \Pi_i, t) = v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i) + V(\mathbf{x}, \Pi_i, t)$ , which is equal to  $V(\mathbf{y}, \Pi_i, t) = g_j(\mathbf{y}, \mathbf{x}, \Pi_i) + V(\mathbf{x}, \Pi_i, t)$ . Therefore, with respect to  $V(\mathbf{x}, \Pi_i, t)$ , i.e. the reward that would be expected at  $t \gg t_0$  if the connection had been rejected in state  $\mathbf{x}$ , the reward  $V(\mathbf{y}, \Pi_i, t)$  differs in  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$  (ru). If  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i) < 0$ , then at  $t \gg t_0$  the reward  $V(\mathbf{y}, \Pi_i, t)$  is expected to be  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$  (ru) less than  $V(\mathbf{x}, \Pi_i, t)$ , otherwise (i.e. positive gain) an increment  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$  is expected over  $V(\mathbf{x}, \Pi_i, t)$ . Then,  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$  is the long-term reward that a class- $j$  connection brings if it gets admission in state  $\mathbf{x}$ . Ideally, it would be desirable that  $r_j = g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$ , however, in most cases,  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i) \leq r_j$ . The reason is that for the network to earn a short-term reward  $r_j$ , it has to allocate to the connection  $b_j$  slots on a path selected from the set of routes  $\Gamma_j$ . These resources are then unavailable for connection requests arriving within the connection holding time  $\mu_j^{-1}$ , which might prevent the admission of connections with higher rewards. Therefore, the gain  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$  takes into account the future reward losses (i.e. the long-term effect) that the connection has on the reward earned by the network. A negative gain then means that the connection (in the long-term) decreases the reward by  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$  (ru) owing to the losses it causes; otherwise, a reward increment is expected. In Equation (3.32), the optimum decision  $\Pi_{i+1}(\mathbf{x}, j)$  is then given by the state  $\mathbf{y}^* \in \Lambda_{\mathbf{x}}^{j+}$  that yields the maximum long-term reward. The PIR calculates the policy  $\Pi_{i+1}$  by solving Equation (3.32) for all  $\mathbf{x} \in \Omega_{\mathbf{x}}, j = 1, 2, \dots, J$ .



$v(\mathbf{x}, \Pi_1), R(\Pi_1)$	Total Offered Traffic A (Erlangs)			
	0.01	0.1	1.0	10
$v(\mathbf{x}_1, \Pi_1)$	0.000	0.000	0.000	0.000
$v(\mathbf{x}_2, \Pi_1)$	2.500	2.500	2.472	2.243
$v(\mathbf{x}_3, \Pi_1)$	2.496	2.466	2.301	2.335
$v(\mathbf{x}_4, \Pi_1)$	2.496	2.467	2.318	2.840
$v(\mathbf{x}_5, \Pi_1)$	2.496	2.467	2.308	2.313
$v(\mathbf{x}_6, \Pi_1)$	2.500	2.500	2.478	2.152
$v(\mathbf{x}_7, \Pi_1)$	4.996	4.966	4.799	5.050
$v(\mathbf{x}_8, \Pi_1)$	4.995	4.955	4.686	3.943
$v(\mathbf{x}_9, \Pi_1)$	4.998	4.983	4.876	4.813
$v(\mathbf{x}_{10}, \Pi_1)$	4.993	4.938	4.601	3.989
$v(\mathbf{x}_{11}, \Pi_1)$	4.995	4.955	4.685	3.909
$v(\mathbf{x}_{12}, \Pi_1)$	4.996	4.966	4.802	5.018
$v(\mathbf{x}_{13}, \Pi_1)$	7.495	7.453	7.190	6.904
$v(\mathbf{x}_{14}, \Pi_1)$	3.964	3.665	2.017	-0.407
$v(\mathbf{x}_{15}, \Pi_1)$	3.941	3.440	-0.080	-12.693
$v(\mathbf{x}_{16}, \Pi_1)$	3.964	3.665	2.017	-0.366
$v(\mathbf{x}_{17}, \Pi_1)$	6.462	6.145	4.324	0.945
$v(\mathbf{x}_{18}, \Pi_1)$	6.462	6.144	4.323	0.990
$R(\Pi_1)$	0.0059	0.0560	0.4080	1.6693

**Table 3.3:** Solution to the linear system in Fig. 3.11 for different traffic loads.

**Example 3.14** Assume that in a cycle period  $\Delta T$ , the network in Fig. 3.2 allocates resources with the policy  $\Pi_1$  that is defined in Fig. 3.4. In Table 3.3, we present the transient reward values  $v(\mathbf{x}, \Pi_1)$  and the rate  $R(\Pi_1)$  calculated by the VDO under different offered traffic loads  $A$ . This offered traffic is given by  $A = \lambda_1/\mu_1 + \lambda_2/\mu_2$ , where  $\mu_1 = 1.0$ ,  $\mu_2 = 0.1$  and  $\lambda_1 = \lambda_2 = \lambda$  (con/uoT). (Note that for a given traffic load  $A$ , we have that  $\lambda = A/(\mu_1^{-1} + \mu_2^{-1})$ . For example, the column  $A = 0.1$  in Table 3.3 is obtained when the exogenous arrival rates are  $\lambda_1 = \lambda_2 = 0.009$  (con/uoT).) Furthermore, we define  $r_1 = 2.5$  (ru) and  $r_2 = 4.0$  (ru). With that, for each traffic load, the 18 values and the rate  $R(\Pi_1)$  are obtained by solving linear system defined in Fig. 3.11, where we set  $v(\mathbf{x}_1, \Pi_1) = 0$  (ru), with the state-dependent rates defined as  $\lambda_1(\mathbf{x}, \Pi_1) = \lambda_1$  and  $\lambda_2(\mathbf{x}, \Pi_1) = \lambda_2$ . The results in Table 3.3 show that the policy performance depends on the traffic load offered to the network during the cycle period  $\Delta T$ . That load defines the rates  $\lambda_j(\mathbf{x}, \Pi_1)$  used by the PIA to solve the linear system. This illustrates the importance of knowing these rates as they quantify the performance attainable by a policy under actual traffic conditions.

**Example 3.15** Let us use the values  $v(\mathbf{x}, \Pi_1)$  in Table 3.3 to execute the PIR to calculate the decisions for class-1 and class-2 requests arriving in state  $\mathbf{x}_2$ . In Example 3.13 we defined the sets of decisions  $\Lambda_{\mathbf{x}_2}^{1+}$  and  $\Lambda_{\mathbf{x}_2}^{2+}$ . In the PIR, these sets define the network reward gains (or long-term rewards) shown in Table 3.4 (for class-1 and class-2, there are four gains  $g_1(\mathbf{y}, \mathbf{x}_2, \Pi_1)$  and two gains  $g_2(\mathbf{y}, \mathbf{x}_2, \Pi_1)$ , respectively). By solving Equation (3.32), for class-1, the best decision is the state  $\mathbf{y}^*$  in  $\Lambda_{\mathbf{x}_2}^{1+} = \{\mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$  for which  $g_1(\mathbf{y}^*, \mathbf{x}_2, \Pi_1) = \max\{g_1(\mathbf{x}_2, \mathbf{x}_2, \Pi_1), g_1(\mathbf{x}_7, \mathbf{x}_2, \Pi_1), g_1(\mathbf{x}_8, \mathbf{x}_2, \Pi_1), g_1(\mathbf{x}_9, \mathbf{x}_2, \Pi_1)\}$ . In a similar way, for class-2 connections, the best decision is defined by the network state  $\mathbf{y}^*$  in  $\Lambda_{\mathbf{x}_2}^{2+} = \{\mathbf{x}_2, \mathbf{x}_{18}\}$  for which  $g_2(\mathbf{y}^*, \mathbf{x}_2, \Pi_1) = \max\{g_2(\mathbf{x}_2, \mathbf{x}_2, \Pi_1), g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1)\}$ . Table 3.5 summarizes the solutions to this maximization problem for each traffic load, and in Fig 3.14 we show the lightpaths that each decision defines. For example, consider the case  $A = 1.0$  Erlangs. The new policy decision  $\Pi(\mathbf{x}_2, 1)$  instructs the network that class-1 requests arriving in state  $\mathbf{x}_2$  must be admitted and routed on the path  $\rho = (A, B, C)$ , seizing the slots 5-6 (see Fig. 3.14). This decision, when executed, causes the state transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_9$ , thereby yielding a gain of  $g_1(\mathbf{x}_9, \mathbf{x}_2, \Pi_1) = 2.404$  (ru) as seen in Table 3.5. An interesting case occurs with the decision calculated for class-2 when  $A = 10$  Erlangs. Note that in Table 3.4,  $g_2(\mathbf{y}^*, \mathbf{x}_2, \Pi_1) = \max\{g_2(\mathbf{x}_2, \mathbf{x}_2, \Pi_1) = 0, g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = -1.253\} = 0$ , and hence, although resources are available, the connection request must be rejected. The reason is that, if admitted, the connection would bring a reward loss of  $g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = -1.253$  (ru).

**Example 3.16** For low traffic loads, i.e.  $A = 0.01$  and  $A = 0.1$ , the gains in Table 3.4 tend to equal the reward parameters  $r_j$ , i.e.  $r_j \approx g_j(\mathbf{y}, \mathbf{x}_2, \Pi_1)$ , as the network is not overloaded, and thus, the admission of a connection does not likely prevent the network from accepting more ‘‘profitable’’ traffic. (Recall that, as defined in Example 3.14,  $r_1 = 2.5$  and  $r_2 = 4.0$ .) However, as the offered load increases, i.e.  $A = 1.0$

Network Reward Gain (ru)	Total Offered Traffic A (Erlangs)			
	0.01	0.1	1.0	10
$g_1(\mathbf{x}_2, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$	0.000	0.000	0.000	0.000
$g_1(\mathbf{x}_7, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_7, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$	2.496	2.466	2.327	2.807
$g_1(\mathbf{x}_8, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_8, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$	2.495	2.455	2.214	1.700
$g_1(\mathbf{x}_9, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_9, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$	2.498	2.483	2.404	2.570
$g_2(\mathbf{x}_2, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_2, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$	0.000	0.000	0.000	0.000
$g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_{18}, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$	3.962	3.644	1.851	-1.253

**Table 3.4:** Network reward gains used to calculate a decision for class-1/class-2 arrivals in state  $\mathbf{x}_2$ .

Optimum Decision		Total Offered Traffic A (Erlangs)			
		0.01	0.1	1.0	10
Class-1	Decision $\mathbf{y}^*$	$\mathbf{x}_9$	$\mathbf{x}_9$	$\mathbf{x}_9$	$\mathbf{x}_7$
	Gain $g_1(\mathbf{y}^*, \mathbf{x}_2, \Pi_1)$	2.498	2.483	2.404	2.807
Class-2	Decision $\mathbf{y}^*$	$\mathbf{x}_{18}$	$\mathbf{x}_{18}$	$\mathbf{x}_{18}$	$\mathbf{x}_2$
	Gain $g_2(\mathbf{y}^*, \mathbf{x}_2, \Pi_1)$	3.962	3.644	1.851	0.000

**Table 3.5:** Decisions calculated by the PIR for class-1 and class-2 arrivals in state  $\mathbf{x}_2$ .

- Decisions  $\Pi_1(\mathbf{x}_2, 1)$  and  $\Pi_1(\mathbf{x}_2, 2)$  defined in policy  $\Pi_1$ :

Policy $\Pi = \Pi_1$						
State $\mathbf{x}$	Class- $j$					
	1			2		
	Admit	Path	Slots	Admit	Path	Slots
$\mathbf{x}_2$	Yes	(A,B,C)	3-4	Yes	(B,C)	3-6

- After executing the PIR step, a new policy  $\Pi$  is obtained. For each traffic load, the new decisions  $\Pi(\mathbf{x}_2, 1)$  and  $\Pi(\mathbf{x}_2, 2)$  are defined as:

A (Erlangs)	State $\mathbf{x}$	Class- $j$					
		1			2		
		Admit	Path	Slots	Admit	Path	Slots
0.01	$\mathbf{x}_2$	Yes	(A,B,C)	5-6	Yes	(A,B,C)	3-6
0.1	$\mathbf{x}_2$	Yes	(A,B,C)	5-6	Yes	(A,B,C)	3-6
1.0	$\mathbf{x}_2$	Yes	(A,B,C)	5-6	Yes	(A,B,C)	3-6
10.0	$\mathbf{x}_2$	Yes	(A,B,C)	3-4	No	-----	---

**Figure 3.14:** Lightpaths defined by the decisions in Table 3.5.

and  $A = 10$ , in some cases  $g_j(\mathbf{y}, \mathbf{x}_2, \Pi_1) < r_j$ , which means that connections admitted may cause reward losses. For example, consider the case  $A = 10$  Erlangs in Table 3.4. If a class-2 request gets admission in state  $\mathbf{x}_2$  at  $t_0$ , it causes the transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$ , and brings an immediate reward of  $r_2 = 4.0$  (ru) with a reward gain of  $g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = -1.253$  (ru). The long-term effect of this decision is that at  $t \gg t_0$  the expected reward earned by the network would be  $V(\mathbf{x}_{18}, \Pi, t) = g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) + V(\mathbf{x}_2, \Pi_1, t) = V(\mathbf{x}_2, \Pi_1, t) - 1.253$  (ru). This means that the connection brings a loss of  $-1.253$  (ru), and therefore, the best decision (as seen in Fig. 3.14) is to reject the connection.

### 3.4.3 Further Remarks on the Policy Iteration Algorithm

The PIR step implements the policy calculation procedure outlined in Section 3.2. First, the determination of the sets of decisions  $\Lambda_{\mathbf{x}}^{j+}$  involves the calculation of candidate lightpaths via an RSA algorithm. Second, connection admission control is implemented when Equation (3.32) is solved. The admission decision rule follows from the definition of the maximization problem: select the decision in  $\Lambda_{\mathbf{x}}^{j+}$  which yields the maximum gain  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$ . Note that this rule guarantees that  $R(\Pi_{i+1}(\mathbf{x}, j)) \geq R(\Pi_i(\mathbf{x}, j))$ . The reason is that the set  $\Lambda_{\mathbf{x}}^{j+}$  always contains the state implicitly defined by the decision  $\Pi_i(\mathbf{x}, j)$ , and therefore, in the worst case we have that  $\Pi_{i+1}(\mathbf{x}, j) = \Pi_i(\mathbf{x}, j)$ . The importance of calculating the values  $v(\mathbf{x}, \Pi_i)$  at the end of the  $i$ th iteration cycle is that they allow the PIR to estimate the long-term rewards  $g_j(\mathbf{y}, \mathbf{x}, \Pi_i)$ . With this information, the PIR may determine from the sets  $\Lambda_{\mathbf{x}}^{j+}$  the decisions in  $\Pi_{i+1}$  that outperform  $\Pi_i$ . Observe that  $\Pi_{i+1}(\mathbf{x}, j)$  outperforms  $\Pi_i(\mathbf{x}, j)$  if and only if,  $\Pi_{i+1}(\mathbf{x}, j)$  yields a long-term reward higher than  $\Pi_i(\mathbf{x}, j)$ .

**Example 3.17** For the network in Fig. 3.2, the policy decision  $\Pi_1(\mathbf{x}_2, 1)$  causes the transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_7$ . Note that  $\mathbf{x}_7$  is in the set  $\Lambda_{\mathbf{x}_2}^{1+} = \{\mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$ . If the PIA executes the PIR to solve Equation (3.32), it has to calculate  $g_1(\mathbf{x}_7, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_7, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$ , which is the long-term reward that quantifies the performance of the decision  $\Pi_1(\mathbf{x}_2, 1)$ . This decision would be selected again by the PIR, i.e.  $\Pi(\mathbf{x}, j) = \Pi_1(\mathbf{x}, j)$ , if and only if,  $g_1(\mathbf{x}_7, \mathbf{x}_2, \Pi_1)$  yields the maximum reward. Otherwise, the best option is chosen from  $\{\mathbf{x}_2, \mathbf{x}_8, \mathbf{x}_9\}$ , thereby guaranteeing that  $R(\Pi(\mathbf{x}_2, 1)) \geq R(\Pi_1(\mathbf{x}_2, 1))$ .

The linear system described by Equation (3.29a) defines, for the stochastic process  $\{\mathbf{X}_t^{\pi}, t \geq 0\}$ , an exact state-dependent model that relates the network reward rate  $R(\Pi)$  with all states  $\mathbf{x} \in \Omega_{\mathbf{x}}$ . The advantage of this state-dependent description is that for any class- $j$  arrival in state  $\mathbf{x}$ , the PIA improves the network reward rate by maximizing  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$ . The PIA guarantees that the optimum policy is always found [How60]. However, in most real networks, an online execution of the PIA is infeasible owing to the size  $|\Omega_{\mathbf{x}}|$  of the network state-space, which may be large enough as to preclude the calculations performed by both the VDO and the PIR. This is even the case for large capacity single-link networks. This fact is studied in [YZZ<sup>+</sup>14], where a thorough analysis is presented that calculates the number of states for a flex-grid link. Therein it is shown that the calculation of the state-space itself may be infeasible. A large state-space demands stringent memory requirements and computation times which are not realizable by current computational resources. Therefore, to circumvent this drawback, and to render the policy calculation feasible, a simplified model is needed that reduces the cardinality  $|\Omega_{\mathbf{x}}|$  of the network state-space. That model is proposed and discussed to the detail in Chapter 4, where it is shown that the process  $\{\mathbf{X}_t^{\pi}, t \geq 0\}$  can be described by linear systems formulated in the macrostate-spaces  $\Omega_n^l$  of the network links.

**Example 3.18** Consider a single-link network with a capacity of  $C_l = 128$  slots which serves three connections classes with  $b_1 = 2$ ,  $b_2 = 8$  and  $b_3 = 16$ . By using the formulae in [YZZ<sup>+</sup>14] (we skip the presentation of those formulas here as they are too voluminous and are not relevant for the forthcoming discussions), we have that  $|\Omega_{\mathbf{x}}| = 2.27 \cdot 10^{27}$  states. For this network, the solution to the linear system is computationally infeasible.

## 3.5. Chapter Summary

In this chapter a stochastic network model has been proposed for dynamic optical networks. The model was used to introduce four fundamental concepts. First, the concept of network state, which is defined as a valid configuration of the optical spectrum in the network. A spectrum configuration corresponds to the spectrum seized by the lightpaths allocated to carried connections, those lightpaths fulfil the spectrum continuity and contiguity constraints. Second, the concept of network macrostate, which is the traffic carried by the network at a given instant of time. It was shown that a macrostate can be represented by different network states. Third, the concept of connection reward, which is the benefit that the network receives from an admitted connection. And last, the concept of resource allocation policy, which is a collection of state-dependent decisions that dictate the course of action to take for any connection request. A policy decision instructs the network whether to accept or reject a connection request. In case of admission, it further defines a lightpath for the connection. Based on these concepts, an algorithm was proposed for online resource allocation that is suitable for stochastic networks which are Markovian. The algorithm tackles the problem of policy calculation as a Markov decision process, and uses an iteration procedure that finds the policy (i.e. the collection of decisions on resource allocation) that maximizes the rate at which the network earns reward. Although for networks with large state-spaces the algorithm may not be computationally feasible, a reward-based approach is advantageous, as by a proper definition of the reward parameters different optimization objectives can be achieved. In Chapter 4 it will be discussed a simplified version of the algorithm that renders the optimization problem solvable.



# Chapter 4

## Approximate Approach to Online Resource Allocation

In this chapter we formulate a simplified version of the policy iteration algorithm outlined in Chapter 3. The proposed algorithm is based on a link decomposition approach, whereby the rate of reward from the network is approximated as the sum of the link reward rates. Therefore, the problem is decomposed into separate link problems by assuming statistical independence of the link state distributions. This approach not only simplifies mathematical complexity, but the computational effort required to calculate decisions. Section 4.1 defines the network reward maximization problem based on the link independence assumption. To solve this problem, in Section 4.2 we formulate an approximate macrostate-dependent network reward model, whereby the link reward rates are calculated from linear systems defined in the link macrostate-spaces. Section 4.3 presents a method to estimate policy and macrostate-dependent link arrival rates from online measurements. The proposed method renders the macrostate-dependent network model adaptive to changing traffic conditions. In Section 4.4 the approximate policy iteration algorithm is formulated based on the macrostate-dependent model. In order to improve the accuracy of the algorithm, admission control rules are defined that mitigate at their best the correlations among links. To facilitate the presentation of the concepts introduced in this chapter, in Table 4.1 we summarize relevant variables and parameters defined throughout all chapter sections. The content in this chapter includes results published by the author in [RB16a, RB16b, RB16c, RB16d, RB17a].

### 4.1. Approximate Reward Maximization Problem

A topic of interest in the field of stochastic networks has been to find approximate models that circumvent the computational complexity involved in exact (state-dependent) network models. For loss or circuit-switched networks, an approach which has been proved to be suitable under certain limiting regimes is the reduced load approximation [Ke191, KY14, CR93]. It is based on the link independence assumption which supposes statistical independence of the link state distributions. In this section we derive the state-dependent network reward model that this assumption defines. Then in the subsequent sections we use the model to formulate an approximate approach to online resource allocation.

#### 4.1.1 The Link Independence Assumption

A stochastic optical network is said to fulfil the link independence property if there are no correlations (i.e. no mutual relationships) among the state distributions of the network links. In this case, each link blocks independently, and thus, the probability  $B_j^\rho$  that a path  $\rho \in \Gamma_j$  blocks a class- $j$  connection is:

$$B_j^\rho = 1 - \prod_{l \in \rho} (1 - B_j^l) \quad (4.1a)$$

where  $B_j^l$  is the probability that a class- $j$  connection gets blocked by the link  $l$  in  $\rho$ . Since the class- $j$  traffic is offered to all paths in  $\Gamma_j$ , the probability that the network blocks a class- $j$  connection is:

$$B_j = \prod_{\rho \in \Gamma_j} B_j^\rho \quad (4.1b)$$

Symbol	Units	Description
$R^l(\Pi)$	ru/uot	Mean rate at which link $l$ earns reward from carried connections
$r_j^l$	ru/con	Reward parameter of class- $j$ connections on link $l$
$\lambda_j^l(\Pi)$	con/uot	Mean rate at which class- $j$ connections arrive at link $l$
$A_j^l$	Erlangs	Mean class- $j$ traffic offered to link $l$
$A_{c_j}^l$	Erlangs	Mean class- $j$ traffic carried over link $l$
$B_j^l$	-----	Blocking probability of class- $j$ connections on link $l$
$\lambda_j^l(\mathbf{x}^l, \Pi)$	con/uot	Mean rate at which class- $j$ connection arrivals cause the link state transition $\mathbf{x}^l \rightarrow \mathbf{y}^l$
$v(\mathbf{x}^l, \Pi)$	ru	Transient reward earned by having set link $l$ in state $\mathbf{x}^l$ at $t_0$
$V(\mathbf{x}^l, \Pi, t)$	ru	Reward expected at $t \gg t_0$ , if link $l$ were set in state $\mathbf{x}^l$ at $t_0$ . This reward is earned at a constant link reward rate $R^l(\Pi)$
$\Gamma_{\mathbf{x}^l}^{j+}, \Gamma_{\mathbf{x}^l}^{j-}$	-----	Sets of link states reachable due to class- $j$ connection admissions and departures in link state $\mathbf{x}^l$ , respectively
$g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$	ru	State-dependent link reward gain $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) = v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi)$ . It is the long-term reward earned by link $l$ at $t \gg t_0$ from a class- $j$ connection arriving in state $\mathbf{x}^l$ that causes the link state transition $\mathbf{x}^l \rightarrow \mathbf{y}^l$
$\lambda_j^l(\mathbf{n}^l, \Pi)$	con/uot	Mean rate at which class- $j$ connection arrivals cause the link macrostate transition $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$
$v(\mathbf{n}^l, \Pi)$	ru	Transient reward earned by having set link $l$ in macrostate $\mathbf{n}^l$ at $t_0$
$V(\mathbf{n}^l, \Pi, t)$	ru	Reward expected at $t \gg t_0$ , if link $l$ were set in macrostate $\mathbf{n}^l$ at $t_0$ , this reward is earned at a constant link reward rate $R^l(\Pi)$
$g_j^l(\mathbf{n}^l, \Pi)$		$g_j^l(\mathbf{n}^l, \Pi) = v(\mathbf{n}^l + \delta_j^l, \Pi) - v(\mathbf{n}^l, \Pi)$ . Macrostate-dependent link reward gain. It is the long-term reward earned by link $l$ from a class- $j$ connection arriving in macrostate $\mathbf{n}^l$ that causes the transition $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$
$g_j(\mathbf{y}, \mathbf{x}, \Pi)$	ru	State-dependent network reward gain. It is the long-term reward brought to the network by a class- $j$ connection arriving in state $\mathbf{x}$ . Upon admission, the connection causes the network state transition $\mathbf{x} \rightarrow \mathbf{y}$ . This reward is estimated as follows: - Exact state-dependent model: $g_j(\mathbf{y}, \mathbf{x}, \Pi) = v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$ - Approximate state-dependent model: $g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$ - Approximate macrostate-dependent model: $g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi)$ where $\rho$ is the path on which the connection is routed
$\hat{X}_n^l$	-----	Set of link states $\mathbf{x}^l$ that yield the carried traffic (or macrostate) $\mathbf{n}^l$
$P_{n,j}^l$	-----	Probability of observing the macrostate $\mathbf{n}^l$ defined by a non-blocking state $\mathbf{x}^l$ in $\hat{X}_n^l$ for class- $j$ traffic

**Table 4.1:** Notation for relevant variables and parameters defined in Chapter 4.

As a result of this, if class- $j$  connections arrive at the network following a Poisson process with mean arrival rate  $\lambda_j$ , then class- $j$  traffic arrives at link  $l$  with a rate:

$$\lambda_j^l(\Pi) = \lambda_j^\rho(\Pi) \cdot \prod_{s \in \rho \setminus \{l\}} (1 - B_j^s) \quad (4.2)$$

which describes an arrival process that is Poissonian as well [Kel91, KY14, Dzi97], where  $\lambda_j^\rho(\Pi)$  is the arrival rate of class- $j$  traffic at path  $\rho$ . The rates  $\lambda_j^\rho(\Pi)$  fulfil:

$$\lambda_j = \sum_{\rho \in \Gamma_j} \lambda_j^\rho(\Pi) \quad (4.3)$$

Observe that both  $\lambda_j^\rho(\Pi)$  and  $\lambda_j^l(\Pi)$  depend on the routing decisions made by the policy  $\Pi$ . Based on this, we have that  $\lambda_j^\rho(\Pi)$  can be determined as [Dzi97]:

$$\lambda_j^\rho(\Pi) = \lambda_j \cdot \frac{\hat{\lambda}_j^\rho(\Pi)}{\sum_{\rho \in \Gamma_j} \hat{\lambda}_j^\rho(\Pi)} \quad (4.4)$$

where  $\hat{\lambda}_j^\rho(\Pi)$  is the acceptance rate of class- $j$  traffic on path  $\rho$ . This rate can be determined from online traffic measurements.

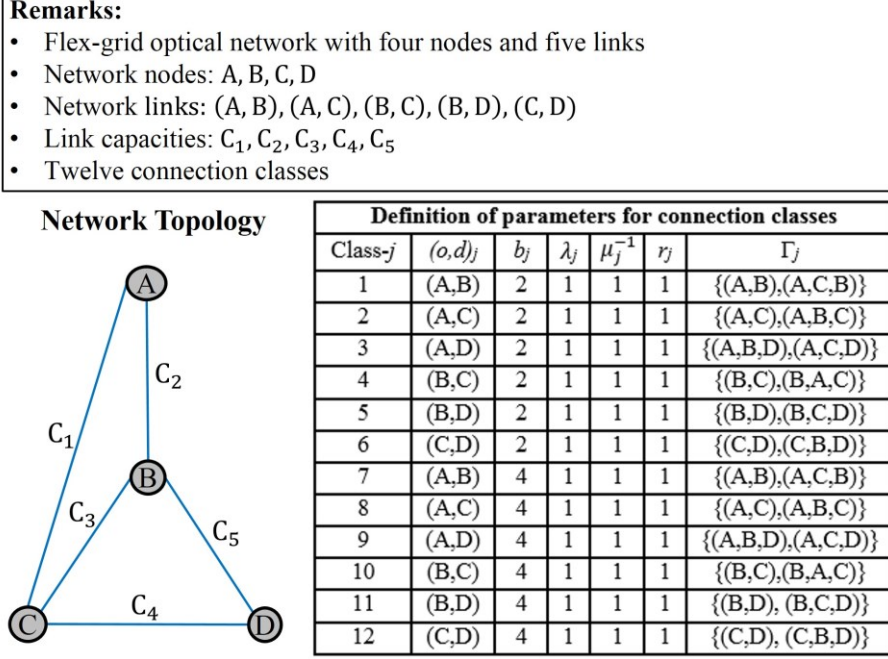
Equation (4.2) states that the traffic offered to link  $l$  is the traffic offered to the path  $\rho$  reduced at every link  $s \neq l$  in  $\rho$  by a factor  $(1 - B_j^s)$ . As a result,  $\lambda_j^l(\Pi)$  describes a Poisson process from which the link performance, defined as the link reward rate  $R^l(\Pi)$ , can be evaluated from a performance function that depends on  $\lambda_j^l(\Pi)$ , the link capacity  $C_l$ , and the parameters  $\mu_j^{-1}$ ,  $b_j$  and  $r_j$  of the  $J$  connection classes:

$$R^l(\Pi) = f(\lambda_j^l(\Pi), \mu_j^{-1}, b_j, r_j, C_l) \quad (4.5)$$

Therefore, if the network fulfils the link independence property - as defined by Equations (4.1)-(4.4), two equivalent approaches may evaluate the network reward rate  $R(\Pi)$ . First, the exact state-dependent model derived in Chapter 3. And secondly, a link-based approach whereby  $R(\Pi)$  is determined by the performance functions of the network links. Such functions are given by Equation (4.5) which, as it will be shown in Section 4.1.3, defines a linear system similar to that obtained for the exact network model.

From a computational perspective, the link based-approach is more advantageous as the performance functions are defined over link state-spaces which have smaller cardinalities than the network state-space. By this approach exact results are only obtained when the link independence property is satisfied. This occurs in networks that mainly route traffic on direct link paths or in networks that, besides operating at low traffic loads, use multi-link paths with a few number of links [Dzi97, CR93]. Although not all real network scenarios comply with these conditions, the link-based approach can still be applied (owing to its reduced computational complexity) as an approximate method to estimate the network reward rate  $R(\Pi)$ . In most real networks correlations exist and they can be classified into two types, namely, link and path correlations. The former ensue from connections routed over multi-link paths (the larger the number of links in the path, the larger the correlations). The latter stem from the routing decisions that assign traffic flows to the paths in  $\Gamma_j$ . If the link-based approach estimates  $R(\Pi)$  in networks where correlations of either type exist, there will be an estimation error that grows with the strength of the correlations. In a specific network scenario, it is difficult to quantify how strong or weak these correlations are. The reason is that they are not only influenced by the definition of the set  $\Gamma_j$  (i.e. the definition of the number of routes and the number of links in each route) but are also determined by the traffic characteristics and the policy in use. In spite of this, the accuracy of the approach can be improved by employing simple network design rules that counteract correlations. One of them is to calculate sets  $\Gamma_j$  with a few number of paths, where each path has a short length w.r.t. the number of links. Another strategy is to implement admission control rules that avoid accepting connections on large multi-link paths (especially in high traffic load conditions).

The term ‘‘link independence assumption’’ has widely been used in the literature to refer to network models that assume statistical independence of the link state distributions – and thus, those models rely on link-based approaches that estimate the network performance. For example, the performance evaluation results in [Kel88, Kel11, Kel91, KY14, CR93, DPKW88, DM89, DM92, DM94, Kri91, HKT00, Hwa93, Dzi97, Nor02] show that, for telephone and packet-switched data networks, the assumption is a valid approach to designing network control mechanisms. Inspired by these results, in the following we use the assumption to formulate an approximate state-dependent stochastic model that estimates the network reward rate  $R(\Pi)$ . A comprehensive study of the link independence assumption, and its implications, can be found in [Dzi97].



**Figure 4.1:** Flex-grid optical network with four nodes, five links and 12 classes [RB16d].

## 4.1.2 Reformulation of the Network Reward Maximization Problem

The link independence assumption has successfully been applied to reward-based routing in packet-switched networks, see for example the approaches proposed in [DPKW88, DM89, DM92, DM94, Kri91, HKT00, Hwa93, Dzi97, Nor02]. By using this approximation, we have that the rate  $R(\Pi)$  at which the network earns reward can be calculated as the sum of the link reward rates  $R^l(\Pi)$ :

$$R(\Pi) = \sum_l R^l(\Pi) \quad (4.6)$$

and therefore, the network reward maximization problem is now defined as:

$$R^* = \max_{\Pi} \sum_l R^l(\Pi) \quad (4.7)$$

Since Equation (4.6) assumes that all network links block independently, then the assumption is made that any connection modifies  $R(\Pi)$  by solely changing the rates  $R^l(\Pi)$  of the links on which it is carried. This implies that for a class- $j$  arrival in state  $\mathbf{x}$ , the network reward rate attainable by the policy decision  $\Pi(\mathbf{x}, j)$  is calculated from Equation (4.6) as:

$$R(\Pi(\mathbf{x}, j)) = \sum_{l \in \rho} R^l(\Pi(\mathbf{x}, j)) \quad (4.8)$$

where the sum is over the links in the path  $\rho$  on which the connection is routed. Notice that, as seen in Chapter 3, the selected path  $\rho$  is implicitly defined by the state  $\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}$ , which results from the transition  $\mathbf{x} \rightarrow \mathbf{y}$  caused by the decision  $\Pi(\mathbf{x}, j)$ . The striking consequence of this is that to calculate a decision that outperforms  $\Pi(\mathbf{x}, j)$ , it is only necessary to know the reward rates of the links that make up the paths in  $\Gamma_j$ . The applicability of this result will be explored in Section 4.4. In the meantime, let us formulate the state-dependent network reward model that results from the link independence assumption.

**Example 4.1** Consider the optical network in Fig. 4.1, which has four nodes, five links and serves 12 connection classes (two classes per node-pair). Under the link independence assumption, the network reward rate is given by  $R(\Pi) = R^1(\Pi) + R^2(\Pi) + R^3(\Pi) + R^4(\Pi) + R^5(\Pi)$ . Let us consider class-7 connections, which can only be routed over the paths in the set  $\Gamma_7 = \{(A, B), (A, C, B)\}$ . Under the link independence assumption, it is assumed that class-7 can only modify  $R(\Pi)$  by changing the rates of the links in  $\Gamma_7$ . For instance, if the policy  $\Pi$  has a decision  $\Pi(\mathbf{x}, 7)$ , such that a class-7 connection is admitted in state  $\mathbf{x}$  on the path  $\rho = (A, C, B)$ , then from Equation (4.8) we have that  $R(\Pi(\mathbf{x}, 7)) = R^1(\Pi(\mathbf{x}, 7)) + R^3(\Pi(\mathbf{x}, 7))$ , with  $l = 1$  and  $l = 3$  denoting link (A, C) and link (C, B), respectively.



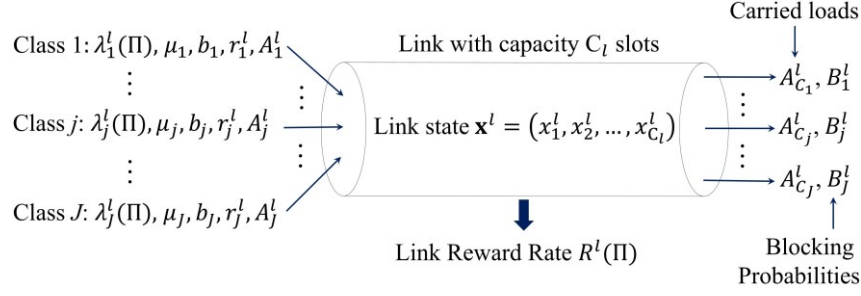


Figure 4.2: Flex-grid optical link.

### 4.1.3 Approximate State-Dependent Network Reward Model

Consider a dynamic flex-grid optical network with  $N$  nodes and  $L$  links that serves  $J$  connection classes. Every link  $l$  has a capacity of  $C_l$  spectrum slots. The exact state-dependent reward model for this network is given by the set of Equations (3.29) in Chapter 3, where  $R(\Pi)$  is calculated by solving a linear system defined in the state-space  $\Omega_{\mathbf{x}}$ . Before applying the link independence assumption to the derivation of a simplified network reward model, let us first calculate the rate at which a network link  $l$  earns reward.

The network stochastic process  $\{\mathbf{X}_t^{\pi}, t \geq 0\}$  implicitly defines for every link  $l$ , a continuous-time stochastic process  $\{\mathbf{X}_t^{l,\pi}, t \geq 0\}$ . This process describes the time evolution of the link state when a policy  $\Pi$  is used. The random variable  $\mathbf{X}_t^{l,\pi}$  takes its values from the link state-space  $\Omega_{\mathbf{x}}^l$ , and thereby,  $\mathbf{X}_t^{l,\pi}$  is the link state at time  $t$ . The stochastic properties of this process stem from the parameters  $(o, d)_j$ ,  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$  that define each connection class. In Fig. 4.2, we present a simplified view of the most relevant parameters and variables that describe the stochastic behaviour of a link  $l$ .

Class- $j$  connections arrive at link  $l$  at a rate  $\lambda_j^l(\Pi)$  (con/uo $t$ ), which is policy-dependent as it originates from the decisions defined in  $\Pi$ , and thus, we have that according to Equation (4.2),  $\lambda_j^l(\Pi)$  is not equal to the arrival rate  $\lambda_j$ . Then class- $j$  offers to the link a traffic load (in Erlangs) given by:

$$A_j^l = \frac{\lambda_j^l(\Pi)}{\mu_j} \quad (4.9)$$

From this load, the link carries a traffic:

$$A_{C_j}^l = A_j^l \cdot (1 - B_j^l) \quad (4.10)$$

where  $B_j^l$  is the probability that a class- $j$  connection request is blocked by the link. If the network is in state  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^l, \dots, \mathbf{x}^L)$ , and the policy  $\Pi$  admits a class- $j$  connection on a path  $\rho$ , such that  $l \in \rho$ , then the link  $l$  makes the state transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , where  $\mathbf{x}^l, \mathbf{y}^l \in \Omega_{\mathbf{x}}^l$ . As a result of this event, the connection seizes  $b_j$  adjacent slots in the link, and the network earns an immediate reward of  $r_j$  (ru). From this reward,  $r_j^l$  (ru) are earned over link  $l$ . We denote this reward as the link reward parameter which is calculated as:

$$r_j^l = \begin{cases} \frac{r_j}{|\rho|}, & \text{if } j \in J^l \\ 0, & \text{if } j \notin J^l \end{cases} \quad (4.11)$$

Recall that  $J^l$  is the set of class- $j$  connections which can be carried on link  $l$  (see Table 3.1 in Chapter 3). Thus, for class- $j$  connections which can never be routed through link  $l$ , we have that  $r_j^l = 0$ , i.e. the link  $l$  is not within the paths in  $\Gamma_j$ . Otherwise,  $r_j^l$  is  $r_j$  divided by the number of links  $|\rho|$  in the path  $\rho$ , i.e. the reward parameter  $r_j$  is evenly distributed among all links in  $\rho$ . This reward division rule has been studied along with other rules in [DM89, DM92, DM94, Dzi97] for routing in telephone and packet-switched networks. The performance results therein presented show that Equation (4.11) gives the simplest and most effective rule to apportion the reward parameters  $r_j$  among network links. We then interpret  $r_j^l$  as the immediate or short-term reward earned by a link  $l$  from a class- $j$  connection.

**Example 4.2** In order to illustrate the reward division rule, consider the link  $l = 5$  in Fig. 4.1, i.e. that interconnecting the nodes B – D. This link is in the path  $\rho = (A, B, D) \in \Gamma_3, \Gamma_9$ , which has  $|\rho| = 2$  links, it is also in the path  $\rho = (B, D) \in \Gamma_5, \Gamma_{11}$ , which has  $|\rho| = 1$ , and it is in the path  $\rho = (C, B, D) \in \Gamma_6, \Gamma_{12}$ , which has  $|\rho| = 2$  (see Fig. 4.1). Therefore, the classes carried by this link are  $J^5 = \{3, 5, 6, 9, 11, 12\}$ . From

Equation (4.11), these six classes define on the link B – D the reward parameters:  $r_3^5 = 0.5$ ,  $r_5^5 = 1.0$ ,  $r_6^5 = 0.5$ ,  $r_9^5 = 0.5$ ,  $r_{11}^5 = 1.0$  and  $r_{12}^5 = 0.5$  (ru). For the remaining six classes  $r_j^5 = 0$ , as they are not defined in  $J^5$ . If a class-12 connection is routed over the path (C,B,D), it brings an immediate reward of  $r_{12} = 1$  (ru) to the network, where from this reward  $r_{12}^5 = 0.5$  (ru) are earned over the link B – D.

Since  $A_{c_j}^l$  is the mean number of class- $j$  connections simultaneously carried on link  $l$ , class- $j$  traffic is expected to yield reward at a mean rate  $r_j^l \cdot \mu_j \cdot A_{c_j}^l$ . Therefore, link  $l$  earns reward at a mean rate  $R^l(\Pi) = \sum_j r_j^l \cdot \mu_j \cdot A_{c_j}^l$  (ru/uot), from which we have that the link performance function in Equation (4.5) is:

$$R^l(\Pi) = \sum_{j=1}^J r_j^l \cdot \lambda_j^l(\Pi) \cdot (1 - B_j^l) \quad (4.12)$$

Let us now apply the link independence assumption to calculate the dependence of Equation (4.12) on the link states. If at time  $t_n$  the network is in state  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^l, \dots, \mathbf{x}^L)$ , the probability of observing at time  $t_{n+1}$  the link  $l$  in a state  $\mathbf{y}^l$  (as a result of a transition  $\mathbf{x} \rightarrow \mathbf{y}$  which causes the link state transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ ) depends only on the link state  $\mathbf{x}^l$  at time  $t_n$ :

$$P\{\mathbf{X}_{t_{n+1}}^{l,\pi} = \mathbf{y}^l | \mathbf{X}_{t_n}^{1,\pi} = \mathbf{x}^1, \dots, \mathbf{X}_{t_n}^{l,\pi} = \mathbf{x}^l, \dots, \mathbf{X}_{t_n}^{L,\pi} = \mathbf{x}^L\} = P\{\mathbf{X}_{t_{n+1}}^{l,\pi} = \mathbf{y}^l | \mathbf{X}_{t_n}^{l,\pi} = \mathbf{x}^l\} \quad (4.13)$$

i.e. the states of the remaining links at time  $t_n$  have no influence on the link state transition (and thus, the correlations among links are negligible). Based on this, we have that each network link can be treated independently from one another, i.e. the optical network behaves as a compound of  $L$  independent single-link networks. Therefore, the dependence of  $R^l(\Pi)$  on the link state  $\mathbf{x}^l$  is obtained by applying the set of Equations (3.29) in Chapter 3 to the case of a single-link network [RB16b]:

$$R^l(\Pi) = q(\mathbf{x}^l) + \sum_{j=1}^J \sum_{\mathbf{y}^l \in \Gamma_{\mathbf{x}^l}^{j+}} \lambda_j^l(\mathbf{x}^l, \Pi) \cdot [v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi)] + \sum_{j=1}^J \sum_{\mathbf{y}^l \in \Gamma_{\mathbf{x}^l}^{j-}} \mu_j \cdot [v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi)] \quad , \mathbf{x}^l \in \Omega_{\mathbf{x}}^l \quad (4.14a)$$

where  $\Gamma_{\mathbf{x}^l}^{j+}$  and  $\Gamma_{\mathbf{x}^l}^{j-}$  are the sets of link states which are reachable due to class- $j$  connection admissions and departures in state  $\mathbf{x}^l$ , respectively. The first double summation is the contribution to  $R^l(\Pi)$  owing to connection requests that arrive at the link in state  $\mathbf{x}^l$ . Connections arrive at a state-dependent rate  $\lambda_j^l(\mathbf{x}^l, \Pi)$  - the actual relationship between  $\lambda_j^l(\mathbf{x}^l, \Pi)$  and  $\lambda_j^l(\Pi)$  will be studied in Section 4.3. The second double summation is the contribution to the link reward rate due to departures. Moreover,  $q(\mathbf{x}^l)$  is the rate at which the link yields reward in state  $\mathbf{x}^l$ , and is given by:

$$q(\mathbf{x}^l) = \sum_{j=1}^J r_j^l \cdot \mu_j(\mathbf{x}^l) \quad (4.14b)$$

with  $\mu_j(\mathbf{x}^l)$  being the termination rate of carried class- $j$  connections, which is calculated as:

$$\mu_j(\mathbf{x}^l) = \mu_j \cdot n_j(\mathbf{x}^l) \quad (4.14c)$$

where  $n_j(\mathbf{x}^l) \geq 0$  is the number of class- $j$  connections carried by the link in state  $\mathbf{x}^l$ . In addition to Equation (4.14a), link  $l$  fulfils the following two equations:

$$V(\mathbf{x}^l, \Pi, t) = R^l(\Pi) \cdot t + v(\mathbf{x}^l, \Pi) \quad , \mathbf{x}^l \in \Omega_{\mathbf{x}}^l \quad (4.14d)$$

$$g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) = V(\mathbf{y}^l, \Pi, t) - V(\mathbf{x}^l, \Pi, t) = v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi) \quad (4.14e)$$

with  $V(\mathbf{x}^l, \Pi, t)$  being the reward expected at  $t \gg t_0$ , if the link was in state  $\mathbf{x}^l$  at  $t_0$ . Equation (4.14d) states that in steady state, the link earns reward at a rate  $R^l(\Pi)$  regardless of the link state  $\mathbf{x}^l$  at  $t_0$ . Furthermore, Equation (4.14e) defines  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$  as the state-dependent link reward gain. It is the long-term reward that a class- $j$  connection brings to the link if it gets admission in state  $\mathbf{x}^l$ .

Equation (4.14a) defines a linear system over all link states in  $\Omega_{\mathbf{x}}^l$ . Given that for all  $\mathbf{x}^l$  the parameters  $q(\mathbf{x}^l)$ ,  $\lambda_j^l(\mathbf{x}^l, \Pi)$  and  $\mu_j$  are known, the linear system is solved for  $R^l(\Pi)$  and the values  $v(\mathbf{x}^l, \Pi)$ . This is accomplished by arbitrarily setting one of the values  $v(\mathbf{x}^l, \Pi)$  to zero. Thus, under the link independence assumption, Equations (4.14) define an approximate, state-dependent network reward model whereby the

Variable	Exact State-Dependent Network Reward Model	Approximate State-Dependent Network Reward Model
Rate $R(\Pi)$	Obtained by solving a linear system of $ \Omega_{\mathbf{x}} $ equations	Obtained by solving $L$ linear systems, one for each link reward rate $R^l(\Pi)$
Transient values	There are $ \Omega_{\mathbf{x}} $ values $v(\mathbf{x}, \Pi)$ which are obtained from the solution to the linear system that calculates $R(\Pi)$	For each link $l$ , there are $ \Omega_{\mathbf{x}}^l $ values $v(\mathbf{x}^l, \Pi)$ which are obtained from the solution to the linear system for link $l$
Gain $g_j(\mathbf{y}, \mathbf{x}, \Pi)$	Calculated from the values $v(\mathbf{x}, \Pi)$ as: $g_j(\mathbf{y}, \mathbf{x}, \Pi) = v(\mathbf{y}, \Pi) - v(\mathbf{x}, \Pi)$	Calculated as follows: $g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$ where the link gains are given by: $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) = v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi)$

**Table 4.2:** Comparison between the exact and the approximate state-dependent reward models.

the calculation of the rate  $R(\Pi)$  is performed as follows. First, besides the policy  $\Pi$ , the parameters  $(o, d)_j$ ,  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$  are defined. Second, for each link, these parameters and the link capacity  $C_l$  are used to calculate the state-space  $\Omega_{\mathbf{x}}^l$  (every state in  $\Omega_{\mathbf{x}}^l$  must fulfil the spectrum contiguity constraint). Third, the linear system for every link is defined from Equation (4.14a) based on the decisions of the policy  $\Pi$ . Then each system is solved for  $R^l(\Pi)$  and the values  $v(\mathbf{x}^l, \Pi)$ . Finally, Equation (4.6) is used to calculate  $R(\Pi)$  as the sum of the  $L$  reward rates  $R^l(\Pi)$ . Compared to the exact model, which only requires the solution of a linear system, the approximate approach involves  $L$  independent linear systems. However, those systems have cardinalities  $|\Omega_{\mathbf{x}}^l| \ll |\Omega_{\mathbf{x}}|$ , which to some extent alleviate the computational requirements needed to estimate the network reward rate  $R(\Pi)$ .

From the exact state-dependent model derived in Chapter 3, we know that when a class- $j$  connection gets admission in state  $\mathbf{x}$ , it brings a short-term reward  $r_j$  and a long-term reward  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$ . Under the link independence assumption, from Equation (4.8) we have that  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$  is approximated as:

$$g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) \quad (4.15)$$

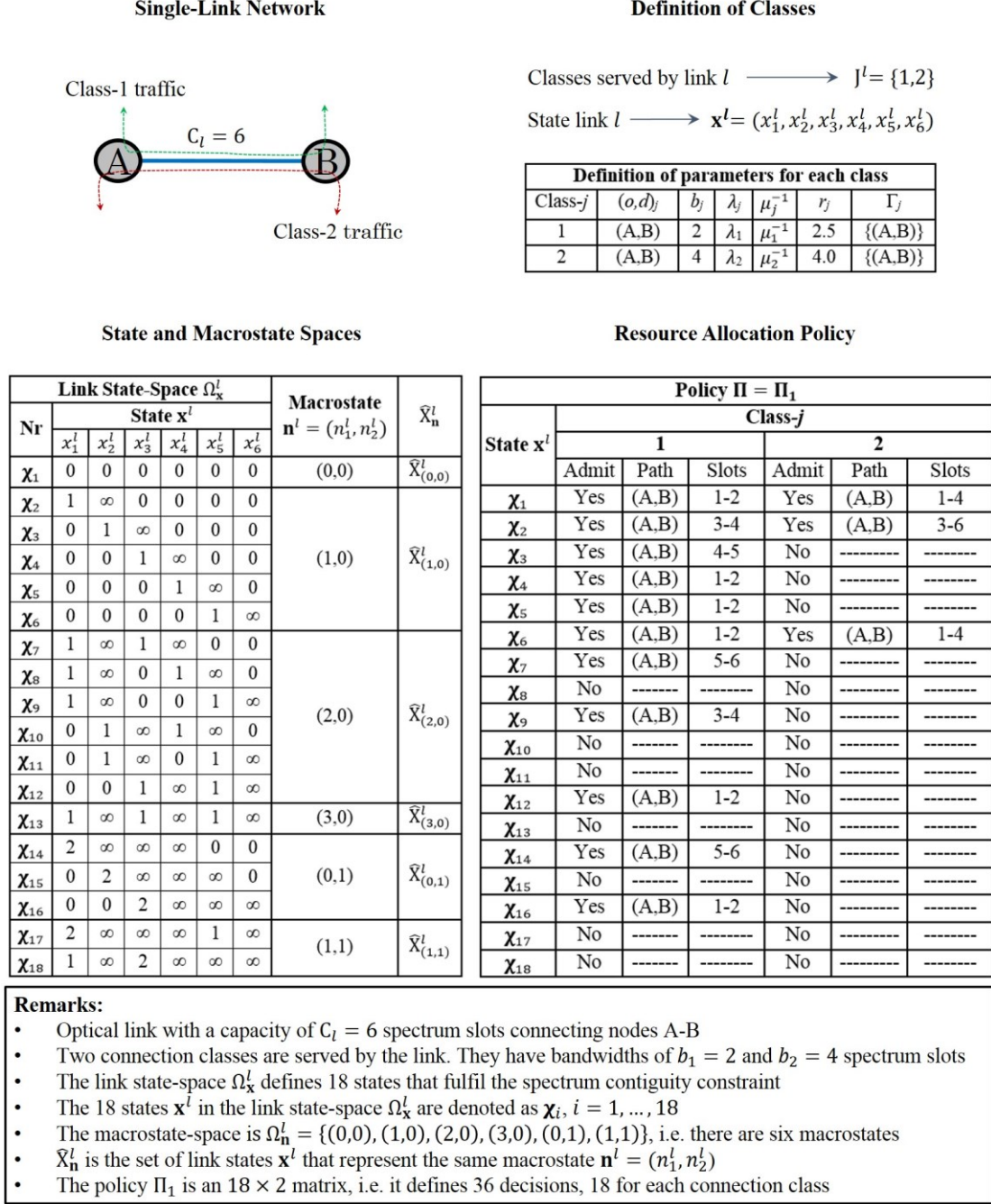
i.e.  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$  is simply the sum of the long-term rewards that the connection brings to the links in the path  $\rho$  that it uses. Equation (4.15) implicitly assumes that any carried connection affects the network reward process by solely changing the rewards of the links in  $\rho$ .

In Table 4.2 we compare the exact reward model studied in Chapter 3 and the approximate model that uses the link independence assumption. We have previously argued that the drawback of the exact model is that for most real networks the cardinality  $|\Omega_{\mathbf{x}}|$  is large enough as to preclude the calculation of  $R(\Pi)$  and the values  $v(\mathbf{x}, \Pi)$ . This may also be the case for the link state-spaces in the approximate reward model (see Example 3.18 in Chapter 3). To circumvent this problem, an alternative is to calculate the link reward rates  $R^l(\Pi)$  by defining linear systems in the link macrostate-spaces  $\Omega_{\mathbf{n}}^l$  rather than in  $\Omega_{\mathbf{x}}^l$ .

## 4.2. Macrostate-Dependent Network Reward Model

In this section we formulate an approximate, macrostate-dependent network reward model based on a simplification of the state-dependent link model in Equation (4.14a). The purpose is to calculate the link reward rate  $R^l(\Pi)$  as a function of the link macrostates. Since the macrostate-space  $\Omega_{\mathbf{n}}^l$  has a cardinality  $|\Omega_{\mathbf{n}}^l| \ll |\Omega_{\mathbf{x}}^l|$ , the computational complexity of the state-dependent model (for large networks) is avoided, which makes the reward maximization problem, under the link independence assumption, solvable.

Any link state transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  is caused either by the admission or departure of a connection. These two events manifest through changes in the traffic carried by the link. This traffic was defined in Chapter 3 as the link macrostate  $\mathbf{n}^l = (n_1^l, \dots, n_j^l, \dots, n_J^l)$ , where  $n_j^l$  is the number of carried class- $j$  connections. Let  $\delta_j^l$  be a  $J$ -dimensional vector with a one in position  $j$  and zeros in the other positions, and let us assume that the link state  $\mathbf{x}^l$  defines a carried traffic  $\mathbf{n}^l$ . If the transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  is caused by a class- $j$  arrival, the carried traffic changes as  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l = (n_1^l, \dots, n_j^l + 1, \dots, n_J^l)$ . Otherwise, if the transition originates



**Figure 4.3:** State and macrostate spaces for a single-link network that serves two connection classes.

from a class- $j$  departure, the carried traffic varies as  $\mathbf{n}^l \rightarrow \mathbf{n}^l - \delta_j^l = (n_1^l, \dots, n_j^l - 1, \dots, n_j^l)$ . In both cases, the macrostates  $\mathbf{n}^l + \delta_j^l$  and  $\mathbf{n}^l - \delta_j^l$  are defined by their corresponding destination states  $\mathbf{y}^l$ . In Chapter 3, it was also shown that any link macrostate  $\mathbf{n}^l$  is explicitly defined by a sub-set of network states  $\mathbf{x}$ , i.e. there can be different spectrum configurations  $\mathbf{x}$  that define the same macrostate  $\mathbf{n}^l$ , and thereby different link states  $\mathbf{x}^l$  may represent the same traffic load  $\mathbf{n}^l$ . Let  $\bar{X}_n^l$  be the set of link states  $\mathbf{x}^l$  that yield the carried traffic  $\mathbf{n}^l$ . Then we have that the macrostate transitions  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$  and  $\mathbf{n}^l \rightarrow \mathbf{n}^l - \delta_j^l$  need not be caused by a unique transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ . In general, the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$  can be caused by any of the possible link state transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , such that  $\mathbf{x}^l \in \bar{X}_n^l$  and  $\mathbf{y}^l \in \bar{X}_{\mathbf{n}+\delta_j^l}^l$ . Similarly,  $\mathbf{n}^l \rightarrow \mathbf{n}^l - \delta_j^l$  can be caused by any of the possible link state transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , such that  $\mathbf{x}^l \in \bar{X}_n^l$  and  $\mathbf{y}^l \in \bar{X}_{\mathbf{n}-\delta_j^l}^l$ .

**Example 4.3** Consider the single-link network shown in Fig. 4.3. It has a capacity of  $C_l = 6$  slots and serves two connection classes with bandwidths  $b_1 = 2$  and  $b_2 = 4$ . The link state-space  $\Omega_x^l$  consists of 18 states  $\mathbf{x}^l$  denoted as  $\mathbf{x}^l = (x_1^l, x_2^l, x_3^l, x_4^l, x_5^l, x_6^l)$ , which are indexed in Fig. 4.3 as  $\chi_i, i = 1, \dots, 18$ . Each

state fulfils the contiguity constraint. The link allocates resources with a policy  $\Pi = \Pi_1$  that consists of 36 decisions, 18 for each connection class. The carried traffic is given by the macrostate  $\mathbf{n}^l = (n_1^l, n_2^l)$ . Any feasible  $\mathbf{n}^l$  satisfies the capacity constraint  $\sum_{j=1}^2 b_j \cdot n_j^l \leq 6$ , and is defined by a sub-set of link states  $\widehat{\mathcal{X}}_{\mathbf{n}}^l$ . The 18 link states define the six macrostates in  $\Omega_{\mathbf{n}}^l = \{(0,0), (1,0), (2,0), (3,0), (0,1), (1,1)\}$ . For each macrostate, Fig. 4.3 shows the corresponding set  $\widehat{\mathcal{X}}_{\mathbf{n}}^l$ . For example, the carried traffic  $\mathbf{n}^l = (1,1)$  is observable when the link is in one of the two states in  $\widehat{\mathcal{X}}_{(1,1)}^l = \{\mathbf{x}_{17}, \mathbf{x}_{18}\}$ . These states are defined as  $\mathbf{x}_{17} = (2, \infty, \infty, \infty, 1, \infty)$  and  $\mathbf{x}_{18} = (1, \infty, 2, \infty, \infty, \infty)$ . This illustrates that different link states  $\mathbf{x}^l$  may configure the same traffic load  $\mathbf{n}^l$ .

**Example 4.4** Consider the macrostate  $\mathbf{n}^l = (1,0)$  in Fig. 4.3, which is observable when the link is in any state in  $\widehat{\mathcal{X}}_{(1,0)}^l = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ . The transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_1^l$ , i.e.  $(1,0) \rightarrow (2,0)$ , is caused by a class-1 arrival. Observe that the destination macrostate  $(2,0)$  is defined by the states in  $\widehat{\mathcal{X}}_{\mathbf{n}+\delta_1^l}^l = \widehat{\mathcal{X}}_{(2,0)}^l = \{\mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{10}, \mathbf{x}_{11}, \mathbf{x}_{12}\}$ . By using the policy  $\Pi_1$  in Fig. 4.3, we have that the transition  $(1,0) \rightarrow (2,0)$  can only be caused by any of the five transitions of the form  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , such that  $\mathbf{x}^l \in \widehat{\mathcal{X}}_{(1,0)}^l$  and  $\mathbf{y}^l \in \widehat{\mathcal{X}}_{(2,0)}^l$ . They are:  $\mathbf{x}_2 \rightarrow \mathbf{x}_7, \mathbf{x}_3 \rightarrow \mathbf{x}_{10}, \mathbf{x}_4 \rightarrow \mathbf{x}_7, \mathbf{x}_5 \rightarrow \mathbf{x}_8$  and  $\mathbf{x}_6 \rightarrow \mathbf{x}_9$ . For the sake of clarity, note that when a class-1 arrival occurs in state  $\mathbf{x}_2 = (1, \infty, 0, 0, 0, 0)$ , the policy in Fig. 4.3 says that the connection must be admitted in slots 3-4. This decision causes the transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_7$ , with  $\mathbf{x}_7 = (1, \infty, 1, \infty, 0, 0)$ , and thereby the traffic load  $(2,0)$  is now carried. By following this reasoning, the remaining aforementioned four state transitions are obtained. Similarly, we have that the transition  $(1,0) \rightarrow (1,1)$ , due to a class-2 arrival, can only be caused by any of the two transitions  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$  or  $\mathbf{x}_6 \rightarrow \mathbf{x}_{17}$ , with  $\mathbf{x}_{17}, \mathbf{x}_{18} \in \widehat{\mathcal{X}}_{(1,1)}^l$ . Furthermore, a class-1 departure causes the transition  $(1,0) \rightarrow (0,0)$ , which may originate from the five transitions  $\mathbf{x}_2 \rightarrow \mathbf{x}_1, \mathbf{x}_3 \rightarrow \mathbf{x}_1, \mathbf{x}_4 \rightarrow \mathbf{x}_1, \mathbf{x}_5 \rightarrow \mathbf{x}_1$  and  $\mathbf{x}_6 \rightarrow \mathbf{x}_1$ , with  $\mathbf{x}_1 \in \widehat{\mathcal{X}}_{(0,0)}^l$ . This example evinces that, in general, macrostate transitions  $\mathbf{n}^l \rightarrow \mathbf{n}^l \pm \delta_j^l$  are not necessarily caused by a unique transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ .

For a given state  $\mathbf{x}^l$ , Equation (4.14a) calculates  $R^l(\Pi)$  as a linear function of all transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  caused by arrivals and departures when the link is in state  $\mathbf{x}^l$ . To simplify this approach, let us use the fact that different link states  $\mathbf{x}^l$  may represent the same traffic  $\mathbf{n}^l$ . Then, for every  $\mathbf{n}^l$ , we define an equation that calculates  $R^l(\Pi)$  as a linear function of all transitions  $\mathbf{n}^l \rightarrow \mathbf{n}^l \pm \delta_j^l$ . By doing so, the link states that may define the macrostate  $\mathbf{n}^l$ , i.e. all  $\mathbf{x}^l \in \widehat{\mathcal{X}}_{\mathbf{n}}^l$ , are no longer described by separate equations, but by a single one that implicitly takes into account all the transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  that cause  $\mathbf{n}^l \rightarrow \mathbf{n}^l \pm \delta_j^l$ . Therefore, let  $\{\mathbf{N}_t^{l,\pi}, t \geq 0\}$  be a continuous-time stochastic process that models the time evolution of the macrostate of a link  $l$  under a policy  $\Pi$ . The random variable  $\mathbf{N}_t^{l,\pi}$  takes its values from the macrostate-space  $\Omega_{\mathbf{n}}^l$  and thereby,  $\mathbf{N}_t^{l,\pi}$  is the traffic carried by the link at time  $t$ . By following the same analysis made for the exact network reward model in Section 3.3.2 - Chapter 3, we have that for a single-link network, the process  $\{\mathbf{N}_t^{l,\pi}, t \geq 0\}$  yields reward at a rate  $R^l(\Pi)$  given by [RB16a, RB16b, RB17a]:

$$R^l(\Pi) = q(\mathbf{n}^l) + \sum_{j=1}^J \lambda_j^l(\mathbf{n}^l, \Pi) \cdot [v(\mathbf{n}^l + \delta_j^l, \Pi) - v(\mathbf{n}^l, \Pi)] + \sum_{j=1}^J n_j^l \cdot \mu_j \cdot [v(\mathbf{n}^l - \delta_j^l, \Pi) - v(\mathbf{n}^l, \Pi)] \quad , \mathbf{n}^l \in \Omega_{\mathbf{n}}^l \quad (4.16a)$$

The first summation is the contribution to  $R^l(\Pi)$  owing to connection requests which arrive at the link in macrostate  $\mathbf{n}^l$ . Connections arrive at a macrostate-dependent rate  $\lambda_j^l(\mathbf{n}^l, \Pi)$ , which is implicitly defined (as it will be shown in Section 4.3) by the arrival rates  $\lambda_j^l(\mathbf{x}^l, \Pi)$  and  $\lambda_j^l(\Pi)$ . The second summation is the contribution to the link reward rate due to departures in macrostate  $\mathbf{n}^l$ . Furthermore,  $q(\mathbf{n}^l)$  is the rate at which the link yields reward in macrostate  $\mathbf{n}^l$  and is given by [RB16a, RB16b, RB17a]:

$$q(\mathbf{n}^l) = \sum_{j=1}^J r_j^l \cdot n_j^l \cdot \mu_j \quad (4.16b)$$

where  $n_j^l \cdot \mu_j$  is the termination rate of class- $j$  traffic in macrostate  $\mathbf{n}^l$ . Moreover, in Equation (4.16a),  $v(\mathbf{n}^l, \Pi)$  is the transient reward value for having set the link in macrostate  $\mathbf{n}^l$  at time  $t_0$ . Under the macrostate-dependent description, we obtain two equations analogous to Equations (4.14d)-(4.14e):

$$V(\mathbf{n}^l, \Pi, t) = R^l(\Pi) \cdot t + v(\mathbf{n}^l, \Pi) \quad , \mathbf{n}^l \in \Omega_{\mathbf{n}}^l \quad (4.16c)$$

$$g_j^l(\mathbf{n}^l, \Pi) = V(\mathbf{n}^l + \delta_j^l, \Pi, t) - V(\mathbf{n}^l, \Pi, t) = v(\mathbf{n}^l + \delta_j^l, \Pi) - v(\mathbf{n}^l, \Pi) \quad (4.16d)$$

with  $V(\mathbf{n}^l, \Pi, t)$  being the reward expected at  $t \gg t_0$ , if the link was in macrostate  $\mathbf{n}^l$  at  $t_0$ . Furthermore, Equation (4.16d) defines  $g_j^l(\mathbf{n}^l, \Pi)$  as the macrostate-dependent link reward gain which is the long-term

## System of 18 Linear Equations

$$\begin{aligned}
\chi_1: R^l(\Pi_1) &= \lambda_1^l(\chi_1, \Pi_1) \cdot [v(\chi_2, \Pi_1) - v(\chi_1, \Pi_1)] + \lambda_2^l(\chi_1, \Pi_1) \cdot [v(\chi_{14}, \Pi_1) - v(\chi_1, \Pi_1)] \\
\chi_2: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + \lambda_1^l(\chi_2, \Pi_1) \cdot [v(\chi_7, \Pi_1) - v(\chi_2, \Pi_1)] + \lambda_2^l(\chi_2, \Pi_1) \cdot [v(\chi_{18}, \Pi_1) - v(\chi_2, \Pi_1)] + \mu_1 \cdot [v(\chi_1, \Pi_1) - v(\chi_2, \Pi_1)] \\
\chi_3: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + \lambda_1^l(\chi_3, \Pi_1) \cdot [v(\chi_{10}, \Pi_1) - v(\chi_3, \Pi_1)] + \mu_1 \cdot [v(\chi_1, \Pi_1) - v(\chi_3, \Pi_1)] \\
\chi_4: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + \lambda_1^l(\chi_4, \Pi_1) \cdot [v(\chi_7, \Pi_1) - v(\chi_4, \Pi_1)] + \mu_1 \cdot [v(\chi_1, \Pi_1) - v(\chi_4, \Pi_1)] \\
\chi_5: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + \lambda_1^l(\chi_5, \Pi_1) \cdot [v(\chi_8, \Pi_1) - v(\chi_5, \Pi_1)] + \mu_1 \cdot [v(\chi_1, \Pi_1) - v(\chi_5, \Pi_1)] \\
\chi_6: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + \lambda_1^l(\chi_6, \Pi_1) \cdot [v(\chi_9, \Pi_1) - v(\chi_6, \Pi_1)] + \lambda_2^l(\chi_6, \Pi_1) \cdot [v(\chi_{17}, \Pi_1) - v(\chi_6, \Pi_1)] + \mu_1 \cdot [v(\chi_1, \Pi_1) - v(\chi_6, \Pi_1)] \\
\chi_7: R^l(\Pi_1) &= r_1^l \cdot 2\mu_1 + \lambda_1^l(\chi_7, \Pi_1) \cdot [v(\chi_{13}, \Pi_1) - v(\chi_7, \Pi_1)] + \mu_1 \cdot [v(\chi_4, \Pi_1) - v(\chi_7, \Pi_1)] + \mu_1 \cdot [v(\chi_2, \Pi_1) - v(\chi_7, \Pi_1)] \\
\chi_8: R^l(\Pi_1) &= r_1^l \cdot 2\mu_1 + \mu_1 \cdot [v(\chi_5, \Pi_1) - v(\chi_8, \Pi_1)] + \mu_1 \cdot [v(\chi_2, \Pi_1) - v(\chi_8, \Pi_1)] \\
\chi_9: R^l(\Pi_1) &= r_1^l \cdot 2\mu_1 + \lambda_1^l(\chi_9, \Pi_1) \cdot [v(\chi_{13}, \Pi_1) - v(\chi_9, \Pi_1)] + \mu_1 \cdot [v(\chi_6, \Pi_1) - v(\chi_9, \Pi_1)] + \mu_1 \cdot [v(\chi_2, \Pi_1) - v(\chi_9, \Pi_1)] \\
\chi_{10}: R^l(\Pi_1) &= r_1^l \cdot 2\mu_1 + \mu_1 \cdot [v(\chi_5, \Pi_1) - v(\chi_{10}, \Pi_1)] + \mu_1 \cdot [v(\chi_3, \Pi_1) - v(\chi_{10}, \Pi_1)] \\
\chi_{11}: R^l(\Pi_1) &= r_1^l \cdot 2\mu_1 + \mu_1 \cdot [v(\chi_6, \Pi_1) - v(\chi_{11}, \Pi_1)] + \mu_1 \cdot [v(\chi_3, \Pi_1) - v(\chi_{11}, \Pi_1)] \\
\chi_{12}: R^l(\Pi_1) &= r_1^l \cdot 2\mu_1 + \lambda_1^l(\chi_{12}, \Pi_1) \cdot [v(\chi_{13}, \Pi_1) - v(\chi_{12}, \Pi_1)] + \mu_1 \cdot [v(\chi_6, \Pi_1) - v(\chi_{12}, \Pi_1)] + \mu_1 \cdot [v(\chi_4, \Pi_1) - v(\chi_{12}, \Pi_1)] \\
\chi_{13}: R^l(\Pi_1) &= r_1^l \cdot 3\mu_1 + \mu_1 \cdot [v(\chi_{12}, \Pi_1) - v(\chi_{13}, \Pi_1)] + \mu_1 \cdot [v(\chi_9, \Pi_1) - v(\chi_{13}, \Pi_1)] + \mu_1 \cdot [v(\chi_7, \Pi_1) - v(\chi_{13}, \Pi_1)] \\
\chi_{14}: R^l(\Pi_1) &= r_2^l \cdot \mu_2 + \lambda_1^l(\chi_{14}, \Pi_1) \cdot [v(\chi_{17}, \Pi_1) - v(\chi_{14}, \Pi_1)] + \mu_2 \cdot [v(\chi_1, \Pi_1) - v(\chi_{14}, \Pi_1)] \\
\chi_{15}: R^l(\Pi_1) &= r_2^l \cdot \mu_2 + \mu_2 \cdot [v(\chi_1, \Pi_1) - v(\chi_{15}, \Pi_1)] \\
\chi_{16}: R^l(\Pi_1) &= r_2^l \cdot \mu_2 + \lambda_1^l(\chi_{16}, \Pi_1) \cdot [v(\chi_{18}, \Pi_1) - v(\chi_{16}, \Pi_1)] + \mu_2 \cdot [v(\chi_1, \Pi_1) - v(\chi_{16}, \Pi_1)] \\
\chi_{17}: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + r_2^l \cdot \mu_2 + \mu_1 \cdot [v(\chi_{14}, \Pi_1) - v(\chi_{17}, \Pi_1)] + \mu_2 \cdot [v(\chi_6, \Pi_1) - v(\chi_{17}, \Pi_1)] \\
\chi_{18}: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + r_2^l \cdot \mu_2 + \mu_1 \cdot [v(\chi_{16}, \Pi_1) - v(\chi_{18}, \Pi_1)] + \mu_2 \cdot [v(\chi_2, \Pi_1) - v(\chi_{18}, \Pi_1)]
\end{aligned}$$

## Remarks:

- To simplify the mathematical notation in the equations, the 18 link states  $\mathbf{x}^l$  are denoted as  $\chi_i$ ,  $i = 1, \dots, 18$
- The state-dependent link arrival rates  $\lambda_j^l(\mathbf{x}^l, \Pi_1)$ , the termination rates  $\mu_j$  and the link reward parameters  $r_j^l$  are known
- The state-dependent link arrival rates  $\lambda_j^l(\mathbf{x}^l, \Pi_1)$  are estimated from online measurements taken during network operation
- The linear system must be solved for the link reward rate  $R^l(\Pi_1)$  and the 18 transient reward values  $v(\mathbf{x}^l, \Pi_1)$

Figure 4.4: System of linear equations in the state-dependent model for the link in Fig. 4.3.

reward that a class- $j$  connection brings to the link if it gets admission in macrostate  $\mathbf{n}^l$ . From this gain, we approximate the state-dependent network reward gain  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$  as:

$$g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi) \quad (4.16e)$$

i.e. to calculate  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$ , it is only necessary to know the macrostate-dependent gains of the links in the path  $\rho$  used by the connection.

Equation (4.16a) defines a linear system with  $|\Omega_n^l|$  equations. For all macrostates  $\mathbf{n}^l$ , the parameters  $q(\mathbf{n}^l)$ ,  $\lambda_j^l(\mathbf{n}^l, \Pi)$  and  $\mu_j$  are known. Therefore, under the macrostate-dependent description of  $R^l(\Pi)$ , the calculation of the network reward rate  $R(\Pi)$  is performed as follows. First, the parameters  $(o, d)_j$ ,  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$  are defined along with the policy  $\Pi$ . Second, for each link, these parameters and the link capacity  $C_l$  are used to calculate the macrostate-space  $\Omega_n^l$  (every macrostate must fulfil the capacity constraint  $\sum_{j=1}^J b_j \cdot n_j^l \leq C_l$ ). Third, for every link, the linear system is defined from Equation (4.16a) based on the decisions of the policy  $\Pi$ . Then each system is solved for  $R^l(\Pi)$  and the values  $v(\mathbf{n}^l, \Pi)$ . Finally, Equation (4.6) is used to calculate  $R(\Pi)$  as the sum of the link rates  $R^l(\Pi)$ .

**Example 4.5** For the flex-grid link in Fig. 4.3, the policy  $\Pi_1$  uses first-fit (FF) as spectrum allocation policy [TAK<sup>+</sup>14]. In an ordered spectrum grid with  $C_l$  slots indexed as  $1, 2, \dots, C_l$ , first-fit assigns the available  $b_j$  adjacent slots with the lowest indices in the grid. Consider the link state  $\chi_4 = (0, 0, 1, \infty, 0, 0)$ . The policy decision  $\Pi_1(\chi_4, 1)$  says that if a class-1 arrival occurs in state  $\chi_4$ , it must be admitted by allocating the slots 1-2, thereby causing the transition  $\chi_4 \rightarrow \chi_7 = (1, \infty, 1, \infty, 0, 0)$ . In this case, the slots 1-2 are the two available adjacent slots with the lowest indices in the grid. For this spectrum allocation policy, we present in Fig. 4.4 the system of 18 linear equations defined by Equation (4.14a). Let us check the equation for  $\chi_4$ . In this state, the link only carries a class-1 connection, i.e.  $\mathbf{n}^l = (1, 0)$ . Then the link earns reward at a rate  $q(\chi_4) = r_1^l \cdot \mu_1$  (ru/ot). Only two transitions are possible in  $\chi_4$ , namely,  $\chi_4 \rightarrow \chi_{11}$  which occurs when the carried connection departs, and  $\chi_4 \rightarrow \chi_7$ , due to a class-1 arrival. As seen in Fig.

### Definition of Macrostates

The macrostate-space is  $\Omega_{\mathbf{n}}^l = \{(0,0), (1,0), (2,0), (3,0), (0,1), (1,1)\}$ , i.e. there are six macrostates. They are denoted in the linear system as:  $\mathbf{n}_1 = (0,0)$ ,  $\mathbf{n}_2 = (1,0)$ ,  $\mathbf{n}_3 = (2,0)$ ,  $\mathbf{n}_4 = (3,0)$ ,  $\mathbf{n}_5 = (0,1)$ ,  $\mathbf{n}_6 = (1,1)$

### System of 6 Linear Equations

$$\begin{aligned} \mathbf{n}_1: R^l(\Pi_1) &= \lambda_1^l(\mathbf{n}_1, \Pi_1) \cdot [v(\mathbf{n}_2, \Pi_1) - v(\mathbf{n}_1, \Pi_1)] + \lambda_2^l(\mathbf{n}_1, \Pi_1) \cdot [v(\mathbf{n}_5, \Pi_1) - v(\mathbf{n}_1, \Pi_1)] \\ \mathbf{n}_2: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + \lambda_1^l(\mathbf{n}_2, \Pi_1) \cdot [v(\mathbf{n}_3, \Pi_1) - v(\mathbf{n}_2, \Pi_1)] + \lambda_2^l(\mathbf{n}_2, \Pi_1) \cdot [v(\mathbf{n}_6, \Pi_1) - v(\mathbf{n}_2, \Pi_1)] + \mu_1 \cdot [v(\mathbf{n}_1, \Pi_1) - v(\mathbf{n}_2, \Pi_1)] \\ \mathbf{n}_3: R^l(\Pi_1) &= 2 \cdot r_1^l \cdot \mu_1 + \lambda_1^l(\mathbf{n}_3, \Pi_1) \cdot [v(\mathbf{n}_4, \Pi_1) - v(\mathbf{n}_3, \Pi_1)] + 2 \cdot \mu_1 \cdot [v(\mathbf{n}_2, \Pi_1) - v(\mathbf{n}_3, \Pi_1)] \\ \mathbf{n}_4: R^l(\Pi_1) &= 3 \cdot r_1^l \cdot \mu_1 + 3 \cdot \mu_1 \cdot [v(\mathbf{n}_3, \Pi_1) - v(\mathbf{n}_4, \Pi_1)] \\ \mathbf{n}_5: R^l(\Pi_1) &= r_2^l \cdot \mu_2 + \lambda_1^l(\mathbf{n}_5, \Pi_1) \cdot [v(\mathbf{n}_6, \Pi_1) - v(\mathbf{n}_5, \Pi_1)] + \mu_2 \cdot [v(\mathbf{n}_1, \Pi_1) - v(\mathbf{n}_5, \Pi_1)] \\ \mathbf{n}_6: R^l(\Pi_1) &= r_1^l \cdot \mu_1 + r_2^l \cdot \mu_2 + \mu_1 \cdot [v(\mathbf{n}_5, \Pi_1) - v(\mathbf{n}_6, \Pi_1)] + \mu_2 \cdot [v(\mathbf{n}_2, \Pi_1) - v(\mathbf{n}_6, \Pi_1)] \end{aligned}$$

#### Remarks:

- To simplify the mathematical notation in the equations, the six link macrostates  $\mathbf{n}^l$  are denoted as  $\mathbf{n}_i$ ,  $i = 1, \dots, 6$
- The macrostate-dependent link arrival rates  $\lambda_j^l(\mathbf{n}^l, \Pi_1)$ , the termination rates  $\mu_j$  and the link reward parameters  $r_j^l$  are known
- The macrostate-dependent link arrival rates  $\lambda_j^l(\mathbf{n}^l, \Pi_1)$  are estimated from online measurements taken during network operation
- The linear system must be solved for the link reward rate  $R^l(\Pi_1)$  and the six transient reward values  $v(\mathbf{n}_i, \Pi_1)$

**Figure 4.5:** System of linear equations in the macrostate-dependent model for the link in Fig. 4.3.

4.3, class-2 arrivals are rejected (i.e.  $\Pi_1(\mathcal{X}_4, 2) = \text{Do not admit}$ ) as in  $\mathcal{X}_4$  there are no four available slots that fulfil the contiguity constraint. Since only two transitions are possible, in Equation (4.14a) we have that  $\Gamma_{\mathcal{X}_4}^{1+} = \{\mathcal{X}_7\}$  and  $\Gamma_{\mathcal{X}_4}^{2+} = \{\phi\}$  for the first double summation. Similarly,  $\Gamma_{\mathcal{X}_4}^{1-} = \{\mathcal{X}_1\}$  and  $\Gamma_{\mathcal{X}_4}^{2-} = \{\phi\}$  for the second one. Hence, for  $\mathcal{X}_4$ , we have that  $R^l(\Pi_1) = r_1^l \cdot \mu_1 + \lambda_1^l(\mathcal{X}_4, \Pi_1) \cdot [v(\mathcal{X}_7, \Pi_1) - v(\mathcal{X}_4, \Pi_1)] + \mu_1 \cdot [v(\mathcal{X}_1, \Pi_1) - v(\mathcal{X}_4, \Pi_1)]$ . To calculate  $R^l(\Pi_1)$ , the linear system in Fig. 4.4 is solved for  $R^l(\Pi_1)$  and the 18 values  $v(\mathcal{X}_i, \Pi_1)$ .

**Example 4.6** For the flex-grid link in Fig. 4.3, the macrostate-dependent reward model is defined by a linear system with six equations. That system is defined in Fig. 4.5, where the macrostates are indexed as  $\mathbf{n}_i$ ,  $i = 1, \dots, 6$ . For each macrostate  $\mathbf{n}_i$ , the corresponding equation describes the effect that the states in  $\widehat{\mathcal{X}}_{\mathbf{n}_i}^l$  have on  $R^l(\Pi_1)$ . To illustrate this, let us consider the macrostate  $\mathbf{n}_2 = (1,0)$ . In the state-dependent model, the five states that configure this macrostate (i.e. those in  $\widehat{\mathcal{X}}_{(1,0)}^l = \{\mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4, \mathcal{X}_5, \mathcal{X}_6\}$ ) define five equations in Fig. 4.4. In the macrostate-dependent model, these five equations are simplified by the equation for  $\mathbf{n}_2$  shown in Fig. 4.5, namely  $R^l(\Pi_1) = r_1^l \cdot \mu_1 + \lambda_1^l(\mathbf{n}_2, \Pi_1) \cdot [v(\mathbf{n}_3, \Pi_1) - v(\mathbf{n}_2, \Pi_1)] + \lambda_2^l(\mathbf{n}_2, \Pi_1) \cdot [v(\mathbf{n}_6, \Pi_1) - v(\mathbf{n}_2, \Pi_1)] + \mu_1 \cdot [v(\mathbf{n}_1, \Pi_1) - v(\mathbf{n}_2, \Pi_1)]$ . Observe that the five equations in Fig. 4.4, for the states in  $\widehat{\mathcal{X}}_{(1,0)}^l$ , define  $q(\mathcal{X}_i) = r_1^l \cdot \mu_1$  which coincides with  $q(\mathbf{n}_2) = r_1^l \cdot \mu_1$ . Furthermore, notice that the simplified equation is a linear function of the transitions  $\mathbf{n}_2 \rightarrow \mathbf{n}_2 \pm \delta_j^l$ . These transitions were defined in Example 4.4, and they are:  $\mathbf{n}_2 = (1,0) \rightarrow \mathbf{n}_3 = (2,0)$ ,  $\mathbf{n}_2 = (1,0) \rightarrow \mathbf{n}_6 = (1,1)$  and  $\mathbf{n}_2 = (1,0) \rightarrow \mathbf{n}_1 = (0,0)$ . From this, the reward change caused by the transition  $\mathbf{n}_2 \rightarrow \mathbf{n}_3$ , i.e. by the admission of a class-1 connection, is given in the macrostate equation by  $v(\mathbf{n}_3, \Pi_1) - v(\mathbf{n}_2, \Pi_1)$ . This reward change can be caused by any of the five state transitions studied in Example 4.4, namely,  $\mathcal{X}_2 \rightarrow \mathcal{X}_7$ ,  $\mathcal{X}_3 \rightarrow \mathcal{X}_{10}$ ,  $\mathcal{X}_4 \rightarrow \mathcal{X}_7$ ,  $\mathcal{X}_5 \rightarrow \mathcal{X}_8$  and  $\mathcal{X}_6 \rightarrow \mathcal{X}_9$ . These transitions originate from class-1 requests admitted when the link is in macrostate  $\mathbf{n}_2 = (1,0)$ . A similar interpretation follows for the reward changes  $v(\mathbf{n}_6, \Pi_1) - v(\mathbf{n}_2, \Pi_1)$  and  $v(\mathbf{n}_1, \Pi_1) - v(\mathbf{n}_2, \Pi_1)$ . With the macrostate-dependent description, the link reward rate  $R^l(\Pi_1)$  is calculated by solving the linear system in Fig. 4.5 for  $R^l(\Pi_1)$  and the six values  $v(\mathbf{n}_i, \Pi_1)$ . This example shows the reduction that the macrostate-dependent description has on the size of the linear system: from 18 equations in the state-dependent model to six equations defined in the set of macrostates  $\Omega_{\mathbf{n}}^l$ .

## 4.2.1 Properties of the Macrostate-Dependent Reward Model

The macrostate-dependent model is an approximation to the state-dependent description. The accuracy of the approximation depends on the estimation of the rates  $\lambda_j^l(\mathbf{n}^l, \Pi)$ , as they determine the transition rates between the macrostates in  $\Omega_{\mathbf{n}}^l$ . This issue will be studied in Section 4.3, where it will be shown that  $\lambda_j^l(\mathbf{n}^l, \Pi)$  implicitly quantifies the effect that the contiguity constraint has on the process  $\{\mathbf{N}_t^{l,\pi}, t \geq 0\}$ . This process defines the following properties for the macrostate-dependent model described by Equations (4.16):

1. Every macrostate  $\mathbf{n}^l \in \Omega_{\mathbf{n}}^l$  is defined by a sub-set of link states  $\mathbf{x}^l \in \widehat{\mathcal{X}}_{\mathbf{n}}^l$  that fulfil the contiguity and capacity constraints.

2. For all  $\mathbf{x}^l \in \hat{\mathcal{X}}_{\mathbf{n}}^l$ , it cannot be stated that the equality  $v(\mathbf{x}^l, \Pi) = v(\mathbf{n}^l, \Pi)$  always holds. Although the states in  $\hat{\mathcal{X}}_{\mathbf{n}}^l$  represent the same traffic load  $\mathbf{n}^l$ , they define different spectrum configurations, and thus, they may have different effects on the link reward process. Therefore, for any  $\mathbf{x}^l, \mathbf{z}^l \in \hat{\mathcal{X}}_{\mathbf{n}}^l$ , it is possible that  $v(\mathbf{x}^l, \Pi) \neq v(\mathbf{z}^l, \Pi)$ , and thereby  $v(\mathbf{n}^l, \Pi) \neq v(\mathbf{x}^l, \Pi) \neq v(\mathbf{z}^l, \Pi)$ .
3. The purpose of the macrostate-dependent reward model is to circumvent the complexity of the state-dependent description. This is accomplished by calculating a unique value  $v(\mathbf{n}^l, \Pi)$  that represents all the states in  $\hat{\mathcal{X}}_{\mathbf{n}}^l$ . The advantage of this strategy is that it makes the linear system in Equation (4.16a) solvable. The drawback is that from the solution to the system, the values  $v(\mathbf{x}^l, \Pi)$  of the states in  $\hat{\mathcal{X}}_{\mathbf{n}}^l$  remain unknown. The transient values  $v(\mathbf{n}^l, \Pi)$  are then used as an approximation to the values  $v(\mathbf{x}^l, \Pi)$  of the states in  $\hat{\mathcal{X}}_{\mathbf{n}}^l$ .
4. If a class- $j$  connection is admitted to link  $l$  in state  $\mathbf{x}^l$ , such that it causes the transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , it brings to the link a short-term reward  $r_j^l$  and a long-term reward  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) = v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi)$ . Under the macrostate-dependent reward model, the actual reward  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$  cannot be calculated, as the values  $v(\mathbf{x}^l, \Pi)$  and  $v(\mathbf{y}^l, \Pi)$  are unknown. Instead, it is approximated as  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) \approx g_j^l(\mathbf{n}^l, \Pi) = v(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi) - v(\mathbf{n}^l, \Pi)$ . Therefore, the assumption is made that all the possible link state transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  that cause the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  yield, on average, the same long-term reward gain  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) \approx g_j^l(\mathbf{n}^l, \Pi)$ .
5. From the link independence assumption, it follows that the state-dependent network reward gain now approximates as  $g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi)$ . It is interpreted as follows: if at time  $t_0$  the network is in state  $\mathbf{x}$  and a class- $j$  connection is admitted to a path  $\rho \in \Gamma_j$ , it causes the transition  $\mathbf{x} \rightarrow \mathbf{y}$ , which implies that every link  $l$  in  $\rho$  undergoes a macrostate transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  that is implicitly caused by a link state move  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ . As a result, the network earns an immediate (or short-term) reward  $r_j$ , from which  $r_j^l$  (ru) are earned through the link  $l$  in the path. Besides  $r_j$ , the connection brings a long-term reward  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$ , which is the sum of the long-term reward contributions on each link in  $\rho$ . As with the approximate state-dependent model, it is assumed that a class- $j$  connection only affects the network reward through the links of the path on which it is routed.

## 4.2.2 The Value Iteration Algorithm

As with the exact and the approximate state-dependent network reward models, the linear system defined by Equation (4.16a) can be solved by setting one of the transient values  $v(\mathbf{n}^l, \Pi)$  to zero. Then a system with  $|\Omega_{\mathbf{n}}^l|$  equations and  $|\Omega_{\mathbf{n}}^l|$  unknowns is obtained that can be solved by standard linear algebra methods. An alternative solution method, known as the value iteration algorithm (VIA), has successfully been applied to reward-based routing in telephone and packet-switched systems [DM92, DM94, Dzi97]. The VIA and its properties are thoroughly studied in [Tij86], and a comprehensive formulation of this method for Markov decision processes is presented in [Dzi97] - appendix B. The idea behind the VIA is to avoid a direct solution of the linear system by using a numerical iteration procedure. The procedure reduces the computation time, thereby facilitating the implementation of the macrostate-dependent model for online resource allocation. To apply the VIA, the first step is to discretize Equation (4.16a). For this, observe that from Equation (4.16c) we have:

$$V(\mathbf{n}^l, \Pi, t) - V(\mathbf{n}^l, \Pi, t - 1) = R^l(\Pi) \quad , \mathbf{n}^l \in \Omega_{\mathbf{n}}^l \quad (4.17)$$

Similarly, from Equation (4.16d):

$$V(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi, t - 1) - V(\mathbf{n}^l, \Pi, t - 1) = v(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi) - v(\mathbf{n}^l, \Pi) \quad (4.18)$$

Therefore, Equation (4.16a) can be re-written from these two equations as:

$$V(\mathbf{n}^l, \Pi, t) - V(\mathbf{n}^l, \Pi, t - 1) = q(\mathbf{n}^l) + \sum_{j=1}^J \lambda_j^l(\mathbf{n}^l, \Pi) \cdot [V(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi, t - 1) - V(\mathbf{n}^l, \Pi, t - 1)] + \sum_{j=1}^J n_j^l \cdot \mu_j \cdot [V(\mathbf{n}^l - \boldsymbol{\delta}_j^l, \Pi, t - 1) - V(\mathbf{n}^l, \Pi, t - 1)], \mathbf{n}^l \in \Omega_{\mathbf{n}}^l \quad (4.19)$$



The discretization of Equation (4.19) is achieved by applying the uniformization technique explained in [Tij86], which consists in multiplying both sides of the equation by the minimum average sojourn time  $\tau$  attainable by a macrostate in  $\Omega_n^l$ . This sojourn time can be calculated as:

$$\tau = \frac{1}{\sum_{j=1}^J \lambda_j^l(\Pi) + \max_{\mathbf{n}^l \in \Omega_n^l} \sum_{j=1}^J n_j^l \cdot \mu_j} \quad (4.20)$$

With the uniformization technique, the continuous-time stochastic process  $\{\mathbf{N}_t^{l,\pi}, t \geq 0\}$  is modelled by an equivalent discrete-time process  $\{\mathbf{N}_k^{l,\pi}, k = \mathbb{Z}^+\}$ . For this process, let  $V(\mathbf{n}^l, \Pi, k)$  be the discrete version of  $V(\mathbf{n}^l, \Pi, t)$ . Then  $V(\mathbf{n}^l, \Pi, k)$  is interpreted as the mean reward received by the link after  $k$  macrostate transitions with  $\mathbf{n}^l$  as starting macrostate. Therefore, the discrete version of Equation (4.19) reads as follows:

$$\begin{aligned} V(\mathbf{n}^l, \Pi, k) - V(\mathbf{n}^l, \Pi, k-1) &= q(\mathbf{n}^l) \cdot \tau + \\ &\sum_{j=1}^J \lambda_j^l(\mathbf{n}^l, \Pi) \cdot \tau \cdot [V(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi, k-1) - V(\mathbf{n}^l, \Pi, k-1)] + \\ &\sum_{j=1}^J n_j^l \cdot \mu_j \cdot \tau \cdot [V(\mathbf{n}^l - \boldsymbol{\delta}_j^l, \Pi, k-1) - V(\mathbf{n}^l, \Pi, k-1)], \mathbf{n}^l \in \Omega_n^l \end{aligned} \quad (4.21)$$

which defines a recurrence relation, with  $k$  being the iteration index. Notice that from Equation (4.17) [Dzi97]:

$$\lim_{k \rightarrow \infty} [V(\mathbf{n}^l, \Pi, k) - V(\mathbf{n}^l, \Pi, k-1)] \rightarrow R^l(\Pi) \cdot \tau \quad (4.22)$$

The reward  $V(\mathbf{n}^l, \Pi, k)$  is then calculated from Equation (4.21) as:

$$\begin{aligned} V(\mathbf{n}^l, \Pi, k) &= q(\mathbf{n}^l) \cdot \tau + \sum_{j=1}^J \lambda_j^l(\mathbf{n}^l, \Pi) \cdot \tau \cdot [V(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi, k-1) - V(\mathbf{n}^l, \Pi, k-1)] + \\ &\sum_{j=1}^J n_j^l \cdot \mu_j \cdot \tau \cdot [V(\mathbf{n}^l - \boldsymbol{\delta}_j^l, \Pi, k-1) - V(\mathbf{n}^l, \Pi, k-1)] + V(\mathbf{n}^l, \Pi, k-1), \mathbf{n}^l \in \Omega_n^l \end{aligned} \quad (4.23)$$

Based on this, for a link  $l$ , the VIA is implemented to solve the linear system for  $R^l(\Pi)$  and the transient reward values  $v(\mathbf{n}^l, \Pi)$  as follows [RB16b]:

1. Define the parameter  $\epsilon \geq 0$  as a scalar that denotes the relative accuracy with which  $R^l(\Pi)$  and the values  $v(\mathbf{n}^l, \Pi)$  need to be estimated.
2. Set  $k = 0$ . For each macrostate  $\mathbf{n}^l \in \Omega_n^l$  assign an arbitrary value to the reward  $V(\mathbf{n}^l, \Pi, 0)$ .
3. Set  $k = k + 1$ . For each macrostate  $\mathbf{n}^l$ , determine  $V(\mathbf{n}^l, \Pi, k)$  by calculating the right hand side of Equation (4.23).
4. With the results obtained in the previous step, calculate the upper estimate of  $R^l(\Pi) \cdot \tau$  as:

$$M_k = \max_{\mathbf{n}^l \in \Omega_n^l} \{V(\mathbf{n}^l, \Pi, k) - V(\mathbf{n}^l, \Pi, k-1)\} \quad (4.24)$$

and the lower estimate of  $R^l(\Pi) \cdot \tau$  as:

$$m_k = \min_{\mathbf{n}^l \in \Omega_n^l} \{V(\mathbf{n}^l, \Pi, k) - V(\mathbf{n}^l, \Pi, k-1)\} \quad (4.25)$$

5. If the upper and lower estimates obtained for  $R^l(\Pi) \cdot \tau$  fulfil the inequality:

$$(M_k - m_k)/m_k \leq \epsilon \quad (4.26)$$

the iteration process is stopped. In this case, the approximate solution is:

$$R^l(\Pi) \cdot \tau \cong \frac{M_k + m_k}{2} \quad (4.27)$$

and the transient reward values are calculated from:

$$v(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi) - v(\mathbf{n}^l, \Pi) \cong V(\mathbf{n}^l + \boldsymbol{\delta}_j^l, \Pi, k) - V(\mathbf{n}^l, \Pi, k) \quad (4.28)$$

If the inequality is not fulfilled, then go back to step tree, i.e. start a new iteration, and continue until  $(M_k - m_k)/m_k \leq \epsilon$ .

The VIA calculates an approximate solution for the linear system whose accuracy depends on the parameter  $\epsilon$ . Independently of the reward values  $V(\mathbf{n}^l, \Pi, 0)$  used for  $k = 0$ , the VIA always converges, which is a consequence of the existence of the limit in Equation (4.22). The computational complexity of the VIA per iteration is quadratic in the number of macrostates, i.e.  $O(\propto |\Omega_{\mathbf{n}}^l|^2)$ . Moreover, the number of iterations required is polynomial in  $|\Omega_{\mathbf{n}}^l|$  (see [Tij86]). The VIA avoids the solution of the linear system by linear algebra methods which may be too slow under an online resource allocation scenario.

### 4.2.3 Simplified Macrostate-Dependent Link Model

The cardinality of  $\Omega_{\mathbf{n}}^l$  grows with the size (i.e. the number of components) of the link macrostates. That size equals the number of classes  $J$  served by the network. The components  $n_j^l$  of  $\mathbf{n}^l = (n_1^l, \dots, n_j^l, \dots, n_J^l)$  have two properties. First,  $n_j^l = 0$  for all classes which can never be carried by the link, i.e. for all  $j$  such that  $j \notin J^l$ . Otherwise,  $n_j^l \geq 0$ . Secondly, there can be two or more classes in  $J^l$  with the same bandwidth  $b_j$  and holding time  $\mu_j^{-1}$ . These two properties can be used to define a new macrostate-space  $\Omega_{\hat{\mathbf{n}}}^l$  with macrostates of the form  $\hat{\mathbf{n}}^l = (\hat{n}_1^l, \dots, \hat{n}_k^l, \dots, \hat{n}_K^l)$ , such that  $K < J$ , which implies  $|\Omega_{\hat{\mathbf{n}}}^l| < |\Omega_{\mathbf{n}}^l|$ . This is accomplished in two steps. First, the classes  $j \notin J^l$  are not included in  $\hat{\mathbf{n}}^l$  as they play no role in the link reward process. Secondly, the classes with the same bandwidth  $b_j$  and holding time  $\mu_j^{-1}$  are aggregated into a new class  $k$ , such that  $b_k = b_j$  and  $\mu_k^{-1} = \mu_j^{-1}$ . The new equivalent class- $k$  has then a macrostate-dependent arrival rate [RB16d, RB17a]:

$$\lambda_k^l(\hat{\mathbf{n}}^l, \Pi) = \sum_{j \in \Lambda_k} \lambda_j^l(\mathbf{n}^l, \Pi) \quad (4.29)$$

where  $\Lambda_k$  is the set of classes with the same bandwidth and holding time, which are represented by the equivalent class  $k$ . This class has a reward parameter  $r_k^l$  calculated as a weighted average over the carried traffics  $A_{c_j}^l, j \in \Lambda_k$ :

$$r_k^l = \frac{\sum_{j \in \Lambda_k} r_j^l \cdot A_{c_j}^l}{\sum_{j \in \Lambda_k} A_{c_j}^l} \quad (4.30)$$

With the new macrostate-space  $\Omega_{\hat{\mathbf{n}}}^l$ , Equation (4.16a) is redefined as follows:

$$R^l(\Pi) = q(\hat{\mathbf{n}}^l) + \sum_{k=1}^K \lambda_k^l(\hat{\mathbf{n}}^l, \Pi) \cdot [v(\hat{\mathbf{n}}^l + \boldsymbol{\delta}_k^l, \Pi) - v(\hat{\mathbf{n}}^l, \Pi)] + \sum_{k=1}^K \hat{n}_k^l \cdot \mu_k \cdot [v(\hat{\mathbf{n}}^l - \boldsymbol{\delta}_k^l, \Pi) - v(\hat{\mathbf{n}}^l, \Pi)] \quad , \hat{\mathbf{n}}^l \in \Omega_{\hat{\mathbf{n}}}^l \quad (4.31a)$$

where:

$$q(\hat{\mathbf{n}}^l) = \sum_{k=1}^K r_k^l \cdot \hat{n}_k^l \cdot \mu_j \quad (4.31b)$$

The equivalent linear system defined by Equation (4.31a) can be solved by the VIA for  $R^l(\Pi)$  and the values  $v(\hat{\mathbf{n}}^l, \Pi)$ . These values define the equivalent gains (or long-term rewards):

$$g_k^l(\hat{\mathbf{n}}^l, \Pi) = v(\hat{\mathbf{n}}^l + \boldsymbol{\delta}_k^l, \Pi) - v(\hat{\mathbf{n}}^l, \Pi) \quad (4.31c)$$

which have the same interpretation as  $g_j^l(\mathbf{n}^l, \Pi)$ , i.e. the gain  $g_k^l(\hat{\mathbf{n}}^l, \Pi)$  is the long-term reward that an admitted connection of class  $j \in \Lambda_k$  brings to the link when it arrives in macrostate  $\hat{\mathbf{n}}^l$ .

**Example 4.7** Consider the network in Fig. 4.1. For each link  $l$ , the macrostate  $\mathbf{n}^l$  has size 12, as the network serves 12 connection classes. If all links have the same capacity  $C_l = 6$ , we have  $|\Omega_{\mathbf{n}}^l| > 100$ . From the parameters shown in Fig. 4.1, on each link  $l$ , classes 1-6 can be aggregated into an equivalent class  $x$  with  $b_x = 2$  and  $\mu_x^{-1} = 1$ , where  $\Lambda_x = \{1,2,3,4,5,6\}$ . Similarly, classes 7-12 are aggregated into

a class  $y$  with  $b_y = 4$  and  $\mu_y^{-1} = 1$ , where  $\Lambda_y = \{7,8,9,10,11,12\}$ . Each link is now described by the macrostate  $\hat{\mathbf{n}}^l = (\hat{n}_x^l, \hat{n}_y^l)$ . Under this equivalent description, a link  $l$  has the state and macrostate-spaces shown in Fig. 4.3. The links are then modelled by systems of  $|\Omega_{\hat{\mathbf{n}}}^l| = 6$  linear equations from which  $R^l(\Pi)$  and the values  $v(\hat{\mathbf{n}}^l, \Pi)$  are calculated. If the network allocates resources with the policy  $\Pi_1$  in Fig. 4.3, we have that Equation (4.31a) defines the linear system shown in Fig. 4.5 (where the macrostates  $\mathbf{n}_i$  represent the macrostates in  $\Omega_{\hat{\mathbf{n}}}^l$ ). By solving the system, the gains  $g_x^l(\hat{\mathbf{n}}^l, \Pi)$  and  $g_y^l(\hat{\mathbf{n}}^l, \Pi)$  are calculated from Equation (4.31c). The reward gain  $g_x^l(\hat{\mathbf{n}}^l, \Pi)$  is interpreted as the long-term reward that an admitted connection of class  $j \in \Lambda_x = \{1,2,3,4,5,6\}$  brings to the link when it arrives in macrostate  $\hat{\mathbf{n}}^l$ .

The class aggregation procedure that we have outlined was originally proposed in [DM89] and was further investigated in [DM92, DM94] for reward-based routing in high-speed data networks. In particular, [DM92] presents results (based on a detailed performance analysis) that show that the class aggregation method simplifies computational complexity without degrading the performance. This renders the method suitable for the reduction of the cardinality  $|\Omega_{\hat{\mathbf{n}}}^l|$  in optical networks that serve a large number of connection classes.

### 4.3. Online Estimation of Link Arrival Rates

The exogenous arrival rate  $\lambda_j$  and the policy  $\Pi$  define the link arrival rates  $\lambda_j^l(\Pi)$ ,  $\lambda_j^l(\mathbf{x}^l, \Pi)$  and  $\lambda_j^l(\mathbf{n}^l, \Pi)$ . An accurate estimation of these rates is crucial for the implementation of either the state or the macrostate dependent reward models. Instead of using analytical methods to calculate the arrival rates, we propose an online estimation procedure that differs from existing methods, e.g. [DM89, DM92, DM94, Kri91, Dzi97], in that the effect of the contiguity constraint is considered. An online estimation is advantageous as it confers the PIA the ability of adapting the policy decisions to changing traffic conditions.

#### 4.3.1 The Policy and Macrostate-Dependent Link Arrival Rates

Consider an optical network in state  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^l, \dots, \mathbf{x}^L)$ , where the link  $l$  is in state  $\mathbf{x}^l$ . The transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  may originate from a class- $j$  arrival if the policy decision  $\Pi(\mathbf{x}, j)$  admits the connection on a lightpath that causes the transition  $\mathbf{x} \rightarrow \mathbf{y}$ , with  $\mathbf{y} = (\mathbf{y}^1, \dots, \mathbf{y}^l, \dots, \mathbf{y}^L)$ . (Note that this lightpath must be routed on a path  $\rho$ , such that  $l \in \rho$ .) In that case, the transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  has a rate  $\lambda_j^l(\mathbf{x}^l, \Pi)$  equal to  $\lambda_j^l(\Pi)$ , i.e. the rate at which class- $j$  connections arrive at the link. Otherwise, i.e. if  $\Pi(\mathbf{x}, j)$  rejects class- $j$  arrivals, the rate is zero. Therefore, we have:

$$\lambda_j^l(\mathbf{x}^l, \Pi) = \begin{cases} \lambda_j^l(\Pi), & \text{if the policy decision } \Pi(\mathbf{x}, j) \text{ causes the transition } \mathbf{x}^l \rightarrow \mathbf{y}^l \\ 0, & \text{if the policy decision } \Pi(\mathbf{x}, j) \text{ rejects class } j \text{ requests} \end{cases} \quad (4.32)$$

**Example 4.8** Consider the single-link network in Fig. 4.3. Class-1 and class-2 connections arrive at rates  $\lambda_1^l(\Pi_1)$  and  $\lambda_2^l(\Pi_1)$ , respectively. The rates  $\lambda_j^l(\boldsymbol{\chi}_i, \Pi_1)$  are derived from Equation (4.32) by checking the transitions that class- $j$  arrivals cause in every link state  $\boldsymbol{\chi}_i$ . Observe that those transitions are given by the policy decisions  $\Pi_1(\boldsymbol{\chi}_i, 1)$  and  $\Pi_1(\boldsymbol{\chi}_i, 2)$ . As an example, for  $\boldsymbol{\chi}_2$ , we have that  $\lambda_1^l(\boldsymbol{\chi}_2, \Pi_1) = \lambda_1^l(\Pi_1)$  and  $\lambda_2^l(\boldsymbol{\chi}_2, \Pi_1) = \lambda_2^l(\Pi_1)$ , as both connection classes can be admitted in this state. Then  $\lambda_1^l(\boldsymbol{\chi}_2, \Pi_1)$  and  $\lambda_2^l(\boldsymbol{\chi}_2, \Pi_1)$  are, respectively, the rates of the transitions  $\boldsymbol{\chi}_2 \rightarrow \boldsymbol{\chi}_7$  and  $\boldsymbol{\chi}_2 \rightarrow \boldsymbol{\chi}_{18}$ . (Both states  $\boldsymbol{\chi}_7$  and  $\boldsymbol{\chi}_{18}$  are implicitly defined by the policy decisions in Fig. 4.3.) An interesting case occurs for  $\boldsymbol{\chi}_3$ , where  $\lambda_1^l(\boldsymbol{\chi}_3, \Pi_1) = \lambda_1^l(\Pi_1)$  and  $\lambda_2^l(\boldsymbol{\chi}_3, \Pi_1) = 0$ . Since  $\boldsymbol{\chi}_3 = (0,1,\infty,0,0,0)$ , then  $\lambda_2^l(\boldsymbol{\chi}_3, \Pi_1)$  must be zero, as the admission of a class-2 connection would violate the spectrum contiguity constraint (i.e. in this state there are no four adjacent slots for incoming class-2 traffic).

The rate  $\lambda_j^l(\mathbf{n}^l, \Pi)$  is the average rate at which link  $l$  jumps from the macrostate  $\mathbf{n}^l$  to  $\mathbf{n}^l + \boldsymbol{\delta}_j^l$ . While in macrostate  $\mathbf{n}^l$ , the actual rate depends on the link state  $\mathbf{x}^l \in \hat{\mathcal{X}}_{\mathbf{n}}^l$  that configures  $\mathbf{n}^l$ . There can be states in  $\hat{\mathcal{X}}_{\mathbf{n}}^l$  that block class- $j$  arrivals. For those states, the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  is forbidden and thus, the rate is zero. Otherwise, the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  occurs at the rate  $\lambda_j^l(\mathbf{x}^l, \Pi)$  of the state  $\mathbf{x}^l$  that does not block class- $j$  traffic. Thus, let  $P_{\mathbf{n},j}^l$  be the probability of observing the macrostate  $\mathbf{n}^l$  defined by a non-blocking state  $\mathbf{x}^l$  in  $\hat{\mathcal{X}}_{\mathbf{n}}^l$  for class- $j$  traffic, then  $\lambda_j^l(\mathbf{n}^l, \Pi)$  is calculated as the weighted average:

$$\lambda_j^l(\mathbf{n}^l, \Pi) = \lambda_j^l(\Pi) \cdot P_{\mathbf{n},j}^l \quad (4.33)$$

which follows from Equation (4.32), as  $\lambda_j^l(\mathbf{x}^l, \Pi) = \lambda_j^l(\Pi)$  for all states  $\mathbf{x}^l \in \hat{\mathcal{X}}_{\mathbf{n}}^l$  that allow the macrostate transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$ . Hence, the macrostate-dependent arrival rate fulfils the inequality:

$$\lambda_j^l(\mathbf{n}^l, \Pi) \leq \lambda_j^l(\Pi) \quad (4.34)$$

The equality is satisfied when all the states in  $\widehat{\mathcal{X}}_{\mathbf{n}}^l$  allow the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$ . In that case it is easy to verify that  $P_{\mathbf{n},j}^l = 1$ .

**Example 4.9** When the single-link network in Fig. 4.3 is in macrostate  $\mathbf{n}^l = (1,0)$ , the rates of the transitions  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_1^l$ , i.e.  $(1,0) \rightarrow (2,0)$ , and  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_2^l$ , i.e.  $(1,0) \rightarrow (1,1)$ , depend on the state  $\boldsymbol{\chi}_i \in \widehat{\mathcal{X}}_{(1,0)}^l = \{\boldsymbol{\chi}_2, \boldsymbol{\chi}_3, \boldsymbol{\chi}_4, \boldsymbol{\chi}_5, \boldsymbol{\chi}_6\}$  that defines  $\mathbf{n}^l = (1,0)$ . According to the policy  $\Pi_1$ , class-1 arrivals are always admitted in all  $\boldsymbol{\chi}_i \in \widehat{\mathcal{X}}_{(1,0)}^l$ , i.e. the five states in  $\widehat{\mathcal{X}}_{(1,0)}^l$  have at least  $b_1 = 2$  free adjacent slots for class-1 traffic. Thus, for each  $\boldsymbol{\chi}_i \in \widehat{\mathcal{X}}_{(1,0)}^l$ , the transition  $(1,0) \rightarrow (2,0)$  has a rate  $\lambda_1^l(\boldsymbol{\chi}_i, \Pi_1) = \lambda_1^l(\Pi_1)$ . By calculating from Equation (4.33) the weighted average over the five states in  $\widehat{\mathcal{X}}_{(1,0)}^l$ , we have that  $\lambda_1^l(\mathbf{n}^l, \Pi_1) = \lambda_1^l(\Pi_1) \cdot P_{(1,0),1}^l = \lambda_1^l(\Pi_1)$ , where  $P_{(1,0),1}^l = 1$ , as the traffic  $(1,0)$  will always be observed in non-blocking states for class-1 traffic. For class-2 arrivals, the transition  $(1,0) \rightarrow (1,1)$  is only possible in states  $\boldsymbol{\chi}_2$  and  $\boldsymbol{\chi}_6$ . These two states define the rates  $\lambda_2^l(\boldsymbol{\chi}_2, \Pi_1) = \lambda_2^l(\Pi_1)$  and  $\lambda_2^l(\boldsymbol{\chi}_6, \Pi_1) = \lambda_2^l(\Pi_1)$ . The remaining three states in  $\widehat{\mathcal{X}}_{(1,0)}^l$ , namely,  $\boldsymbol{\chi}_3, \boldsymbol{\chi}_4, \boldsymbol{\chi}_5$ , block class-2 arrivals (as they do not have four free adjacent slots for class-2 traffic). For these states, the transition  $(1,0) \rightarrow (1,1)$  has rate zero. Therefore,  $\lambda_2^l(\mathbf{n}^l, \Pi_1) = \lambda_2^l(\Pi_1) \cdot P_{(1,0),2}^l$ , where  $P_{(1,0),2}^l$  is the probability of observing  $(1,0)$  defined by the states  $\boldsymbol{\chi}_2$  or  $\boldsymbol{\chi}_6$ .

### 4.3.2 Online Estimation Procedure

From Equations (4.9)-(4.10), it follows that the arrival rate  $\lambda_j^l(\Pi)$  can be calculated as:

$$\lambda_j^l(\Pi) = \frac{\mu_j \cdot A_{c_j}^l}{(1 - B_j^l)} \quad (4.35)$$

Therefore, the policy and macrostate-dependent arrival rate  $\lambda_j^l(\mathbf{n}^l, \Pi)$  given by Equation (4.33) can be calculated as:

$$\lambda_j^l(\mathbf{n}^l, \Pi) = \frac{\mu_j \cdot A_{c_j}^l}{(1 - B_j^l)} \cdot P_{\mathbf{n},j}^l \quad (4.36)$$

By defining time intervals of length  $\Delta T$ , Equation (4.36) can be solved during network operation in two steps [RB16b, RB16d, RB17a]:

1. Within the interval, the network measures the link sojourn time in every state  $\mathbf{x}^l \in \widehat{\mathcal{X}}_{\mathbf{n}}^l$ . Let  $\Delta T_{\mathbf{x}}^l$  be the sojourn time measured in state  $\mathbf{x}^l$ .
2. At the end of the interval, the probability  $P_{\mathbf{n},j}^l$  is estimated as:

$$P_{\mathbf{n},j}^l = \frac{\sum_{\mathbf{x}^l \in \mathcal{Y}_{\mathbf{n},j}^l} \Delta T_{\mathbf{x}}^l}{\sum_{\mathbf{x}^l \in \widehat{\mathcal{X}}_{\mathbf{n}}^l} \Delta T_{\mathbf{x}}^l} \quad (4.37)$$

The sum in the denominator is the total link sojourn time in macrostate  $\mathbf{n}^l$ . In the numerator,  $\mathcal{Y}_{\mathbf{n},j}^l$  is the set of link states  $\mathbf{x}^l \in \widehat{\mathcal{X}}_{\mathbf{n}}^l$  that do not block class- $j$  arrivals (note that  $\mathcal{Y}_{\mathbf{n},j}^l \subseteq \widehat{\mathcal{X}}_{\mathbf{n}}^l$ .) The sum over  $\mathcal{Y}_{\mathbf{n},j}^l$  is then the total time that  $\mathbf{n}^l$  was configured by non-blocking states for class- $j$  traffic. Hence,  $P_{\mathbf{n},j}^l$  is the percentage of the sojourn time in macrostate  $\mathbf{n}^l$  that the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  was not forbidden. Similarly, in Equation (4.36), the probability  $(1 - B_j^l)$  is calculated as the fraction of time, w.r.t.  $\Delta T$ , the link was in non-blocking states for class- $j$  connections. Moreover, from the measurements, the traffic  $A_{c_j}^l$  is estimated as the mean number of class- $j$  connections simultaneously carried in  $\Delta T$ . The parameter  $\mu_j^{-1}$  can easily be determined as the average over all class- $j$  holding times over  $\Delta T$ . With this information, Equation (4.36) is then used to calculate  $\lambda_j^l(\mathbf{n}^l, \Pi)$ .

**Example 4.10** Consider the arrival rate  $\lambda_2^l(\mathbf{n}^l, \Pi_1) = \lambda_2^l(\Pi_1) \cdot P_{(1,0),2}^l$  studied in Example 4.9 for the macrostate  $\mathbf{n}^l = (1,0)$ . In this case,  $\mathcal{Y}_{(1,0),2}^l = \{\boldsymbol{\chi}_2, \boldsymbol{\chi}_6\}$  is the sub-set of link states in  $\widehat{\mathcal{X}}_{(1,0)}^l$  that do not

block class-2 arrivals (i.e. from the states in  $\widehat{X}_{(1,0)}^l$  only  $\mathbf{x}_2$  and  $\mathbf{x}_6$  have  $b_2 = 4$  free slots which are adjacent). To estimate the rate  $\lambda_2^l(\mathbf{n}^l, \Pi_1)$ , the sojourn times  $\Delta T_{\mathbf{x}_i}^l$  are measured for each  $\mathbf{x}_i \in \widehat{X}_{(1,0)}^l$ . Then from Equation (4.37),  $P_{(1,0),2}^l = (\Delta T_{\mathbf{x}_2}^l + \Delta T_{\mathbf{x}_6}^l) / (\Delta T_{\mathbf{x}_2}^l + \Delta T_{\mathbf{x}_3}^l + \Delta T_{\mathbf{x}_4}^l + \Delta T_{\mathbf{x}_5}^l + \Delta T_{\mathbf{x}_6}^l)$ .

### 4.3.3 Effect of the Contiguity Constraint on the Arrival Rates

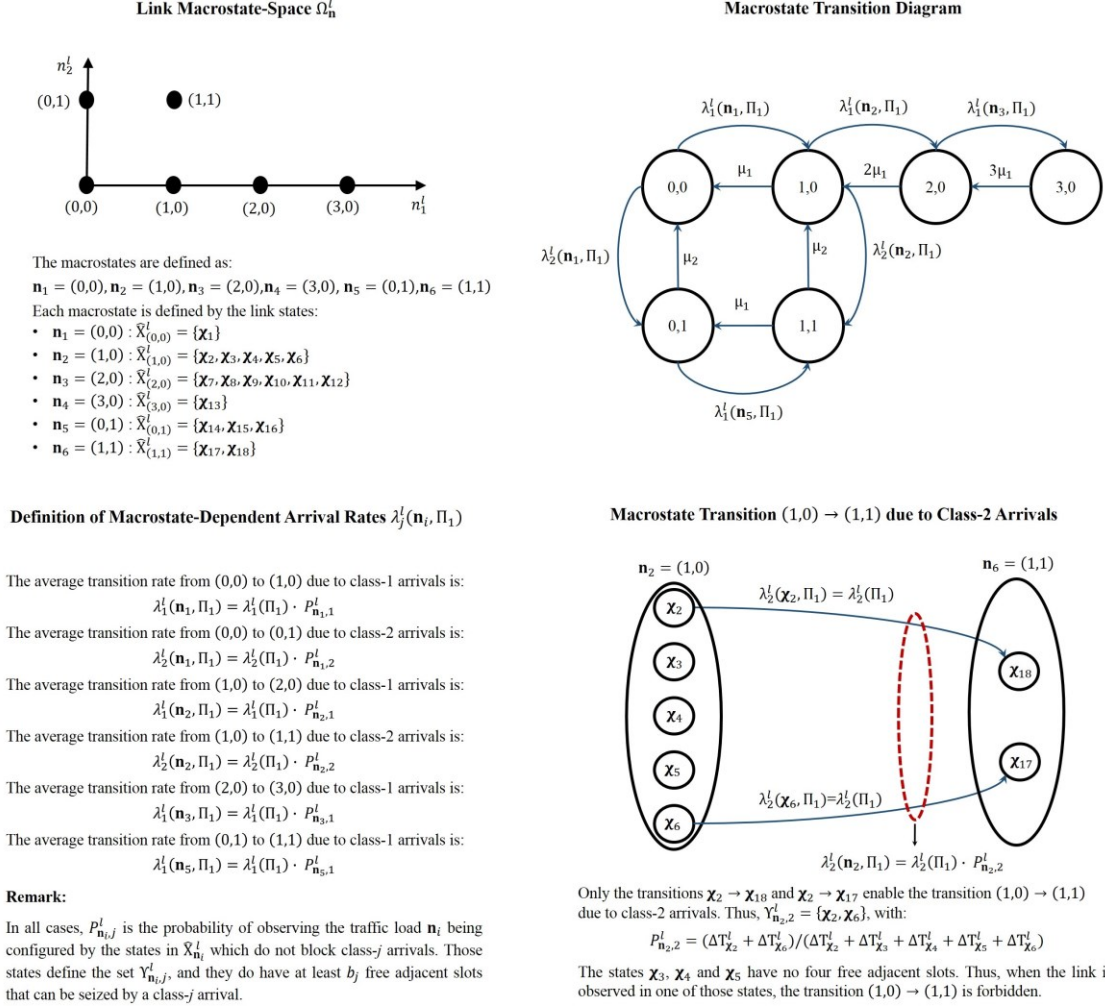
In the macrostate-dependent network reward model, the calculation of the state-spaces  $\Omega_{\mathbf{x}}^l$  is skipped. This does not imply that the effect that the contiguity constraint has on each link reward process is neglected. Instead, as it was pointed out in Section 4.2.1, for each link, every macrostate is defined by a sub-set of link states that fulfil the contiguity constraint. But, if the states in  $\Omega_{\mathbf{x}}^l$  are not calculated, how does the macrostate-dependent model account for the effect that the contiguity constraint has on the stochastic process  $\{\mathbf{N}_t^{l,\pi}, t \geq 0\}$ ? That effect is implicitly quantified by the arrival rates  $\lambda_j^l(\mathbf{n}^l, \Pi)$ . To understand this, note that the constraint has no effect in transitions  $\mathbf{n}^l \rightarrow \mathbf{n}^l - \delta_j^l$  originating from connection departures. It only restricts the possible link state moves  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  that may cause the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$  due to class- $j$  arrivals. For  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  to be feasible,  $\mathbf{x}^l$  must have  $b_j$  available adjacent slots. If that is the case,  $\mathbf{x}^l$  is a link state that does not block class- $j$  arrivals, i.e. it is an element in the set  $Y_{\mathbf{n},j}^l$ , and thus, the transition is allowed. Otherwise, it is denied. The effect of restricting the possible state moves is quantified by the rate  $\lambda_j^l(\mathbf{n}^l, \Pi)$  at which the link moves from  $\mathbf{n}^l$  to  $\mathbf{n}^l + \delta_j^l$ . This rate defines the probability  $P_{\mathbf{n}^l, \mathbf{n}^l + \delta_j^l}(\Pi)$  that the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$  occurs. This probability is less than that observed in a link that relaxes the contiguity constraint. By following an analysis similar to that presented in Chapter 3 - Section 3.3.2, we have that  $P_{\mathbf{n}^l, \mathbf{n}^l + \delta_j^l}(\Pi) = \lambda_j^l(\mathbf{n}^l, \Pi) \cdot \tau_{\mathbf{n}^l}$ , where  $\tau_{\mathbf{n}^l}$  is the mean sojourn time in  $\mathbf{n}^l$ . Thus, the estimation of the arrival rate  $\lambda_j^l(\mathbf{n}^l, \Pi)$  implies the estimation of the effect that the contiguity constraint has on the link performance, i.e. on the probability that a long-term reward  $g_j^l(\mathbf{n}^l, \Pi)$  (ru) is earned provided that a class- $j$  arrival occurs in macrostate  $\mathbf{n}^l$ . Observe that the contiguity constraint is enforced by every policy decision  $\Pi(\mathbf{x}, j)$ . The policy itself determines the state moves  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  which may cause  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$ . The estimation of  $\lambda_j^l(\mathbf{n}^l, \Pi)$  does not involve the calculation of  $\Omega_{\mathbf{x}}^l$  either. To solve Equation (4.36) it is only necessary to measure the sojourn times  $\Delta T_{\mathbf{x}}^l$  in the states which are observed to represent  $\mathbf{n}^l$ . By this strategy, the computational complexity of the state-dependent reward model is avoided and thus, the reward-based resource allocation problem becomes tractable.

**Example 4.11** In Fig. 4.6 we show the macrostate transition diagram defined by the link in Fig. 4.3. The diagram depicts the transitions between the macrostates and their corresponding transition rates. There are only six rates  $\lambda_j^l(\mathbf{n}^l, \Pi)$  that represent transitions of the form  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$ . These rates, as defined in Fig. 4.6, are given by the probabilities  $P_{\mathbf{n},j}^l$  of observing the link in states that do not block class- $j$  arrivals. A non-blocking state for class- $j$  traffic is one that has at least  $b_j$  free adjacent slots, thereby fulfilling the spectrum contiguity constraint. As an example, Fig. 4.6 illustrates the transition  $(1,0) \rightarrow (1,1)$  which was studied in Example 4.10. Note that if  $(1,0)$  is configured by  $\mathbf{x}_3, \mathbf{x}_4$  or  $\mathbf{x}_5$ , class- $j$  arrivals are rejected as the contiguity constraint cannot be satisfied.

## 4.4. Approximate Policy Iteration Algorithm

In this section we use the macrostate-dependent network reward model to formulate an approximate version of the PIA defined in Chapter 3. The approximate PIA also performs iteration cycles every  $\Delta T$  time units. However, it differs from the exact state-dependent algorithm in the following three relevant aspects:

1. At the end of every iteration cycle only the VDO step is executed. Upon completion of the  $i$ th cycle, the VDO calculates for every link  $l$ , the reward rate  $R^l(\Pi_i)$  and the values  $v(\mathbf{n}^l, \Pi_i)$  of the policy  $\Pi_i$  in use.
2. To make the policy calculation feasible, the PIR does not calculate the new policy  $\Pi_{i+1}$  at the end of the  $i$ th cycle. Instead, over the next cycle, the PIR uses a decision calculation algorithm to compute the decisions  $\Pi_{i+1}(\mathbf{x}, j)$  during network operation, one at a time, and only when necessary, i.e. upon arrival of a connection request. The PIR step is then executed every time that a connection request arrives at the network. This strategy not only reduces computational complexity, but it alleviates the memory requirements needed to store the policy decisions.



**Figure 4.6:** Definition of the policy and macrostate-dependent arrival rates for the link in Fig. 4.3.

3. Recall that in Chapter 3 the set  $\Lambda_x^{j+}$  was defined as the collection of all decisions available for a class- $j$  request arriving in network state  $\mathbf{x}$ . In addition to the decision “reject admission”, in the approximate PIA, the set  $\Lambda_x^{j+}$  contains at most one candidate lightpath for each path  $\rho$  in  $\Gamma_j$ .

In the following we describe the approximate PIA and present a comparison of this method with the exact algorithm outlined in Chapter 3.

#### 4.4.1 Definition of the Approximate Policy Iteration Algorithm

Consider a dynamic flex-grid optical network with  $N$  nodes and  $L$  links. Each link  $l$  has a capacity of  $C_l$  spectrum slots. The network serves  $J$  connection classes defined by the known parameters  $(o, d)_j, \lambda_j, \mu_j^{-1}, b_j, \Gamma_j$  and  $r_j$ . Resources are allocated to connections with the approximate PIA shown in Fig. 4.7, where we refer to an event as the arrival or departure of a connection.

During network operation the PIA performs iteration cycles every  $\Delta T$  time units. Within the  $i$ th cycle resources are allocated by a policy  $\Pi_i$  which is calculated in the VDO and the PIR steps of the PIA. This calculation procedure works as follows. Initially, the network enters a first iteration cycle within which resources are allocated by an arbitrarily chosen policy  $\Pi = \Pi_0$ . This policy is responsible for making all decisions on RSA and CAC for any connection arriving in the first interval  $\Delta T$ . Over the interval, each link measures its sojourn times in every observed link state by using the procedure outlined in Section 4.3.2. At the end of the interval, for each link  $l$ , the VDO step solves the linear system defined by Equation (4.16a) for  $R^l(\Pi_0)$  and the transient reward values  $v(\mathbf{n}^l, \Pi_0)$ . For this, the rates  $\lambda_j^l(\mathbf{n}^l, \Pi_0)$  are estimated from the measurements taken during the cycle period  $\Delta T$ . The network then enters the next cycle in which the PIR calculates a policy  $\Pi_1$  that outperforms  $\Pi_0$ . Unlike the exact PIA, which calculates the new policy before starting the next cycle, the approximate approach calculates every policy decision  $\Pi_1(\mathbf{x}, j)$  within

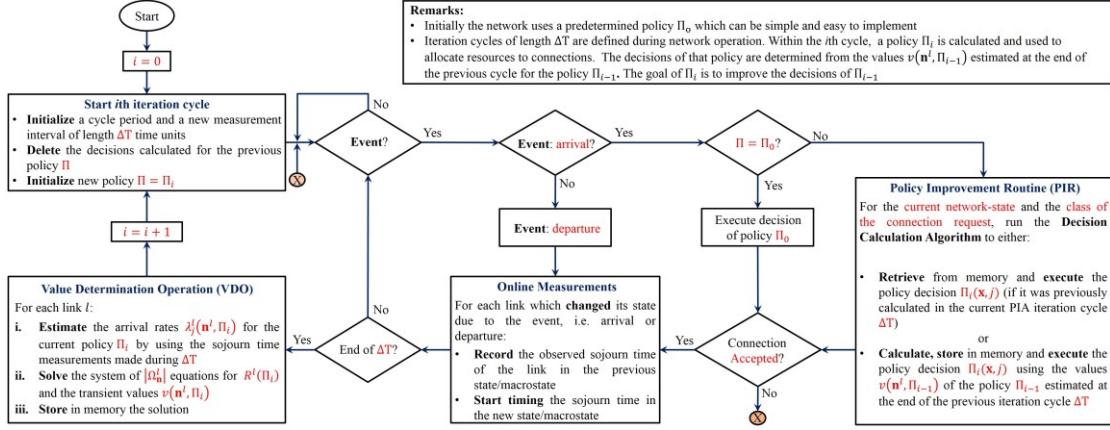


Figure 4.7: Approximate policy iteration algorithm (PIA).

the next cycle and only if a class- $j$  connection request arrives in state  $\mathbf{x}$ . If this event occurs, the PIR uses  $R^l(\Pi_0)$  and the transient values  $v(\mathbf{n}^l, \Pi_0)$  to calculate the decision  $\Pi_1(\mathbf{x}, j)$  that fulfils  $R(\Pi_1(\mathbf{x}, j)) \geq R(\Pi_0(\mathbf{x}, j))$ . The new decision is executed and stored in memory, so that it is used to allocate resources for future class- $j$  requests arriving within the current cycle in state  $\mathbf{x}$ . The impact that the new policy has on the network performance is captured by the link sojourn time measurements taken over the cycle. These measurements are used in the VDO (at the end of the cycle) to estimate  $\lambda_j^l(\mathbf{n}^l, \Pi_1)$  and to calculate  $R^l(\Pi_1)$  and  $v(\mathbf{n}^l, \Pi_1)$ . With this information, another iteration cycle is initiated to determine a policy  $\Pi_2$  that improves  $\Pi_1$ . This process is repeated so as to calculate a sequence of policies  $\Pi_1, \Pi_2, \dots, \Pi_i, \dots, \Pi^*$ , which result, respectively, in the network reward rates  $R(\Pi_1) < R(\Pi_2) < \dots < R(\Pi_i) < \dots \leq R(\Pi^*)$ . As with the exact PIA, the iterations are stopped when the policies obtained in two consecutive cycles attain the same network reward rate. However, this stopping criterion is not likely to be met in reality by the approximate PIA. The reason is that, as previously mentioned, the size of a policy is given by the cardinality of the network state-space and the number of connection classes. Since in a cycle decisions are calculated only when connections arrive, it is unlikely that arrivals occur in all network states for all connection classes within the cycle period  $\Delta T$ . Likewise, it is also unlikely that the decisions calculated for two consecutive cycles correspond exactly to the same states and connection classes. Hence, for real networks, the PIA in Fig. 4.7 is repeatedly executed so as to guarantee that in each cycle the performance is improved compared to the previous one. In the long-run, this process yields policies  $\Pi^*$  which are close (but not necessarily equal) to the optimum, and thus they are sub-optimal.

In summary, the  $i$ th iteration cycle ( $i > 0$ ) of the PIA aims at determining a policy  $\Pi_i$  whose decisions are calculated in the PIR step. The calculation involves the link rates  $R^l(\Pi_{i-1})$  and the values  $v(\mathbf{n}^l, \Pi_{i-1})$  estimated (for the policy  $\Pi_{i-1}$ ) at the end of the preceding iteration. During the  $i$ th cycle, measurements are taken to track the impact the policy  $\Pi_i$  has on the performance. That performance is quantified at the end of the cycle in the VDO step, which estimates  $\lambda_j^l(\mathbf{n}^l, \Pi_i)$ ,  $R^l(\Pi_i)$  and  $v(\mathbf{n}^l, \Pi_i)$  for all network links. This information is stored in memory and used in the next iteration cycle to determine the policy  $\Pi_{i+1}$ .

Any feasible policy  $\Pi_0$  might be used to initialize the PIA. In particular,  $\Pi_0$  can be determined by any of the online RSA algorithms currently proposed in the literature, see for example [CVR<sup>+</sup>12, TAK<sup>+</sup>14, WWH<sup>+</sup>11]. One of the most remarkable properties of the PIA is that, regardless of the chosen  $\Pi_0$ , better policies are always found throughout the iteration process. This justifies the adoption of policies  $\Pi_0$  which are simple to implement, so that their decisions can be calculated online without involving complex calculations in the first iteration cycle. In what follows, we explain in more detail how the VDO and the PIR perform their calculations in the  $i$ th iteration cycle.

#### 4.4.2 Estimation of the Policy Performance in the VDO Step

At the end of the  $i$ th iteration cycle, the PIA performs the VDO step in order to evaluate the performance of the policy  $\Pi_i$ . Such a performance is given by the reward rates  $R^l(\Pi_i)$  and the values  $v(\mathbf{n}^l, \Pi_i)$  of the network links. Based on the fact that the parameters  $(o, d)_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ ,  $r_j$  and  $C_l$  are known, the VDO in Fig. 4.7 implements the following procedure for each link  $l$ :

1. The link reward parameters  $r_j^l$  are calculated with Equation (4.11).

2. Since  $b_j$  and  $C_l$  are known, the capacity constraint  $\sum_{j=1}^J b_j \cdot n_j^l \leq C_l$  is used to calculate the link macrostate-space  $\Omega_{\mathbf{n}}^l$ . There is no need to involve the contiguity constraint in the calculation, as the network policy itself guarantees that any observable macrostate fulfils this restriction.
3. The rates  $\lambda_j^l(\mathbf{n}^l, \Pi_i)$  are estimated with Equation (4.36) from the sojourn time measurements taken over the  $i$ th iteration cycle.
4. The reward parameters  $r_j^l$ , the termination rates  $\mu_j$  and the arrival rates  $\lambda_j^l(\mathbf{n}^l, \Pi_i)$  are used to define the system of linear equations for the link. That system is given by Equation (4.16a).
5. The linear system is solved for  $R^l(\Pi_i)$  and the transient values  $v(\mathbf{n}^l, \Pi_i)$ . The VIA outlined in Section 4.2.2 can be applied as a solution method owing to its simplicity and proved outstanding performance (especially for large size macrostate-spaces). Any alternative fast linear algebra (or numeric) method suffices as well.
6. The solution to the linear system is stored in memory, so that the PIR uses it in the next iteration cycle to calculate the policy  $\Pi_{i+1}$ .

In case the macrostate-space  $\Omega_{\mathbf{n}}^l$  is too large, the simplified link model presented in Section 4.2.3 can be used to further reduce the size of the resulting linear system. If this course of action is taken, the VIA can be adopted as a solution method as well.

### 4.4.3 The Policy Calculation Procedure in the PIR Step

To illustrate the policy calculation procedure used by the PIR in Fig. 4.7, consider a network that enters the  $i$ th iteration cycle after having calculated - in the VDO step, and for all links - the rates  $R^l(\Pi_{i-1})$  and the values  $v(\mathbf{n}^l, \Pi_{i-1})$  of the policy used in the preceding cycle. The task of the PIR is to calculate a policy  $\Pi_i$  that allocates resources to incoming requests in the  $i$ th cycle period. This new policy must have a better performance than  $\Pi_{i-1}$ .

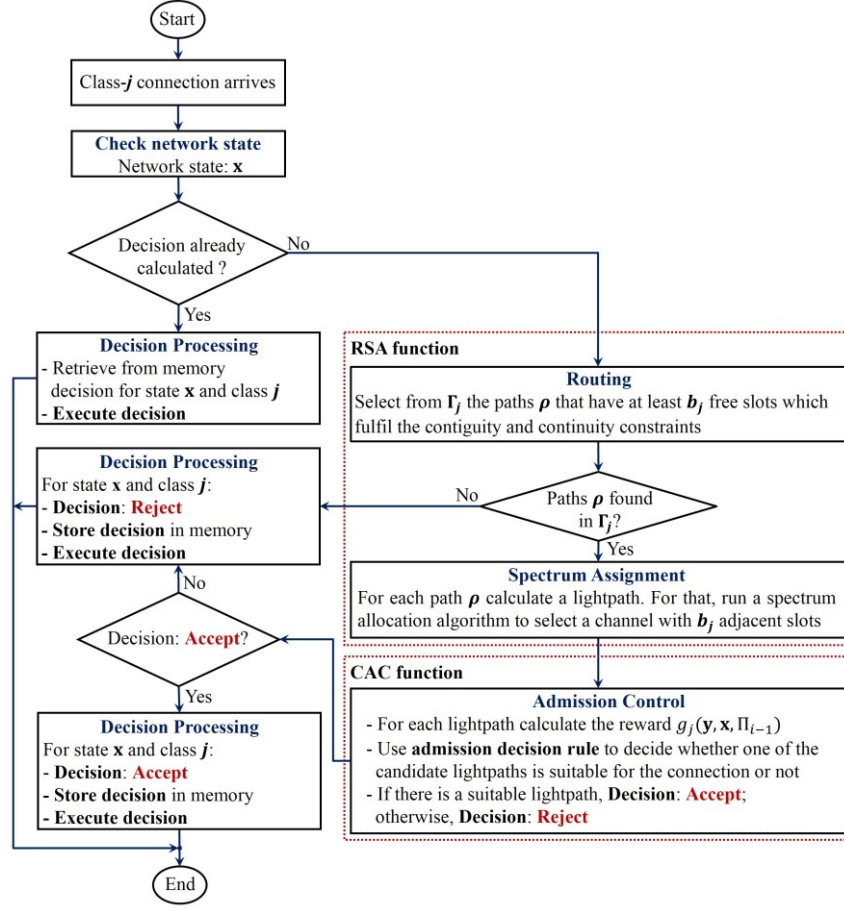
The decisions of  $\Pi_i$  are calculated over the  $i$ th cycle, one at a time, and only when necessary, i.e. upon arrival of a connection request. For that, the PIR applies the decision calculation algorithm shown in Fig. 4.8, which comprises network control functions for RSA and CAC. The algorithm works as follows: when a connection request of class- $j$  arrives in network state  $\mathbf{x}$ , it is checked whether the decision  $\Pi_i(\mathbf{x}, j)$  has already been calculated for that state and connection class. If that decision has not been calculated yet, the RSA function of the algorithm calculates the set of decisions  $\Lambda_{\mathbf{x}}^{j+}$  available for the connection request. This is accomplished in two steps. First, from the set of candidate routes  $\Gamma_j$ , the paths  $\rho$  with at least  $b_j$  free spectrum slots fulfilling the contiguity and continuity constraints are selected. Secondly, for each of these paths, an optical channel with bandwidth  $b_j$  slots is calculated by using a chosen spectrum allocation rule (e.g. first-fit, random-fit [TAK<sup>+</sup>14]). Thus, the RSA function calculates a candidate lightpath on every path  $\rho$  in  $\Gamma_j$  that satisfies the contiguity and continuity constraints. Each lightpath defines a decision in  $\Lambda_{\mathbf{x}}^{j+}$ . Recall that - as explained in Chapter 3 - in the set  $\Lambda_{\mathbf{x}}^{j+}$  a decision is represented by the state  $\mathbf{y}$  that a lightpath would configure in the network if it is allocated to the request that arrives in state  $\mathbf{x}$ . Besides those lightpaths,  $\Lambda_{\mathbf{x}}^{j+}$  contains the state  $\mathbf{x}$  as well, as the decision can be made that the class- $j$  request is rejected, and thus no state transition occurs in the network. The task of the CAC function of the algorithm is to calculate  $\Pi_i(\mathbf{x}, j)$  as the decision  $\mathbf{y}^*$  in  $\Lambda_{\mathbf{x}}^{j+}$  that fulfils  $R(\Pi_i(\mathbf{x}, j)) \geq R(\Pi_{i-1}(\mathbf{x}, j))$ . From Equation (3.30) in Section 3.4.2 - Chapter 3, we know that  $R(\Pi_i(\mathbf{x}, j))$  is obtained in the exact network reward model by solving the problem:

$$R(\Pi_i(\mathbf{x}, j)) = \max_{\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}} \{q(\mathbf{x}) + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j+}} \lambda_j(\mathbf{x}, \Pi_{i-1}) \cdot [v(\mathbf{y}, \Pi_{i-1}) - v(\mathbf{x}, \Pi_{i-1})] + \sum_{j=1}^J \sum_{\mathbf{y} \in \Gamma_{\mathbf{x}}^{j-}} \mu_j \cdot [v(\mathbf{y}, \Pi_{i-1}) - v(\mathbf{x}, \Pi_{i-1})]\} \quad (4.38)$$

where the decision  $\Pi_i(\mathbf{x}, j)$  is the decision  $\mathbf{y}^*$  in  $\Lambda_{\mathbf{x}}^{j+}$  that solves:

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}} \{g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})\} \quad (4.39)$$





**Figure 4.8:** Decision calculation algorithm executed by the PIA in the PIR step.

with  $g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1}) = v(\mathbf{y}, \Pi_{i-1}) - v(\mathbf{x}, \Pi_{i-1})$ . Observe that under the link independence assumption, the network reward rate  $R(\Pi_i(\mathbf{x}, j))$ , i.e. Equation (4.38), is expressed as the sum of the link reward rates:

$$R(\Pi_i(\mathbf{x}, j)) = \sum_l R^l(\Pi_i(\mathbf{x}, j)) \quad (4.40)$$

To analyse the implications of Equation (4.40), let us study the effect that a decision  $\mathbf{y} \in \Lambda_{\mathbf{x}}^{j+}$  has on the reward earned by the network. Let  $\rho$  be the path that corresponds to the lightpath implicitly defined by the decision (or equivalently the network state)  $\mathbf{y}$ . If a class- $j$  connection request arrives in the  $i$ th PIA iteration cycle, while the network is in the state  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^l, \dots, \mathbf{x}^L)$ , then the network carries a traffic given by the macrostate  $\mathbf{n} = \mathbf{n}(\mathbf{x}) = (\mathbf{n}^1, \dots, \mathbf{n}^l, \dots, \mathbf{n}^L)$ , recall that the concept of network macrostate was defined in Section 3.1.2 - Chapter 3. If the connection seizes the lightpath calculated in the path  $\rho$ , it would cause the transition  $\mathbf{x} \rightarrow \mathbf{y}$ . This implies that on each link  $l$  in  $\rho$ , the carried traffic changes as  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$ . From the values  $v(\mathbf{n}^l, \Pi_{i-1})$  calculated in the previous cycle, the network knows that the connection will bring to link  $l$  a long-term reward of  $g_j^l(\mathbf{n}^l, \Pi_{i-1}) = v(\mathbf{n}^l + \delta_j^l, \Pi_{i-1}) - v(\mathbf{n}^l, \Pi_{i-1})$  (ru). Therefore, if the connection is established on the path  $\rho$ , it would bring to the network a long-term reward of  $g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})$  (ru), which according to Equation (4.16e) is calculated as:

$$g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1}) \approx \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi_{i-1}) \quad (4.41)$$

This result is a consequence of the link independence assumption, which states that a connection affects the network reward rate by solely changing the reward rates of the links that it uses. Observe that the decision “reject admission” has gain  $g_j(\mathbf{x}, \mathbf{x}, \Pi_{i-1}) = 0$ , as  $\mathbf{y} = \mathbf{x}$  (i.e. there is no change in the network state). Based on this, the optimization problem in Equation (4.39) is solved by the CAC function in two steps. First, the long-term reward gains  $g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})$  are calculated for all possible decisions in  $\Lambda_{\mathbf{x}}^{j+}$  with Equation (4.41). Secondly, if all decisions  $\mathbf{y} \neq \mathbf{x}$  in  $\Lambda_{\mathbf{x}}^{j+}$  have negative gains, then the decision  $\Pi_i(\mathbf{x}, j)$  is to deny admission to the connection request. (Recall that a negative gain means that the connection yields

reward losses as it prevents admission of more valuable traffic.) Otherwise, since the decisions  $\mathbf{y} \neq \mathbf{x}$  in  $\Lambda_{\mathbf{x}}^{j+}$  define candidate lightpaths routed over different paths  $\rho$ , from Equation (4.41) we have that the optimum decision  $\mathbf{y}^*$  is that which solves:

$$\rho^* = \underset{\rho}{\operatorname{argmax}} \{ \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi_{i-1}) \} \quad (4.42)$$

which means that the decision  $\Pi_i(\mathbf{x}, j)$  is to accept the connection on the lightpath routed on the path  $\rho^*$  that yields that maximum positive long-term reward gain. Then  $\rho^*$  corresponds to the lightpath defined by the decision  $\mathbf{y}^*$  which has the maximum gain  $g_j(\mathbf{y}^*, \mathbf{x}, \Pi_{i-1}) > 0$ ,  $\mathbf{y}^* \neq \mathbf{x}$ . Therefore, under the link independence assumption, the admission decision rule defined by Equation (4.42) is phrased as: for all possible decisions in  $\Lambda_{\mathbf{x}}^{j+}$ , admit the connection on the lightpath with the highest positive reward gain. If all candidate lightpaths have negative gains, reject the connection. This rule was originally proposed in [RB16d], and we denote it as the Markov decision process (MDP) rule.

Once the decision  $\Pi_i(\mathbf{x}, j)$  is calculated by the CAC function, it is executed and stored in memory, thereby avoiding a recalculation of this decision for future class- $j$  connection requests arriving in the state  $\mathbf{x}$ . By this approach the policy is gradually defined while memory requirements and computational complexity are alleviated. Another advantage is that the decisions adapt to variations in the parameters  $\lambda_j$ ,  $\mu_j^{-1}$ ,  $b_j$ ,  $\Gamma_j$ , and  $r_j$ . (Note that the adaptability follows from the fact that, at end of every iteration cycle, the VDO step uses the current values of these parameters to calculate the policy performance.)

The link independence assumption reduces mathematical complexity, but it might be oversimplifying. In reality, any carried connection induces state correlations among links which are not in the path set  $\Gamma_j$  (due to the continuity and contiguity constraints). The contributions from those links to the long-term rewards  $g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})$  are not accounted for by Equation (4.41). However, as studied in [DPKW88, DM89, DM92, DM94, Kri91, HKT00, Hwa93, Dzi97, Nor02] the assumption is still justified by the fact that an exact calculation of  $g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})$  is not feasible even for small networks (because of the huge cardinality of the exact network model). The extent to which the assumption is valid is strongly dependent on the network topology, as networks that route connections on multi-link paths (e.g. ring, partial-mesh networks) are prone to induce more correlations than those which use many direct link paths (e.g. highly meshed networks). To counteract the lack of accuracy of the assumption, Equation (4.41) can still be used to define rules that intend to reduce the correlations among links, thereby improving the accuracy of the decisions calculated by the PIA [RB17a]. Besides the MDP rule, Table 4.3 presents three alternative admission decision rules which can be used by the CAC function shown in Fig. 4.8. For a connection of class- $j$  that requests admission in network state  $\mathbf{x}$ , the rules need as input the candidate lightpaths calculated by the RSA function. The rule denoted as MDP-SP (Shortest-Path) checks if the RSA function has calculated a lightpath on the shortest path (w.r.t. the number of links) in  $\Gamma_j$ . If so, admission is granted on that path; otherwise, the MDP rule is used to place the connection on the path with the maximum positive reward gain. (Remark: thus, regardless of its gain, priority is given to the shortest path as long as it has  $b_j$  free slots fulfilling the constraints.) The MDP-PG (Positive-Gain) rule selects from the set of candidate paths, the route with positive gain which has the shortest length (w.r.t. the number of links). This rule tries to avoid large paths with positive gains. If no path has  $g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1}) \geq 0$ , admission is denied. The MDP-PGMC (Positive-Gain-Maximum-Capacity) rule selects the route with the maximum available capacity (in slots) which has a positive gain. Therefore, this rule discards paths with higher gains and less available capacity.

Ring and partial-mesh are the most common topologies used to deploy optical networks. For these topologies, the sets  $\Gamma_j$  mainly contain multi-link paths which lead to correlations in the resource occupation of the network links. This effect (which due to the link independence assumption is not quantified by MDP rule) can be harmful when connections use routes with large number of links. To counteract this, the MDP-SP and the MDP-PG rules try to minimize the length of the selected path. The former rule follows a less conservative approach (w.r.t. the reward), as it prioritizes the shortest path in  $\Gamma_j$  regardless of its gain. The latter rule selects the shortest path (from the set of candidate routes) that avoids a decrement in the long-term network reward. An alternative to these two strategies is the MDP-PGMC rule. Instead of minimizing the route length, it mitigates the problem by prioritizing the path with positive gain which has more available capacity. Thus, it prevents congestion on highly loaded paths.

In Table 4.4 we present a comparison between the exact state-dependent PIA and the macrostate-dependent approximate approach. Most of the differences therein summarized have already been argued. Nevertheless it is worth discussing a remarkable difference regarding the set of decisions  $\Lambda_{\mathbf{x}}^{j+}$ . In the exact PIA, that set contains all the lightpaths available in state  $\mathbf{x}$  for a class- $j$  request. Thus, there can be more than one candidate lightpath that uses the same path  $\rho$  in  $\Gamma_j$ . In the approximate PIA, at most one candidate lightpath is calculated on each path  $\rho$ . The reason for this stems from the fourth property of the macrostate-

Rule	Description
MDP	<p>i) From the set of candidate lightpaths calculated by the RSA function, select the lightpath that uses the path <math>\rho^*</math> that yields the maximum positive long-term reward gain <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})</math></p> <p>ii) If <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1}) &lt; 0</math> for all lightpaths, reject the connection; otherwise, admit it on the lightpath in the path <math>\rho^*</math></p>
MDP-SP	<p>i) Check if the set of candidate lightpaths calculated by the RSA function contains a lightpath that uses the shortest path (w.r.t. the number of links) in <math>\Gamma_j</math></p> <p>ii) If so, regardless of its gain, admit the connection on that path; otherwise, select the lightpath that yields the maximum positive gain <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})</math>. If for all lightpaths <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1}) &lt; 0</math>, reject the connection</p>
MDP-PG	<p>i) For each route <math>\rho</math> in the set of candidate lightpaths calculated by the RSA function, determine the corresponding long-term reward <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})</math></p> <p>ii) Admit the connection on the path <math>\rho</math> with positive gain which has the shortest length (w.r.t. the number of links). If <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1}) &lt; 0</math>, for all <math>\rho</math>, reject the connection</p>
MDP-PGMC	<p>i) For each route <math>\rho</math> in the set of candidate lightpaths calculated by the RSA function, calculate the corresponding long-term gains <math>g_j(\mathbf{y}, \mathbf{x}, \Pi_{i-1})</math></p> <p>ii) Admit the connection on the path <math>\rho</math> with the maximum available capacity (in slots) that has a positive path-reward gain. If no path fulfils this criterion, reject the connection</p>

**Table 4.3:** Admission decision rules used by the PIR within the  $i$ th iteration cycle [RB17a].

Step	Exact PIA	Approximate PIA
VDO	Executed at the end of the $i$ th iteration cycle to determine the network reward rate $R(\Pi_i)$ and the values $v(\mathbf{x}, \Pi_i)$ . This is achieved by solving a linear system of $ \Omega_{\mathbf{x}} $ equations. The system is defined in the network state-space $\Omega_{\mathbf{x}}$	Executed at the end of the $i$ th iteration cycle. The rate $R(\Pi_i)$ is estimated as the sum of the link reward rates $R^l(\Pi_i)$ . For this, $L$ linear systems are defined in the macrostate-spaces $\Omega_{\mathbf{n}}^l$ , one system for each link $l$ . Every system is solved for $R^l(\Pi_i)$ and the values $v(\mathbf{n}^l, \Pi_i)$
PIR	<p>Executed at the end of the <math>i</math>th cycle, where every decision <math>\Pi_{i+1}(\mathbf{x}, j)</math> is the decision <math>\mathbf{y}^*</math> in <math>\Lambda_{\mathbf{x}}^{j+}</math> that brings the maximum positive long-term reward:</p> $g_j(\mathbf{y}, \mathbf{x}, \Pi_i) = v(\mathbf{y}, \Pi_i) - v(\mathbf{x}, \Pi_i)$ <p>The set <math>\Lambda_{\mathbf{x}}^{j+}</math> contains all the possible lightpaths available for the request in state <math>\mathbf{x}</math>, i.e. more than one lightpath can be recommended on each path <math>\rho</math> in <math>\Gamma_j</math>. The PIA stops when the policies calculated in two consecutive iterations have the same performance</p>	<p>Executed over the next cycle, where <math>\Pi_{i+1}(\mathbf{x}, j)</math> is only calculated if a class-<math>j</math> arrival occurs in state <math>\mathbf{x}</math>. Different admission decision rules can be used to calculate <math>\Pi_{i+1}(\mathbf{x}, j)</math> from the set <math>\Lambda_{\mathbf{x}}^{j+}</math>. By using the link independence assumption the rules estimate the long-term rewards as:</p> $g_j(\mathbf{y}, \mathbf{x}, \Pi_i) \approx \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi_i)$ <p>The set <math>\Lambda_{\mathbf{x}}^{j+}</math> has at most one lightpath for each path <math>\rho</math> in <math>\Gamma_j</math>. The PIA is not likely to satisfy the stopping criterion</p>

**Table 4.4:** Comparison between the exact and the approximate PIA algorithms.

dependent model discussed in Section 4.2.1. This property states that the long-term rewards  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$  of all the possible link state moves  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  that cause the transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  are approximated as  $g_j^l(\mathbf{n}^l, \Pi)$ . (Recall that the transient values  $v(\mathbf{n}^l, \Pi)$  are used as an approximation to the values  $v(\mathbf{x}^l, \Pi)$  of the states in  $\bar{\mathcal{X}}_{\mathbf{n}}^l$ .) Owing to this approximation, all the candidate lightpaths that use the same path  $\rho$ , cause transitions  $\mathbf{x} \rightarrow \mathbf{y}$  which are assumed to yield the same reward gain  $g_j(\mathbf{y}, \mathbf{x}, \Pi) \approx \sum_{l \in \rho} g_j^l(\mathbf{n}^l, \Pi)$ . As a result, under the macrostate-dependent formulation, it suffices that the RSA function selects only one lightpath on each path  $\rho$  that fulfils the contiguity and continuity constraints. This is accomplished by using a spectrum allocation algorithm that calculates an optical channel for every route  $\rho$ . However, since in reality the reward gains  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$  that cause  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \boldsymbol{\delta}_j^l$  need not be alike, the spectrum allocation algorithm used by the RSA function is expected to have an influence on the network performance. This issue will be investigated in more detail in Chapter 5. Regardless of the PIA variant, it is worth pointing out that both the exact and the approximate versions rely on the estimation of transient reward values. Their importance arises from the fact that they define the long-term rewards  $g_j(\mathbf{y}, \mathbf{x}, \Pi)$  that a connection brings to the network. These values inform the PIA (in the PIR) whether in state  $\mathbf{x}$  it is pertinent to admit a class- $j$  connection that would cause the state transition  $\mathbf{x} \rightarrow \mathbf{y}$ .

**Example 4.12** Let us study the calculation of the decision  $\Pi_{i+1}(\boldsymbol{\chi}_1, 2)$  for the single-link network in Fig. 4.3. The decision corresponds to a class-2 arrival in state  $\boldsymbol{\chi}_1 = (0, 0, 0, 0, 0, 0)$ , i.e. when no traffic is being carried. For class-2, the set  $\Gamma_2$  has a single-link path defined as  $\rho = (A, B)$ . From Fig. 4.3, we have that three lightpaths are available for the request in the path  $\rho$ . They are defined by the states  $\boldsymbol{\chi}_{14} = (2, \infty, \infty, \infty, 0, 0)$ ,  $\boldsymbol{\chi}_{15} = (0, 2, \infty, \infty, \infty, 0)$  and  $\boldsymbol{\chi}_{16} = (0, 0, 2, \infty, \infty, \infty)$ . If the exact PIA is used, the set of decisions defined in the PIR step is  $\Lambda_{\mathbf{x}}^{j+} = \{\boldsymbol{\chi}_1, \boldsymbol{\chi}_{14}, \boldsymbol{\chi}_{15}, \boldsymbol{\chi}_{16}\}$ . Then the policy decision  $\Pi_{i+1}(\boldsymbol{\chi}_1, 2)$  is determined by the state  $\boldsymbol{\chi}_i$  in  $\Lambda_{\mathbf{x}}^{j+}$  that yields the maximum reward gain  $g_2(\boldsymbol{\chi}_i, \boldsymbol{\chi}_1, \Pi_i) = v(\boldsymbol{\chi}_i, \Pi_i) - v(\boldsymbol{\chi}_1, \Pi_i)$ . For example, if the maximum reward is provided by  $\boldsymbol{\chi}_{16}$ , the decision  $\Pi_{i+1}(\boldsymbol{\chi}_1, 2)$  is: admit the connection on the path  $\rho = (A, B)$  by allocating the slots 3-6. On the other hand, in the approximate PIA, since the three possible lightpaths use the same path, their long-term rewards are assumed to be alike and are approximated as  $g_j(\boldsymbol{\chi}_i, \boldsymbol{\chi}_1, \Pi_i) \approx g_2^l(\mathbf{n}^l, \Pi_i)$ , with  $\mathbf{n}^l = (0, 0)$ . (Note that these lightpaths define the state transitions  $\boldsymbol{\chi}_1 \rightarrow \boldsymbol{\chi}_{14}$ ,  $\boldsymbol{\chi}_1 \rightarrow \boldsymbol{\chi}_{15}$  and  $\boldsymbol{\chi}_1 \rightarrow \boldsymbol{\chi}_{16}$  that cause the macrostate transition  $(0, 0) \rightarrow (0, 1)$ .) Therefore, in the approximate approach only one of the three options is selected to calculate  $\Pi_{i+1}(\boldsymbol{\chi}_1, 2)$ . For that, the PIR uses a spectrum allocation algorithm. If first-fit is used, then  $\Lambda_{\mathbf{x}}^{j+} = \{\boldsymbol{\chi}_1, \boldsymbol{\chi}_{14}\}$ , as  $\boldsymbol{\chi}_{14}$  is the state that contains the first four adjacent available slots. If a random-fit algorithm is considered, then  $\Lambda_{\mathbf{x}}^{j+} = \{\boldsymbol{\chi}_1, \boldsymbol{\chi}_i\}$ , where  $\boldsymbol{\chi}_i$  is randomly selected from  $\{\boldsymbol{\chi}_{14}, \boldsymbol{\chi}_{15}, \boldsymbol{\chi}_{16}\}$ . Independently of the spectrum allocation algorithm, the approximate PIA defines sets  $\Lambda_{\mathbf{x}}^{j+}$  with only two elements, namely,  $\boldsymbol{\chi}_1$  which represents the decision ‘‘connection rejection’’, and  $\boldsymbol{\chi}_i$ ,  $i = 14, 15, 16$ , which represents a unique candidate lightpath calculated in the path  $\rho$ . The policy decision  $\Pi_{i+1}(\boldsymbol{\chi}_1, 2)$  is obtained by calculating  $g_2^l(\mathbf{n}^l, \Pi_i)$ . Assuming the MDP rule, if  $g_2^l(\mathbf{n}^l, \Pi_i) > 0$ , the connection is accepted on the selected lightpath; otherwise, it is rejected. Observe that in reality, both the exact and the approximate PIA attain the same performance if all three lightpaths have the same gains  $g_j(\boldsymbol{\chi}_i, \boldsymbol{\chi}_1, \Pi_i) = g_2^l(\mathbf{n}^l, \Pi_i)$ . However, since those gains may differ, the approximate PIA is expected to have a performance less than that of the exact model. Such a performance depends on the spectrum allocation algorithm used by the RSA function.

## 4.5. Chapter Summary

In this chapter an approximate approach to online resource allocation was proposed for dynamic optical networks. The approach stems from a simplification made on the exact state-dependent PIA that assumes statistical independence of the link state distributions. As a result, a network with  $L$  links is treated as a compound of  $L$  independent single-link networks, thereby approximating the network reward rate by the sum of the link reward rates. By this assumption, upon arrival of a connection request, the approximate PIA first calculates (based on the network state) candidate lightpaths routed over different paths. Then the long-term reward that each lightpath yields is determined, and admission is granted on the lightpath with the highest positive reward. If no lightpath has positive gain, the connection request is rejected. To counteract the lack of accuracy of the approach (due to the actual presence of correlations among network links), different admission decision rules were defined for the PIA. They intend to minimize the harmful effect that connections routed over large multi-link paths may have in the network performance.



# Chapter 5

## Performance Evaluation Results

In this chapter we present performance evaluation results for selected network scenarios. For that purpose, simulations were set up to assess the performance of the PIA under different traffic load conditions and for different network topologies. Section 5.1 outlines the definitions needed for the interpretation of the results presented in the subsequent sections. In Section 5.2 we present a performance comparison of the exact state-dependent reward model with the approximate macrostate-dependent approach. By considering two single-link networks, special attention is given to two aspects, namely, the accuracy of the macrostate-dependent long-term reward gains and the performance of adaptive and state-dependent admission control. In Section 5.3 the performance of the approximate PIA is evaluated in multi-link networks. For that, full-mesh, partial-mesh and ring topologies are considered to investigate three aspects: first, the reward losses attainable under different traffic loads by the MDP-based admission decision rules; secondly, the influence of the spectrum allocation algorithm on the PIA performance; and last, the problem of GoS control. Most of the results presented in this chapter have been published by the author in [RB16a, RB16b, RB16c, RB16d, RB17a].

### 5.1. General Remarks

In this chapter we present results that are derived from analytical evaluations and simulations. In order to facilitate the discussion of these results, and to avoid unnecessary repetitions, in this section we outline the definitions and assumptions common to all considered performance evaluation scenarios.

#### 5.1.1 Solution Method for the Systems of Linear Equations

All analytical evaluations and simulations that involve the solution to the linear systems of the exact or the approximate models use the VIA as solution method (see Section 4.2.2 in Chapter 4). The iteration procedure of the VIA is stopped by defining a relative accuracy  $\epsilon = 0.1$ . Recall that the VIA circumvents a direct solution of the linear systems, and instead, it uses an iteration procedure which is computationally advantageous for large capacity networks.

#### 5.1.2 Simulation of the Exact and the Approximate PIA Algorithms

All simulations run to evaluate the reward-based algorithms follow the guidelines for connection-level performance simulation discussed in [SDM10]. Therein, it is stated that “*a (performance evaluation) scenario consists of the network model, the traffic model, the simulator and the investigated method*”. In our case, the investigated methods are the exact and the approximate PIA schemes (as defined in Chapters 3 and 4). The network and traffic models are defined in detail in Sections 5.2 and 5.3 for each considered scenario. Suffice it to clarify that, in all cases, class- $j$  connections arrive according to a Poisson process with rate  $\lambda_j$ , and have exponentially distributed holding times with mean  $\mu_j^{-1}$ . On the other hand, to evaluate the performance of both PIA variants, a discrete event simulation framework was implemented in R [Rpr17, LDL13]. All simulations were performed on a 6-core machine with 64 GB RAM running Linux on an Intel X86 server. Furthermore, for all considered performance evaluation scenarios, the simulations were set up with the following assumptions:

- The size of the iteration cycles  $\Delta T$  (for the exact and the approximate PIA algorithms) is defined as  $\Delta T = \alpha \cdot \max_j \mu_j^{-1}$ , where  $\alpha \gg 0$  is a positive constant. Therefore,  $\Delta T$  is proportional to the

maximum holding time  $\mu_j^{-1}$ . In all considered scenarios, we set  $\alpha = 100000$  so as to guarantee that  $\Delta T$  is large enough to provide reliable estimates of the effect that the policy decisions have on the network performance.

- As argued in Chapter 4, the approximate PIA is unlikely to meet the stopping criterion as it is improbable that arrivals occur in all states for all connection classes within a cycle  $\Delta T$ . Moreover, it is also unlikely that the decisions calculated for two consecutive cycles correspond exactly to the same states and connection classes. (Actually, in all scenarios studied in this chapter, the stopping criterion was never met.) Therefore, in order to provide simulation results, we define the following stopping criterion for the approximate PIA. Let  $\delta \geq 0$  be the desired relative accuracy with which  $R(\Pi)$  needs to be estimated. Thus, for a given scenario, let  $R(\Pi_{i-1})$  and  $R(\Pi_i)$  be the simulated network reward rates attained in two consecutive cycles. At the end the  $i$ th cycle, and after executing the VDO step, the following inequality is calculated:

$$|R(\Pi_i) - R(\Pi_{i-1})|/R(\Pi_i) \leq \delta \quad (5.1)$$

If it is fulfilled, then the simulation is stopped and  $R(\Pi) = R(\Pi_i)$  is regarded as the performance achieved by the approximate PIA. In all results reported in this chapter, it is further specified the  $i$ th iteration cycle that meets the stopping criterion. Moreover, for all considered performance evaluation scenarios we set  $\delta = 0.1$ .

- It is worth emphasizing that the stopping criterion defined in Equation (5.1) applies only to the approximate (i.e. the macrostate-dependent) PIA. The exact state-dependent approach follows the stopping rule defined in Chapter 3.

### 5.1.3 Presentation of Results

In most performance evaluation scenarios studied in this chapter the results from analytical evaluations and simulations are presented based on the following definitions:

- Two metrics are defined to assess the performance of the exact and the approximate PIA. One metric is the network reward rate  $R(\Pi) = R(\Pi_i)$ . The second metric is the network reward loss  $R_L(\Pi) = R_L(\Pi_i)$ , which is calculated as:

$$R_L(\Pi) = \frac{V_o - V_r}{V_o} \quad (5.2)$$

with  $V_o$  and  $V_r$  being the rewards offered to and received (or carried) by the network, respectively. Thus,  $R(\Pi)$  and  $R_L(\Pi)$  are to be interpreted, respectively, as the reward rate and the reward loss caused by the resource allocation policy  $\Pi = \Pi_i$ .

- For all considered performance evaluation scenarios, the simulated  $R(\Pi)$  and  $R_L(\Pi)$  are reported with 95% confidence intervals [Bou11, DBÇ15]. These intervals are either plotted in figures (together with the mean estimates) or presented in tables depending on the mechanism chosen to report the simulation results for each scenario.
- In some evaluation scenarios, where performance comparisons are made, the results are compared by calculating the relative approximate error:

$$\text{Relative approximate error (\%)} = 100 \cdot |X_A - X_B|/X_A \quad (5.3)$$

where  $X_A$  and  $X_B$  are the performance metrics to be compared. The error is calculated relative (or with respect) to  $X_A$ . In every evaluation scenario it is explicitly defined the metrics that  $X_A$  and  $X_B$  represent.

It would be desirable to present results that compare the performance of the exact and the approximate PIA algorithms for different network types, however, this is only possible for very small networks (given that the exact model is intractable for large-networks). For this reason, in Section 5.2 we resort to the small single-link network studied in Chapter 4 to compare the exact and approximate schemes. Special attention is given to the approximate long-term reward gains and their variations with the traffic load. Then in Section 5.3 we solely investigate the approximate PIA for the case of multi-link optical networks, where the performance is evaluated for the MDP-based rules defined in Chapter 4.

State and Macrostate Spaces							Resource Allocation Policy								
Nr	Link State-Space $\Omega_x^l$						Macrostate $\mathbf{n}^l = (n_1^l, n_2^l)$	$\bar{X}_{\mathbf{n}}^l$	Policy $\Pi = \Pi_1$						
	State $\mathbf{x}^l$								Class- $j$						
	$x_1^l$	$x_2^l$	$x_3^l$	$x_4^l$	$x_5^l$	$x_6^l$		State $\mathbf{x}^l$	1			2			
									Admit	Path	Slots	Admit	Path	Slots	
$\chi_1$	0	0	0	0	0	0	(0,0)	$\bar{X}_{(0,0)}^l$							
$\chi_2$	1	$\infty$	0	0	0	0	(1,0)	$\bar{X}_{(1,0)}^l$	$\chi_1$	Yes	(A,B)	1-2	Yes	(A,B)	1-4
$\chi_3$	0	1	$\infty$	0	0	0			$\chi_2$	Yes	(A,B)	3-4	Yes	(A,B)	3-6
$\chi_4$	0	0	1	$\infty$	0	0			$\chi_3$	Yes	(A,B)	4-5	No	-----	-----
$\chi_5$	0	0	0	1	$\infty$	0			$\chi_4$	Yes	(A,B)	1-2	No	-----	-----
$\chi_6$	0	0	0	0	1	$\infty$			$\chi_5$	Yes	(A,B)	1-2	No	-----	-----
$\chi_7$	1	$\infty$	1	$\infty$	0	0			$\chi_6$	Yes	(A,B)	1-2	Yes	(A,B)	1-4
$\chi_8$	1	$\infty$	0	1	$\infty$	0	(2,0)	$\bar{X}_{(2,0)}^l$	$\chi_7$	Yes	(A,B)	5-6	No	-----	-----
$\chi_9$	1	$\infty$	0	0	1	$\infty$			$\chi_8$	No	-----	-----	No	-----	-----
$\chi_{10}$	0	1	$\infty$	1	$\infty$	0			$\chi_9$	Yes	(A,B)	3-4	No	-----	-----
$\chi_{11}$	0	1	$\infty$	0	1	$\infty$			$\chi_{10}$	No	-----	-----	No	-----	-----
$\chi_{12}$	0	0	1	$\infty$	1	$\infty$			$\chi_{11}$	No	-----	-----	No	-----	-----
$\chi_{13}$	1	$\infty$	1	$\infty$	1	$\infty$			$\chi_{12}$	Yes	(A,B)	1-2	No	-----	-----
$\chi_{14}$	2	$\infty$	$\infty$	$\infty$	0	0	(3,0)	$\bar{X}_{(3,0)}^l$	$\chi_{13}$	No	-----	-----	No	-----	-----
$\chi_{15}$	0	2	$\infty$	$\infty$	$\infty$	0	(0,1)	$\bar{X}_{(0,1)}^l$	$\chi_{14}$	Yes	(A,B)	5-6	No	-----	-----
$\chi_{16}$	0	0	2	$\infty$	$\infty$	$\infty$			$\chi_{15}$	No	-----	-----	No	-----	-----
$\chi_{17}$	2	$\infty$	$\infty$	$\infty$	1	$\infty$			$\chi_{16}$	Yes	(A,B)	1-2	No	-----	-----
$\chi_{18}$	1	$\infty$	2	$\infty$	$\infty$	$\infty$	(1,1)	$\bar{X}_{(1,1)}^l$	$\chi_{17}$	No	-----	-----	No	-----	-----
									$\chi_{18}$	No	-----	-----	No	-----	-----

### Link Macrostate-Space $\Omega_n^l$

The six macrostates are denoted as:

$$\mathbf{n}_1 = (0,0), \mathbf{n}_2 = (1,0), \mathbf{n}_3 = (2,0), \mathbf{n}_4 = (3,0), \mathbf{n}_5 = (0,1), \mathbf{n}_6 = (1,1)$$

Each macrostate is defined by the link states:

- $\mathbf{n}_1 = (0,0) : \bar{X}_{(0,0)}^l = \{\chi_1\}$
- $\mathbf{n}_2 = (1,0) : \bar{X}_{(1,0)}^l = \{\chi_2, \chi_3, \chi_4, \chi_5, \chi_6\}$
- $\mathbf{n}_3 = (2,0) : \bar{X}_{(2,0)}^l = \{\chi_7, \chi_8, \chi_9, \chi_{10}, \chi_{11}, \chi_{12}\}$
- $\mathbf{n}_4 = (3,0) : \bar{X}_{(3,0)}^l = \{\chi_{13}\}$
- $\mathbf{n}_5 = (0,1) : \bar{X}_{(0,1)}^l = \{\chi_{14}, \chi_{15}, \chi_{16}\}$
- $\mathbf{n}_6 = (1,1) : \bar{X}_{(1,1)}^l = \{\chi_{17}, \chi_{18}\}$

#### Remarks:

- Optical link with a capacity of  $C_l = 6$  spectrum slots connecting two nodes (A,B)
- Two connection classes are served by the link. They have bandwidths of  $b_1 = 2$  and  $b_2 = 4$  spectrum slots
- The link state-space  $\Omega_x^l$  defines 18 states that fulfil the spectrum contiguity constraint
- The 18 states  $\mathbf{x}^l$  in the link state-space  $\Omega_x^l$  are denoted as  $\chi_i, i = 1, \dots, 18$
- The link macrostate-space is  $\Omega_n^l = \{(0,0), (1,0), (2,0), (3,0), (0,1), (1,1)\}$ , i.e. there are six macrostates
- The six macrostates  $\mathbf{n}^l$  in the link macrostate-space  $\Omega_n^l$  are denoted as  $\mathbf{n}_i, i = 1, \dots, 6$
- $\bar{X}_{\mathbf{n}}^l$  is the set of link states  $\mathbf{x}^l$  that represent the same macrostate  $\mathbf{n}^l = (n_1^l, n_2^l)$
- The policy  $\Pi_1$  is an  $18 \times 2$  matrix, i.e. it defines 36 decisions, 18 for each connection class

**Figure 5.1:** Definition of the policy, the state and the macrostate spaces for the single-link network.

## 5.2. Performance Comparison of the Exact and the Approximate Reward Models

In this section we present a performance comparison of the exact state-dependent reward model with the approximate macrostate-dependent approach. Since the exact model is unsolvable for large networks, we consider a small single-link network with capacity  $C_l = 6$  slots that serves two connection classes with bandwidths  $b_1 = 2$  and  $b_2 = 4$ . Besides the resource allocation policy, Fig 5.1 defines the state and the macrostate-spaces for this link. The link state-space  $\Omega_x^l$  consists of 18 states  $\mathbf{x}^l$  denoted as  $\chi_i, i = 1, \dots, 18$ . These states define a macrostate-space  $\Omega_n^l$  with six macrostates denoted as  $\mathbf{n}_i, i = 1, \dots, 6$ . The



$A$ (Erl)	$A_1$ (Erl)	$A_2$ (Erl)	$\lambda_1$ (con/uot)	$\lambda_2$ (con/uot)	$\mu_1$ (con/uot)	$\mu_2$ (con/uot)	$r_1$ (ru)	$r_2$ (ru)
0.001	0.0001	0.0009	0.0001	0.0001	1.0	0.1	2.5	4.0
0.010	0.0009	0.0091	0.0009	0.0009	1.0	0.1	2.5	4.0
0.100	0.0091	0.0909	0.0091	0.0091	1.0	0.1	2.5	4.0
1.000	0.0909	0.9091	0.0909	0.0909	1.0	0.1	2.5	4.0
10.00	0.9091	9.0909	0.9091	0.9091	1.0	0.1	2.5	4.0

**Table 5.1:** Parameters for the comparison of the exact and the approximate reward models.

	Link Reward Rate $R(\Pi_1)$ (ru/uot)				
	$A = 0.001$	$A = 0.01$	$A = 0.1$	$A = 1.0$	$A = 10$
Exact	0.0006	0.0065	0.0610	0.4142	1.688
Approximate	0.0006	0.0065	0.0610	0.4261	1.652
Relative Error (%)	0.00	0.00	0.00	2.87	2.13

**Table 5.2:** Comparison of the exact and the approximate link reward rates.

link allocates resources with a policy  $\Pi = \Pi_1$  that consists of 36 decisions. The exact and the approximate reward models for this single-link network have already been studied in Chapter 4. In Section 5.2.1 we will use these models to present an analytical comparison of the performance when the policy  $\Pi_1$  is used under different traffic loads. Then in Section 5.2.2 simulations will be used to determine the reward losses caused by both methods when the PIA implements  $\Pi_1$  as starting policy. These losses are compared against the performance achieved by a basic CAC scheme, which only rejects connections that violate the contiguity and capacity constraints.

### 5.2.1 Exact and Approximate Long-Term Reward Gains

For the link in Fig. 5.1, we define in Table 5.1 the parameters used to compare the solutions of the exact and the approximate reward models under different traffic loads  $A$ . This traffic is calculated as:

$$A = A_1 + A_2 \quad (5.4)$$

where  $A_1 = \lambda_1/\mu_1$  and  $A_2 = \lambda_2/\mu_2$ . Class-1 and class-2 connections bring, respectively, immediate (or short-term) rewards of  $r_1 = 2.5$  (ru) and  $r_2 = 4.0$  (ru). The performance is calculated by solving the linear systems derived in Section 4.2 - Chapter 4, for the single-link network. In particular, in chapter 4, the exact model is defined by the linear system in Fig. 4.4, whereas the approximate approach is given by the system in Fig. 4.5. Two important remarks are needed regarding these linear systems. First, since we are studying a single-link network, the mathematical notation is kept simple by dropping off the superscript  $l$  from  $R(\Pi_1)$ ,  $A$ ,  $A_j$ ,  $\lambda_j$  and  $r_j$ , i.e. there is no need to differentiate among links. Secondly, as explained in Chapter 4, the linear system in Fig. 4.4 defines the exact state-dependent reward model for the link in Fig. 5.1. The reason is that it solely represents the reward process of a single-link network.

In Table 5.2 and Table 5.3 we summarize the solutions to both linear systems. Table 5.2 shows the exact and the approximate link reward rates  $R(\Pi_1)$ . These rates are compared by calculating the relative approximate error with Equation (5.3) – the error is calculated relative to the exact rate. The results show that for all traffic loads, the macrostate-dependent approach gives good estimates for  $R(\Pi_1)$ . On the other hand, in Table 5.3 a comparison is made for the transient reward values calculated by each approach. The six macrostates in which the link can be observed are described by the transient reward values  $v(\mathbf{n}_i, \Pi_1)$  obtained from the macrostate-dependent model. Recall that in the approximate PIA, the value  $v(\mathbf{n}_i, \Pi_1)$  is used as an approximation to the exact reward values  $v(\mathbf{x}_i, \Pi_1)$  of the states in the set  $\widehat{X}_n^l$  that define  $\mathbf{n}_i$ . In Fig. 5.1 the six sets  $\widehat{X}_n^l$  are defined for the single-link network. For these sets, in Table 5.3 the exact values are presented with the macrostate approximation. For example, the reward value  $v(\mathbf{n}_2, \Pi_1)$  - where  $\mathbf{n}_2 = (1,0)$  - is an approximation to the values  $v(\mathbf{x}_i, \Pi_1)$  of the states in  $\widehat{X}_{(1,0)}^l = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ . The results show that in all cases the values  $v(\mathbf{n}_i, \Pi_1)$  provide a good approximation to the exact values  $v(\mathbf{x}_i, \Pi_1)$ . However, to better quantify and judge the accuracy of the approximation, let us recall that the importance of calculating the transient values in either model is to determine the long-term reward gains. These gains are only earned when a connection arrives and gets admission to the network. As studied in Chapter 4, a macrostate transition  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$  can be caused by arrivals that trigger any of the possible link state transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , such that  $\mathbf{x}^l \in \widehat{X}_n^l$  and  $\mathbf{y}^l \in \widehat{X}_{\mathbf{n}+\delta_j^l}^l$ . Thus, if a class- $j$  connection is admitted

Type of value	Value Definition	Transient Reward Values $v(\mathbf{x}_i, \Pi_1)$ and $v(\mathbf{n}_i, \Pi_1)$ (ru)				
		$A = 0.001$	$A = 0.01$	$A = 0.1$	$A = 1.0$	$A = 10$
Exact value state $\mathbf{x}_1$	$v(\mathbf{x}_1, \Pi_1)$	0.047	0.338	1.867	6.432	36.219
Approximate value	$v(\mathbf{n}_1, \Pi_1)$	0.047	0.338	1.867	6.046	36.659
Exact value state $\mathbf{x}_2$	$v(\mathbf{x}_2, \Pi_1)$	2.547	2.838	4.366	8.904	38.409
Exact value state $\mathbf{x}_3$	$v(\mathbf{x}_3, \Pi_1)$	2.547	2.835	4.334	8.739	38.526
Exact value state $\mathbf{x}_4$	$v(\mathbf{x}_4, \Pi_1)$	2.547	2.835	4.334	8.755	39.062
Exact value state $\mathbf{x}_5$	$v(\mathbf{x}_5, \Pi_1)$	2.547	2.835	4.334	8.746	38.499
Exact value state $\mathbf{x}_6$	$v(\mathbf{x}_6, \Pi_1)$	2.547	2.841	4.388	9.082	38.936
Approximate value	$v(\mathbf{n}_2, \Pi_1)$	2.547	2.838	4.366	8.525	38.775
Exact value state $\mathbf{x}_7$	$v(\mathbf{x}_7, \Pi_1)$	5.047	5.335	6.834	11.240	41.294
Exact value state $\mathbf{x}_8$	$v(\mathbf{x}_8, \Pi_1)$	5.047	5.334	6.822	11.124	40.111
Exact value state $\mathbf{x}_9$	$v(\mathbf{x}_9, \Pi_1)$	5.047	5.338	6.861	11.395	41.251
Exact value state $\mathbf{x}_{10}$	$v(\mathbf{x}_{10}, \Pi_1)$	5.047	5.332	6.806	11.041	40.169
Exact value state $\mathbf{x}_{11}$	$v(\mathbf{x}_{11}, \Pi_1)$	5.047	5.335	6.833	11.209	40.388
Exact value state $\mathbf{x}_{12}$	$v(\mathbf{x}_{12}, \Pi_1)$	5.047	5.336	6.845	11.325	41.475
Approximate value	$v(\mathbf{n}_3, \Pi_1)$	5.047	5.336	6.850	10.937	41.259
Exact value state $\mathbf{x}_{13}$	$v(\mathbf{x}_{13}, \Pi_1)$	7.547	7.834	9.328	13.686	43.278
Approximate value	$v(\mathbf{n}_4, \Pi_1)$	7.547	7.834	9.331	13.307	43.162
Exact value state $\mathbf{x}_{14}$	$v(\mathbf{x}_{14}, \Pi_1)$	4.043	4.297	5.484	8.404	35.884
Exact value state $\mathbf{x}_{15}$	$v(\mathbf{x}_{15}, \Pi_1)$	4.041	4.274	5.263	6.647	24.821
Exact value state $\mathbf{x}_{16}$	$v(\mathbf{x}_{16}, \Pi_1)$	4.043	4.297	5.484	8.393	35.684
Approximate value	$v(\mathbf{n}_5, \Pi_1)$	4.043	4.297	5.484	7.893	36.409
Exact value state $\mathbf{x}_{17}$	$v(\mathbf{x}_{17}, \Pi_1)$	6.543	6.795	7.966	10.726	37.263
Exact value state $\mathbf{x}_{18}$	$v(\mathbf{x}_{18}, \Pi_1)$	6.543	6.795	7.964	10.700	37.038
Approximate value	$v(\mathbf{n}_6, \Pi_1)$	6.543	6.795	7.964	10.201	37.758

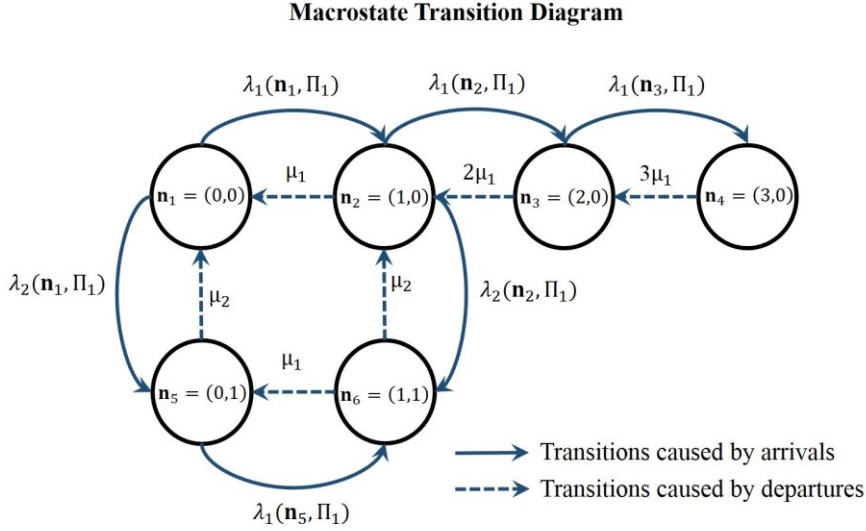
**Table 5.3:** Comparison of the exact and the approximate transient reward values.

in state  $\mathbf{x}^l$ , such that it causes the transition  $\mathbf{x}^l \rightarrow \mathbf{y}^l$ , it brings a long-term reward gain  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) = v(\mathbf{y}^l, \Pi) - v(\mathbf{x}^l, \Pi)$ . Under the macrostate-dependent model, the gains of all transitions  $\mathbf{x}^l \rightarrow \mathbf{y}^l$  that cause  $\mathbf{n}^l \rightarrow \mathbf{n}^l + \delta_j^l$  are assumed to be identical and thus approximated as  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi) \approx g_j^l(\mathbf{n}^l, \Pi) = v(\mathbf{n}^l + \delta_j^l, \Pi) - v(\mathbf{n}^l, \Pi)$ . Therefore, to compare both methods, it would be more reasonable to analyse how well  $g_j^l(\mathbf{n}^l, \Pi)$  approximates  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$ . For that, consider the macrostate transition diagram depicted in Fig. 5.2. In the diagram, it is seen that arrivals may cause six possible macrostate transitions in the single-link network. They are  $\mathbf{n}_1 \rightarrow \mathbf{n}_2$ ,  $\mathbf{n}_2 \rightarrow \mathbf{n}_3$ ,  $\mathbf{n}_3 \rightarrow \mathbf{n}_4$ ,  $\mathbf{n}_5 \rightarrow \mathbf{n}_6$ ,  $\mathbf{n}_1 \rightarrow \mathbf{n}_5$  and  $\mathbf{n}_2 \rightarrow \mathbf{n}_6$ . Each transition yields a long-term reward gain  $g_j(\mathbf{n}_i, \Pi_1)$ . These gains are defined in Fig. 5.2 and are calculated with the transient reward values  $v(\mathbf{n}_i, \Pi_1)$  listed in Table 5.3. (In order to keep the mathematical notation simple, in Fig. 5.2 we drop from  $g_j^l(\mathbf{y}^l, \mathbf{x}^l, \Pi)$  and  $g_j^l(\mathbf{n}^l, \Pi)$  the superscript  $l$ . Furthermore, the states  $\mathbf{x}^l$  and  $\mathbf{y}^l$  are denoted by the corresponding states  $\mathbf{x}_i$ , whereas  $\mathbf{n}^l$  is denoted by  $\mathbf{n}_i$ .)

Figure 5.2 further defines the state transitions that cause each macrostate transition. For example,  $\mathbf{n}_2 \rightarrow \mathbf{n}_6$ , i.e.  $(1,0) \rightarrow (1,1)$ , may occur if a class-2 request arrives when the link is in state  $\mathbf{x}_2$  or  $\mathbf{x}_6$ . Therefore, according to the policy decisions  $\Pi_1(\mathbf{x}_2, 2)$  and  $\Pi_1(\mathbf{x}_6, 2)$  in Fig. 5.1, we have that  $\mathbf{n}_2 \rightarrow \mathbf{n}_6$  is observed if and only if either transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$  or  $\mathbf{x}_6 \rightarrow \mathbf{x}_{17}$  occurs. These transitions yield, respectively, the long-term rewards  $g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = v(\mathbf{x}_{18}, \Pi_1) - v(\mathbf{x}_2, \Pi_1)$  and  $g_2(\mathbf{x}_{17}, \mathbf{x}_6, \Pi_1) = v(\mathbf{x}_{17}, \Pi_1) - v(\mathbf{x}_6, \Pi_1)$ . In principle, these two gains need not be alike. However, the macrostate-dependent reward model approximates them as  $g_2(\mathbf{n}_2, \Pi_1) = v(\mathbf{n}_6, \Pi_1) - v(\mathbf{n}_2, \Pi_1)$  - see Fig. 5.2.

In Tables 5.4-5.9 we summarize for each macrostate transition the corresponding gain  $g_j(\mathbf{n}_i, \Pi_1)$  as well as the exact gains  $g_j(\mathbf{x}_k, \mathbf{x}_i, \Pi_1)$  that cause the transition. The gains are shown for each traffic load and are calculated from the reward values in Table 5.3. The exact gains are compared with  $g_j(\mathbf{n}_i, \Pi_1)$  by calculating the relative approximate error with Equation (5.3) - the error is calculated relative to the exact gains. The results show that the macrostate-dependent model provides good approximations, especially at low traffic loads. (Note that for all macrostate transitions, the relative approximate error increases with the traffic load.)

Two important attributes of the reward models are revealed by the results. First, in most traffic loads the inequalities  $g_j(\mathbf{n}_i, \Pi_1) \leq r_j$  and  $g_j(\mathbf{x}_k, \mathbf{x}_i, \Pi_1) \leq r_j$  are verified. In particular, as seen in Tables 5.4-5.9, the equalities  $g_j(\mathbf{n}_i, \Pi_1) = r_j$ ,  $g_j(\mathbf{x}_k, \mathbf{x}_i, \Pi_1) = r_j$  are satisfied at low traffic loads. The reason is that



**Macrostate and State Transitions Caused by Connection Arrivals**

Macrostate Transition	Approximate Long-Term Reward Gain	State Transition	Exact Long-Term Reward Gain	Cause of the Transition
$n_1 \rightarrow n_2$	$g_1(n_1, \Pi_1) = v(n_2, \Pi_1) - v(n_1, \Pi_1)$	$\chi_1 \rightarrow \chi_2$	$g_1(\chi_2, \chi_1, \Pi_1) = v(\chi_2, \Pi_1) - v(\chi_1, \Pi_1)$	class-1 arrival
$n_2 \rightarrow n_3$	$g_1(n_2, \Pi_1) = v(n_3, \Pi_1) - v(n_2, \Pi_1)$	$\chi_2 \rightarrow \chi_7$	$g_1(\chi_7, \chi_2, \Pi_1) = v(\chi_7, \Pi_1) - v(\chi_2, \Pi_1)$	class-1 arrival
		$\chi_3 \rightarrow \chi_{10}$	$g_1(\chi_{10}, \chi_3, \Pi_1) = v(\chi_{10}, \Pi_1) - v(\chi_3, \Pi_1)$	
		$\chi_4 \rightarrow \chi_7$	$g_1(\chi_7, \chi_4, \Pi_1) = v(\chi_7, \Pi_1) - v(\chi_4, \Pi_1)$	
		$\chi_5 \rightarrow \chi_8$	$g_1(\chi_8, \chi_5, \Pi_1) = v(\chi_8, \Pi_1) - v(\chi_5, \Pi_1)$	
$n_3 \rightarrow n_4$	$g_1(n_3, \Pi_1) = v(n_4, \Pi_1) - v(n_3, \Pi_1)$	$\chi_7 \rightarrow \chi_{13}$	$g_1(\chi_{13}, \chi_7, \Pi_1) = v(\chi_{13}, \Pi_1) - v(\chi_7, \Pi_1)$	class-1 arrival
		$\chi_9 \rightarrow \chi_{13}$	$g_1(\chi_{13}, \chi_9, \Pi_1) = v(\chi_{13}, \Pi_1) - v(\chi_9, \Pi_1)$	
		$\chi_{12} \rightarrow \chi_{13}$	$g_1(\chi_{13}, \chi_{12}, \Pi_1) = v(\chi_{13}, \Pi_1) - v(\chi_{12}, \Pi_1)$	
$n_5 \rightarrow n_6$	$g_1(n_5, \Pi_1) = v(n_6, \Pi_1) - v(n_5, \Pi_1)$	$\chi_{14} \rightarrow \chi_{17}$	$g_1(\chi_{17}, \chi_{14}, \Pi_1) = v(\chi_{17}, \Pi_1) - v(\chi_{14}, \Pi_1)$	class-1 arrival
		$\chi_{16} \rightarrow \chi_{18}$	$g_1(\chi_{18}, \chi_{16}, \Pi_1) = v(\chi_{18}, \Pi_1) - v(\chi_{16}, \Pi_1)$	
$n_1 \rightarrow n_5$	$g_2(n_1, \Pi_1) = v(n_5, \Pi_1) - v(n_1, \Pi_1)$	$\chi_1 \rightarrow \chi_{14}$	$g_2(\chi_{14}, \chi_1, \Pi_1) = v(\chi_{14}, \Pi_1) - v(\chi_1, \Pi_1)$	class-2 arrival
$n_2 \rightarrow n_6$	$g_2(n_2, \Pi_1) = v(n_6, \Pi_1) - v(n_2, \Pi_1)$	$\chi_2 \rightarrow \chi_{18}$	$g_2(\chi_{18}, \chi_2, \Pi_1) = v(\chi_{18}, \Pi_1) - v(\chi_2, \Pi_1)$	class-2 arrival
		$\chi_6 \rightarrow \chi_{17}$	$g_2(\chi_{17}, \chi_6, \Pi_1) = v(\chi_{17}, \Pi_1) - v(\chi_6, \Pi_1)$	

**Figure 5.2:** Definition of the exact and the approximate reward gains for the single-link network.

A (Erl)	$g_1(n_1, \Pi_1)$ (ru)	$g_1(\chi_2, \chi_1, \Pi_1)$	
		Gain (ru)	Error (%)
0.001	2.500	2.500	0.00
0.010	2.500	2.500	0.00
0.100	2.499	2.499	0.00
1.000	2.479	2.472	0.28
10.00	2.116	2.190	3.38

**Table 5.4:** Relative approximate error long-term reward gain  $g_1(n_1, \Pi_1)$ .

A (Erl)	$g_1(\mathbf{n}_2, \Pi_1)$ (ru)	$g_1(\mathcal{X}_7, \mathcal{X}_2, \Pi_1)$		$g_1(\mathcal{X}_{10}, \mathcal{X}_3, \Pi_1)$		$g_1(\mathcal{X}_7, \mathcal{X}_4, \Pi_1)$		$g_1(\mathcal{X}_8, \mathcal{X}_5, \Pi_1)$		$g_1(\mathcal{X}_9, \mathcal{X}_6, \Pi_1)$	
		Gain (ru)	Error (%)	Gain (ru)	Error (%)	Gain (ru)	Error (%)	Gain (ru)	Error (%)	Gain (ru)	Error (%)
0.001	2.500	2.500	0.00	2.500	0.00	2.500	0.00	2.500	0.00	2.500	0.00
0.010	2.498	2.497	0.04	2.497	0.04	2.500	0.08	2.499	0.04	2.497	0.04
0.100	2.484	2.468	0.65	2.472	0.49	2.500	0.64	2.488	0.16	2.473	0.44
1.000	2.412	2.336	3.25	2.302	4.78	2.485	2.94	2.378	1.43	2.313	4.28
10.00	2.484	2.885	13.90	1.643	51.19	2.232	11.29	1.612	54.09	2.315	7.30

Table 5.5: Relative approximate error long-term reward gain  $g_1(\mathbf{n}_2, \Pi_1)$ .

A (Erl)	$g_1(\mathbf{n}_3, \Pi_1)$ (ru)	$g_1(\mathcal{X}_{13}, \mathcal{X}_7, \Pi_1)$		$g_1(\mathcal{X}_{13}, \mathcal{X}_9, \Pi_1)$		$g_1(\mathcal{X}_{13}, \mathcal{X}_{12}, \Pi_1)$	
		Gain (ru)	Error (%)	Gain (ru)	Error (%)	Gain (ru)	Error (%)
0.001	2.500	2.500	0.00	2.500	0.00	2.500	0.00
0.010	2.498	2.499	0.04	2.496	0.08	2.498	0.00
0.100	2.481	2.494	0.52	2.467	0.57	2.483	0.08
1.000	2.370	2.446	3.11	2.291	3.45	2.361	0.38
10.00	1.903	1.984	4.08	2.027	6.12	1.803	5.55

Table 5.6: Relative approximate error long-term reward gain  $g_1(\mathbf{n}_3, \Pi_1)$ .

A (Erl)	$g_1(\mathbf{n}_5, \Pi_1)$ (ru)	$g_1(\mathcal{X}_{17}, \mathcal{X}_{14}, \Pi_1)$		$g_1(\mathcal{X}_{18}, \mathcal{X}_{16}, \Pi_1)$	
		Gain (ru)	Error (%)	Gain (ru)	Error (%)
0.001	2.500	2.500	0.00	2.500	0.00
0.010	2.498	2.498	0.00	2.498	0.00
0.100	2.480	2.482	0.08	2.480	0.00
1.000	2.308	2.322	0.60	2.307	0.04
10.00	1.349	1.379	2.18	1.354	0.37

Table 5.7: Relative approximate error long-term reward gain  $g_1(\mathbf{n}_5, \Pi_1)$ .

A (Erl)	$g_2(\mathbf{n}_1, \Pi_1)$ (ru)	$g_2(\mathcal{X}_{14}, \mathcal{X}_1, \Pi_1)$	
		Gain (ru)	Error (%)
0.001	3.996	3.996	0.00
0.010	3.959	3.959	0.00
0.100	3.617	3.617	0.00
1.000	1.847	1.972	6.34
10.00	-0.250	-0.335	25.34

Table 5.8: Relative approximate error long-term reward gain  $g_2(\mathbf{n}_1, \Pi_1)$ .

A (Erl)	$g_2(\mathbf{n}_2, \Pi_1)$ (ru)	$g_2(\mathcal{X}_{17}, \mathcal{X}_6, \Pi_1)$		$g_2(\mathcal{X}_{18}, \mathcal{X}_2, \Pi_1)$	
		Gain (ru)	Error (%)	Gain (ru)	Error (%)
0.001	3.996	3.996	0.00	3.996	0.00
0.010	3.957	3.954	0.08	3.957	0.00
0.100	3.598	3.578	0.56	3.598	0.00
1.000	1.676	1.644	1.95	1.796	6.68
10.00	-1.017	-1.673	39.2	-1.371	25.8

Table 5.9: Relative approximate error long-term reward gain  $g_2(\mathbf{n}_2, \Pi_1)$ .

$\lambda_1$ (con/uot)	$\lambda_2$ (con/uot)	$\mu_1$ (con/uot)	$\mu_2$ (con/uot)	$r_1$ (ru)	$r_2$ (ru)
variable	variable	1.0	0.1	2.5	4.0

**Table 5.10:** Parameter settings for performance evaluation of the PIA in the single-link network.

when the traffic load is low, an admitted class- $j$  connection is less likely to prevent the admission of more “profitable” connections, i.e. the probability that during the connection holding time  $\mu_j^{-1}$  the carried connection blocks new arrivals is negligible. Therefore, the admission of class-1 and class-2 connections yield, respectively, long-term reward gains  $g_1(\mathbf{n}_i, \Pi_1) \approx g_1(\mathbf{x}_k, \mathbf{x}_i, \Pi_1) \approx r_1 = 2.5$  and  $g_2(\mathbf{n}_i, \Pi_1) \approx g_2(\mathbf{x}_k, \mathbf{x}_i, \Pi_1) \approx r_2 = 4.0$ . As the traffic load increases, most of the gains become lower than the reward parameters. The higher the traffic load, the higher the probability that a carried connection blocks more valuable traffic, thereby  $g_j(\mathbf{n}_i, \Pi_1) < r_j$  and  $g_j(\mathbf{x}_k, \mathbf{x}_i, \Pi_1) < r_j$ . As an example, consider the transition  $\mathbf{n}_2 \rightarrow \mathbf{n}_6$  which was previously explained. The results for this case are summarized in Table 5.9. When the link is offered a load of  $A = 0.01$  (Erlangs), if a class-2 arrival gets admission in state  $\mathbf{x}_2$ , then the transition  $\mathbf{x}_2 \rightarrow \mathbf{x}_{18}$  occurs and an immediate reward of  $r_2 = 4.0$  (ru) is earned. However, owing to the harmful effects that this connection may cause, the link will effectively earn a long-term reward of  $g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = 3.957$  (ru), which is comparable to  $r_2$ . This gain is approximated in the macrostate-dependent reward model as  $g_2(\mathbf{n}_2, \Pi_1) = 3.957$  (ru). If the load rises to  $A = 1.0$  Erlangs, the link now earns from the same connection a reward of  $g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) = 1.796$  (ru) – note that this gain is approximated as  $g_2(\mathbf{n}_2, \Pi_1) = 1.676$  (ru). This connection becomes more harmful as the  $b_2 = 4$  slots that it seizes may block the admission of connections that can bring more reward to the link.

The second attribute revealed by the results is that (as expected) for a given macrostate transition, the exact gains  $g_j(\mathbf{x}_k, \mathbf{x}_i, \Pi_1)$  resulting from the transition are in most cases comparable, but not alike. For instance, in Table 5.9 we can see that the macrostate transition  $\mathbf{n}_2 \rightarrow \mathbf{n}_6$  has gains  $g_2(\mathbf{x}_{18}, \mathbf{x}_2, \Pi_1) \neq g_2(\mathbf{x}_{17}, \mathbf{x}_6, \Pi_1)$  which are approximated by  $g_2(\mathbf{n}_2, \Pi_1)$ . However, the results show that for the six macrostate transitions, the exact gains tend to be similar, especially in low traffic load regions.

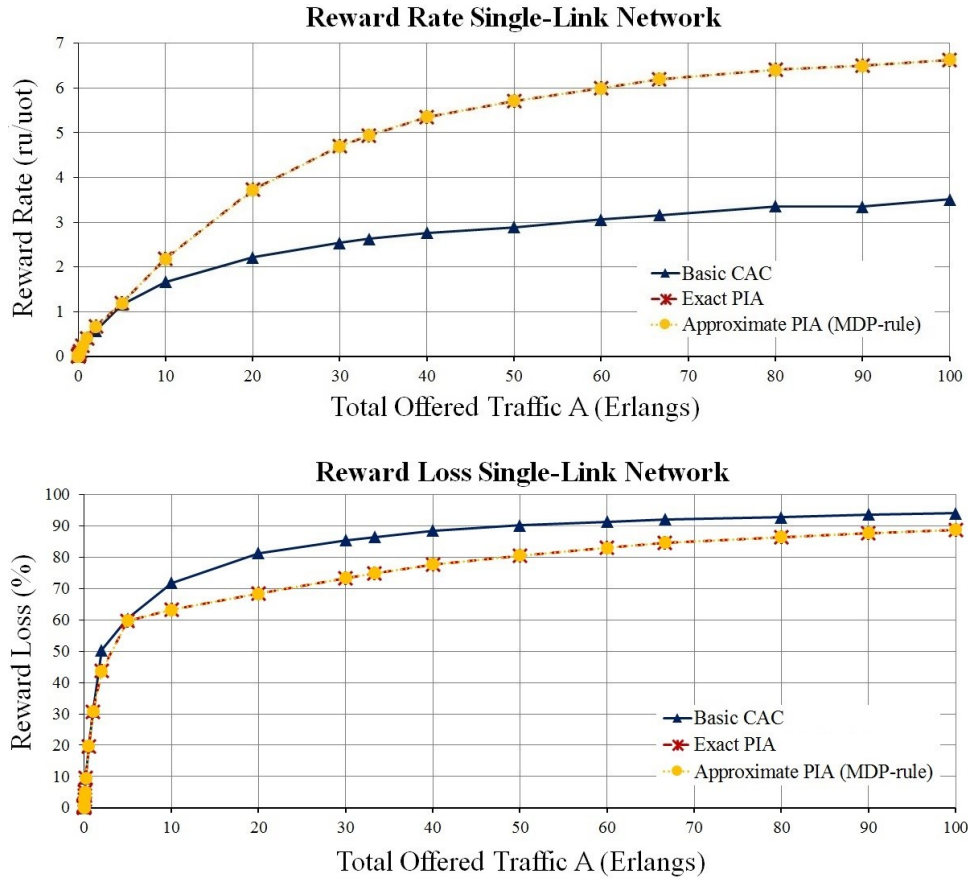
An important remark needs to be clarified regarding the relative approximate error at high traffic loads. In most cases, the error is small, however the macrostate-dependent model tends to overestimate the long-term gains for transitions  $\mathbf{x}_i \rightarrow \mathbf{x}_k$  in which the destination state  $\mathbf{x}_k$  blocks arrivals of either class. Consider as an example the results in Table 5.5 for  $A = 10$  Erlangs, which show the gains for the transition  $\mathbf{n}_2 \rightarrow \mathbf{n}_3$ . The gain  $g_1(\mathbf{n}_2, \Pi_1)$  approximates the exact gains  $g_1(\mathbf{x}_k, \mathbf{x}_i, \Pi_1)$  of the five state transitions that cause  $\mathbf{n}_2 \rightarrow \mathbf{n}_3$ . From the results, the gain  $g_1(\mathbf{n}_2, \Pi_1) = 2.484$  (ru) is a reasonable approximation to the exact gains of the three state transitions  $\mathbf{x}_2 \rightarrow \mathbf{x}_7$ ,  $\mathbf{x}_4 \rightarrow \mathbf{x}_7$  and  $\mathbf{x}_6 \rightarrow \mathbf{x}_9$ . Note that the destination states  $\mathbf{x}_7 = (1, \infty, 1, \infty, 0, 0)$  and  $\mathbf{x}_9 = (1, \infty, 0, 0, 1, \infty)$  allow the admission of class-1 arrivals as both have two free adjacent slots. However,  $g_1(\mathbf{n}_2, \Pi_1)$  overestimates the exact gains  $g_1(\mathbf{x}_{10}, \mathbf{x}_3, \Pi_1) = 1.643$  (ru) and  $g_1(\mathbf{x}_8, \mathbf{x}_5, \Pi_1) = 1.612$  (ru) of the transitions  $\mathbf{x}_3 \rightarrow \mathbf{x}_{10}$  and  $\mathbf{x}_5 \rightarrow \mathbf{x}_8$ , respectively. The destination states  $\mathbf{x}_8 = (1, \infty, 0, 1, \infty, 0)$  and  $\mathbf{x}_{10} = (0, 1, \infty, 1, \infty, 0)$ , in contrast to  $\mathbf{x}_7$  and  $\mathbf{x}_9$ , block the admission of class-1 requests. For this reason, the exact gains  $g_1(\mathbf{x}_{10}, \mathbf{x}_3, \Pi_1)$  and  $g_1(\mathbf{x}_8, \mathbf{x}_5, \Pi_1)$  are less than  $g_1(\mathbf{x}_7, \mathbf{x}_2, \Pi_1)$ ,  $g_1(\mathbf{x}_7, \mathbf{x}_4, \Pi_1)$  and  $g_1(\mathbf{x}_9, \mathbf{x}_6, \Pi_1)$ . Hence,  $\mathbf{x}_8$  and  $\mathbf{x}_{10}$  are more harmful than  $\mathbf{x}_7$  and  $\mathbf{x}_9$ . This effect is more noticeable at high traffic loads because in those states the link is more likely to block arrivals of either class.

The analytical study presented for the single-link network shows that, in the considered traffic load region, the macrostate-dependent model is a reasonable approach to modelling the link reward process. Since for multi-link networks the assumption is made that each link behaves independently, it can be argued that - in those networks - the macrostate-dependent approach can be a good approximation for each link reward process. We defer this discussion to Section 5.3, where the case of multi-link networks is tackled. In the meantime, let us complement the analytical study by analysing the performance of the exact and the approximate PIA when  $\Pi_1$  is used as the starting policy.

## 5.2.2 Adaptive and State-Dependent Connection Admission Control

To assess the performance of the exact and the approximate PIA for the single-link network in Fig. 5.1, simulations were set up with the parameters  $\lambda_j$ ,  $\mu_j$ , and  $r_j$  defined in Table 5.10. For different traffic loads  $A$ , the mean link reward rate  $R(\Pi)$  and the mean the link reward loss  $R_L(\Pi)$  are evaluated for each PIA variant. By setting  $\lambda_1 = \lambda_2 = \lambda$ , the traffic loads are generated from Equation (5.4) by varying  $\lambda$ . Therefore, since  $A_1 = \lambda/\mu_1$  and  $A_2 = \lambda/\mu_2$ , then  $A = \lambda \cdot (\mu_1^{-1} + \mu_2^{-1})$ , where  $\mu_1$  and  $\mu_2$  are fixed and equal to the values defined in Table 5.10.

As starting policy, both PIA variants use the policy  $\Pi_1$  shown in Fig 5.1. Furthermore, the approximate PIA implements the MDP rule and uses first-fit (FF) as spectrum allocation algorithm [TAK<sup>+</sup>14]. (Recall that in a spectrum grid with  $C_l$  slots indexed as  $1, 2, \dots, C_l$ , first-fit assigns the available  $b_j$  adjacent slots



**Figure 5.3:** Simulation results for the basic CAC scheme, the exact and the approximate PIA.

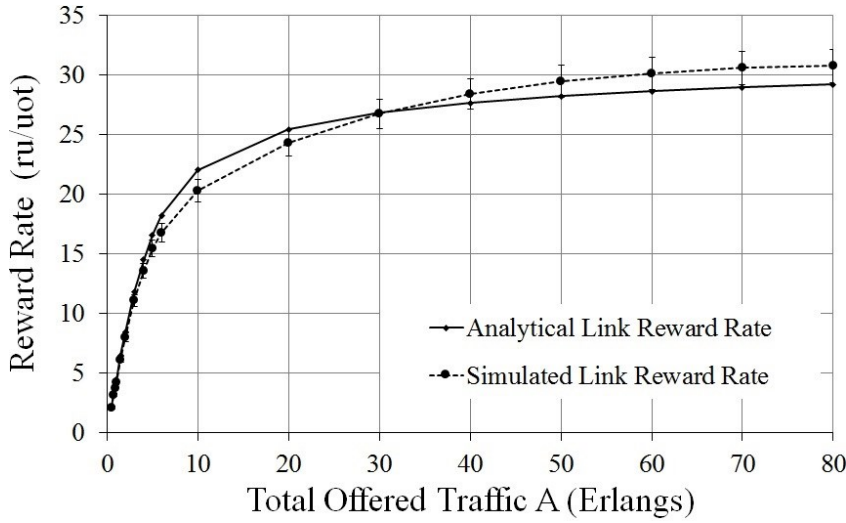
with the lowest indices.) Both methods are further compared against a basic CAC scheme that only rejects connections that violate the contiguity and capacity constraints (i.e. connections are admitted if resources are available disregarding any harmful effect that they may cause).

Figure 5.3 depicts the simulation results, where the reward rates and losses of the approximate PIA correspond to the policy calculated in the fifth iteration cycle, i.e. the cycle that meets Equation (5.1). The results show that both PIA algorithms have similar performance, which indicates that - as with the results in Section 5.2.1 - the macrostate-dependent PIA is a good approximation to the exact model. The average relative approximate error of the reward losses (calculated over all traffic loads) is 0.47%. The advantage of the approximate method is its reduced complexity (note that the exact PIA involves the solution of a system with 18 linear equations, whereas in the macrostate-dependent variant the system has only size six). In Fig. 5.3 it is also observed that both PIA strategies outperform the basic CAC scheme, especially for high traffic loads. This difference stems from the admission decision rules defined by the resource allocation algorithms. The basic CAC approach accepts connections whenever resources are available for them, thereby any harmful effect that an admitted connection may cause is omitted. On the other hand, the PIA variants exclude and reject connection requests that, in spite of the availability of resources, yield negative reward gains. Thus, only traffic that increases the long-term reward earned by the link is admitted.

To further study the approximate PIA, let us consider a larger single-link network with a capacity of  $C_l = 32$  slots that serves  $J = 2$  connection classes. Class-1 and class-2 have, respectively, bandwidths and reward parameters  $b_1 = 2$  (slots),  $r_1 = 2$  (ru) and  $b_2 = 8$  (slots),  $r_2 = 6.5$  (ru). Two evaluation scenarios are defined in Table 5.11 for this link. In both scenarios the PIA implements first-fit as spectrum allocation algorithm, and the basic CAC scheme is implemented as starting policy  $\Pi_0$  (using first-fit as well). This means that in the first iteration cycle, connection requests are always admitted if resources are available for them. From the second cycle onwards, the PIA calculates and executes new policies according to the MDP rule. For both scenarios, a performance comparison with the exact PIA is infeasible given that this approach is unsolvable for the considered link (the state-space contains 5054773 states, which defines a linear system that cannot be solved online). In contrast, the macrostate-space has 69 macrostates, thereby allowing the implementation of the approximate PIA. The aim of scenario 1 is to evaluate the link reward rate  $R(\Pi)$  attained by the approximate PIA under different traffic loads  $A$ . Since a comparison with the exact PIA is not possible, we compare  $R(\Pi)$  with the analytical rate obtained by solving the linear system defined by the policy  $\Pi$ . The traffic loads  $A = \lambda_1/\mu_1 + \lambda_2/\mu_2$  are determined by setting  $\lambda_1 = \lambda_2 = \lambda$ . Therefore,  $\lambda$  is varied so that the traffic  $A$  changes from 0.5 to 80 Erlangs. On the other hand, the goal of

Parameter	Scenario 1	Scenario 2
$C_l$	32 (slots)	32 (slots)
$J$	2 (classes)	2 (classes)
$\lambda_1$	variable	0.75 (con/uot)
$\lambda_2$	variable	0.75 (con/uot)
$\mu_1$	1 (con/uot)	1 (con/uot)
$\mu_2$	1 (con/uot)	Variable
$b_1$	2 (slots)	2 (slots)
$b_2$	8 (slots)	8 (slots)
$r_1$	2 (ru/con)	2 (ru/con)
$r_2$	6.5 (ru/con)	6.5 (ru/con)
rule	MDP	MDP
Spectrum allocation	first-fit	first-fit

**Table 5.11:** Parameter settings for performance evaluation of the approximate PIA [RB16b].



**Figure 5.4:** Analytical and simulated average link reward rate vs. offered traffic [RB16b].

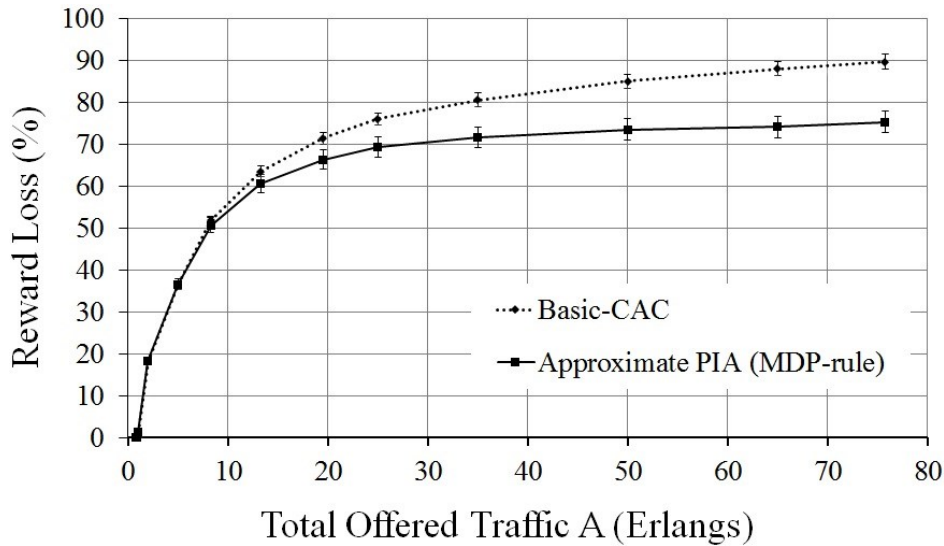
scenario 2 is to assess the approximate PIA by comparing its reward losses with those achieved by the basic CAC scheme. The assessment is performed for different offered traffic loads  $A$ , which are obtained by varying  $\mu_2$  from 0.01 to 20 (con/uot). The results of this study are published in [RB16b].

Figure 5.4 depicts the analytical and the simulated average link reward rate for scenario 1. The results refer to the performance achieved in the eight iteration cycle, which is the cycle that meets the stopping criterion in Equation (5.1). The analytical link reward rate is calculated by solving the linear system (with 69 equations) defined by the policy  $\Pi$  executed in that cycle. Such a system is solved by using the parameters defined in Table 5.1. Moreover, in the simulations, the link reward rate is estimated from the observed carried traffics as  $R(\Pi) = \sum_j r_j \cdot \mu_j \cdot A c_j$ . Notice that this expression was derived in Chapter 3, and corresponds to Equation (3.4). The results indicate that the approximate PIA closely matches the analytical results. More specifically, in Fig. 5.4 it is seen that most analytical link reward rates are within the 95% confidence intervals. The average relative approximate error (calculated over all simulated traffic loads) is 5.55%. This allows us to infer that the online estimation method of arrival rates (as implemented by the approximate PIA) gives reliable estimates. To prove this, consider the comparison between the exact and the estimated arrival rates  $\lambda_j$  shown in Table 5.12. Therein, the exact rates  $\lambda_1 = \lambda_2 = \lambda$  are the exogenous arrival rates that yield the traffic loads  $A$ . The estimated rates correspond to  $\lambda_1^l(\Pi)$  and  $\lambda_2^l(\Pi)$ , which are determined by the procedure outlined in Chapter 4. (Note that as explained in chapter 4, the rates  $\lambda_j^l(\Pi)$  are estimated with Equation (4.35) from online measurements.) The relative approximate errors in Table 5.12 evince the good accuracy of the estimation procedure. Besides, it shows that the PIA adapts well to changes in the traffic load. This explains why in Fig. 5.4 the approximate PIA fits well the analytical results.

For scenario 2, Fig. 5.5 shows the average simulated link reward loss as function of the total offered traffic for the basic CAC scheme and the approximate PIA. For the PIA, the reward losses are those obtained in the eight iteration cycle. As with the small single-link network, the results show that the PIA yields lower reward losses compared to the basic CAC scheme, especially for high traffic loads. Although

A (Erl)	$\lambda_1$ (con/uot)			$\lambda_2$ (con/uot)		
	Exact	Estimated	Error (%)	Exact	Estimated	Error (%)
0.50	0.250	0.2507	0.28	0.250	0.2488	0.48
0.75	0.375	0.3749	0.03	0.375	0.3760	0.27
0.90	0.450	0.4493	0.16	0.450	0.4474	0.58
1.00	0.500	0.5008	0.16	0.500	0.4986	0.28
1.50	0.750	0.7451	0.65	0.750	0.7434	0.88
2.00	1.000	0.9938	0.62	1.000	0.9911	0.89
3.00	1.500	1.4670	2.20	1.500	1.4850	1.00
4.00	2.000	1.9970	0.15	2.000	1.9930	0.35
5.00	2.500	2.4840	0.64	2.500	2.4800	0.80
6.00	3.000	3.0100	0.33	3.000	2.9720	0.93
10.00	5.000	4.9810	0.38	5.000	5.0400	0.80
20.00	10.00	10.07	0.70	10.00	9.8970	1.03
30.00	15.00	14.94	0.40	15.00	15.67	4.47
40.00	20.00	20.10	0.50	20.00	20.93	4.65
50.00	25.00	24.99	0.04	25.00	23.68	5.28
60.00	30.00	29.68	1.07	30.00	31.81	6.03
70.00	35.00	34.67	0.94	35.00	34.30	2.0
80.00	40.00	40.73	1.82	40.00	42.10	5.25

**Table 5.12:** Relative approximate error for the arrival rates estimated by the PIA.



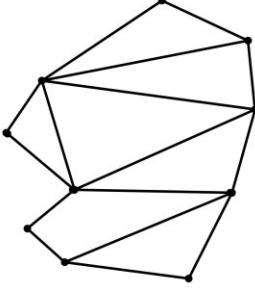
**Figure 5.5:** Reward loss vs offered traffic for the basic CAC scheme and the PIA [RB16b].

the exact PIA is not implementable for this link, the results suggest that the approximate variant is a good at approximating the exact long-term reward gains. Thus, with the MDP rule, the PIA detects and rejects adverse connections that would diminish the reward earned by the link. Furthermore, the decisions taken by the PIA adapt to the traffic load, which explains the superiority of the method, especially at high traffic loads. In this case, the PIA is more efficient than the basic CAC scheme at detecting and discarding harmful connection requests.

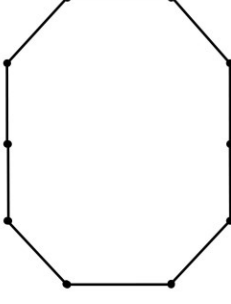
Based on the results presented in this section, we can argue that the macrostate-dependent PIA is a plausible approximation to solving the reward-based online resource allocation problem. In the following section, we evaluate the performance of the method for full-mesh, partial-mesh and ring topologies, where three aspects are investigated. First, a comparison is made for the reward losses attainable by the four MDP-based admission decision rules. Secondly, an analysis is made on the influence of the spectrum allocation algorithm on the PIA performance. And last, we present an analysis on GoS control. These three aspects are investigated by setting up simulations under different traffic load conditions.



Parameter	Value
$C_l$	32
$J$	90
$\lambda_1 = \dots = \lambda_{90}$	variable
$\mu_1 = \dots = \mu_{45}$	1.0
$\mu_{46} = \dots = \mu_{90}$	0.1
$b_1 = \dots = b_{45}$	2.0
$b_{46} = \dots = b_{90}$	8.0
$r_1 = \dots = r_{90}$	1.0
Initial policy $\Pi_0$	KSP
Spectrum allocation	first-fit



**Partial-mesh network**



**Ring network**

**Figure 5.6:** Network topologies and definition of connection class parameters [RB16d].

### 5.3. Performance Analysis of the Approximate PIA

To assess the performance of the approximate PIA for multi-link networks, simulations were set up for three network topologies: the partial-mesh and the ring topologies shown in Fig. 5.6, and the full-mesh version of these networks. The topologies consist of flex-grid optical links with capacities of  $C_l = 32$  slots, and have 10 nodes which are assumed to have sufficient transponders to serve the traffic demand. In these networks it is assumed that each of the 45 end-to-end node-pairs serves two connection classes, namely, a narrowband class with  $b_j = 2$  and a wideband class with  $b_j = 8$ . Thus, the three networks serve  $J = 90$  classes, where classes 1-45 and 46-90 comprise narrowband and wideband connections, respectively. The settings of the parameters  $\lambda_j$ ,  $\mu_j$ ,  $b_j$  and  $r_j$  are shown in Fig. 5.6.

As initial online policy  $\Pi_0$  for the PIA, we implemented the KSP (i.e. the  $k$ -shortest paths) algorithm proposed in [WWH<sup>+</sup>11]. For any connection request, the KSP assigns a lightpath on the first available shortest path (in  $\Gamma_j$ ) that fulfils the contiguity and continuity constraints. Furthermore, it is assumed, that for each connection class, the set of candidate paths  $\Gamma_j$  comprises the 2-shortest paths (w.r.t. the number of links) connecting the nodes  $(o, d)_j$ . Both the KSP and the PIA (in the PIR) use first-fit as spectrum allocation algorithm.

In the VDO the PIA executes the six steps outlined in Section 4.4.2 – Chapter 4. However, these steps are performed by defining the linear systems (for each link) in a simplified macrostate-space  $\Omega_n^l$  rather than in  $\Omega_n^l$ . Observe that in  $\Omega_n^l$  each link  $l$  would be described by macrostates  $\mathbf{n}^l = (n_1^l, \dots, n_{90}^l)$  with 90 components. Since for classes 1-45,  $b_j = 2$  and  $\mu_j = 1$ , and for classes 46-90,  $b_j = 8$  and  $\mu_j = 0.1$  (see Fig. 5.6), the cardinality of  $\Omega_n^l$  is reduced by the simplified link model from Section 4.2.3 - Chapter 4. With that model, classes 1-45 and 46-90 are aggregated into two classes  $i, k$  with  $\mu_i = 1$ ,  $b_i = 2$  and  $\mu_k = 0.1$ ,  $b_k = 8$ , respectively. On each link this reduces the number of classes from  $J = 90$  to  $J' = 2$ , so that every link  $l$  is described by macrostates  $\hat{\mathbf{n}}^l = (\hat{n}_i^l, \hat{n}_k^l)$  with two components. These macrostates define the simplified space  $\Omega_n^l$ . Thus, for each link  $l$ , the VDO calculates the link reward parameters from Equation (4.30) – Chapter 4, as follows:

$$r_i^l = \frac{\sum_{j=1}^{45} (r_j^l \cdot A c_j^l)}{\sum_{j=1}^{45} A c_j^l} \quad (5.5)$$

$$r_k^l = \frac{\sum_{j=46}^{90} (r_j^l \cdot A c_j^l)}{\sum_{j=46}^{90} A c_j^l} \quad (5.6)$$

To evaluate the performance of the approximate PIA, simulations are run to estimate the mean network reward rate  $R(\Pi)$  and the mean the network reward loss  $R_L(\Pi)$ . These performance metrics are evaluated for different offered traffic loads  $A$  which are calculated as:

$$A = \sum_{j=1}^{90} A_j \quad (5.7)$$

with  $A_j = \lambda_j / \mu_j$  (Erlangs). The traffic loads are generated from Equation (5.7) by varying the arrival rates  $\lambda_j$ , which are set equal for all  $j$ . Hence, if  $\lambda_j = \lambda, \forall j$ , we have:

$$A = \lambda \cdot \sum_{j=1}^{90} \mu_j^{-1} \quad (5.8)$$

where the termination rates  $\mu_j$  are fixed and equal to the values defined in Fig. 5.6. On the other hand, note that the reward parameters are set as  $r_j = 1, \forall j$ , and thus, the PIA maximizes (for all considered traffic loads) the overall mean rate of accepted connections, i.e. the objective function defined in Equation (3.6) - Chapter 3.

In the following we evaluate the performance of the approximate PIA for the four admission decision rules defined in Chapter 4. Recall that the MDP rule arises as a consequence of the link independence assumption, and hence, it must calculate the optimum policy if and only if there are no correlations among network links. The remaining three strategies aim at minimizing the lack of accuracy of the MDP rule. Based on this, in Section 5.3.1 we compare the performance of the MDP rule with that obtained when the networks apply the KSP scheme only. Then in Section 5.3.2 we focus our attention on the performance attainable by each of the four admission decision rules. The purpose is to evaluate the accuracy of the link independence assumption, and to determine which rule counteracts the correlations among links at its best. For each rule, in Section 5.3.3 we study the impact that the spectrum allocation algorithm used by the PIA has on the network performance. Finally, in Section 5.3.4 simulation results are presented that show how the PIA can implement grade of service control and equalization.

### 5.3.1 Resource Allocation with the KSP Scheme and the MDP Rule

It would be desirable to compare the performance of the approximate PIA with the optimum solution obtained by the exact algorithm formulated in Chapter 3. However, since the optimum solution cannot be calculated, the performance of the approximate approach is compared against that of the KSP scheme. Two reasons motivate this comparison. First, the results in [WWH<sup>+</sup>11] show that, compared to existing RSA algorithms [TAK<sup>+</sup>14], the KSP has an outstanding performance. Secondly, the KSP is very simple to implement, which makes it suitable as starting policy  $\Pi_0$  for the PIA.

In Fig. 5.7 we show performance evaluation results for the three network topologies when resources are allocated by the KSP scheme only and by the PIA applying the MDP rule (using the KSP as starting policy  $\Pi_0$ ). The graphs in Fig. 5.7 depict the network reward rates  $R(\Pi)$  and the network reward losses  $R_L(\Pi)$  as function of the total offered traffic. The results correspond to the policy calculated in the sixth iteration cycle, i.e. the cycle that meets Equation (5.1). The reward losses are expressed as a percentage (%) calculated with Equation (5.2). From the results, it can be seen that regardless of the topology, the MDP rule yields a lower reward loss compared to the KSP scheme, especially for high traffic loads. As with the single-link networks studied in Section 5.2, it is observed that admitting all connection requests that fulfil the contiguity and continuity constraints (as the KSP does), renders the network indifferent to the harmful effects that the connections can bring. In contrast, with the MDP rule, the long-term reward gains allow for detection and rejection of adverse connections, i.e. requests that bring to the network a negative reward gain. Since  $r_j = 1, \forall j$ , a connection with negative gain would decrease, if admitted, the overall mean rate of accepted connections, or equivalently, it would increase the overall blocking probability. Observe that these results are also verified by the reward rates shown in Fig. 5.7. In the simulated traffic load regions, the network reward rate is higher for the MDP rule.

A remark regarding the optimality of the PIA - with MDP rule - should be made. The presence of multi-link shortest paths in  $\Gamma_j$  causes non-zero correlations between the network links. As this effect is neglected by the MDP rule, the performance of the PIA shown in Fig 5.7 corresponds to a policy  $\Pi$  that is sub-optimal. The partial-mesh and the ring networks in Fig. 5.6 have average shortest path lengths of 1.93 and 2.77, respectively. In these topologies, correlation do exist. In spite of this, the MDP rule outperforms the KSP scheme in the simulated traffic load regions. In the case of the full-mesh network, the superiority of the PIA scheme is obvious. Here, the connections are mainly routed on the direct links between node pairs. Thus, for the full-mesh network, the inter-link correlations are negligible.

### 5.3.2 Resource Allocation with the MDP-based Admission Rules

Although the MDP rule outperforms the KSP algorithm, it is important to test its accuracy. For this, we compare its performance with that attained by the three MDP-based rules which aim at counteracting the effect of link correlations. The comparison is made by performing simulations for the ring and the partial-mesh networks in Fig. 5.6 (i.e. for the topologies which cause correlations among the network links). In Fig. 5.8 we present the reward losses (depending on the traffic load) that these four MDP-based strategies yield. As seen, regardless of the network topology, the MDP-PGMC rule shows the best performance (especially at very low traffic loads), whereas the reward losses induced by the MDP-SP and the MDP-PG rules are similar. The worst performance is attained by applying the MDP rule. This result evinces that, although better than the KSP scheme, this rule calculates a sub-optimal policy  $\Pi$  owing to its lack of accuracy.

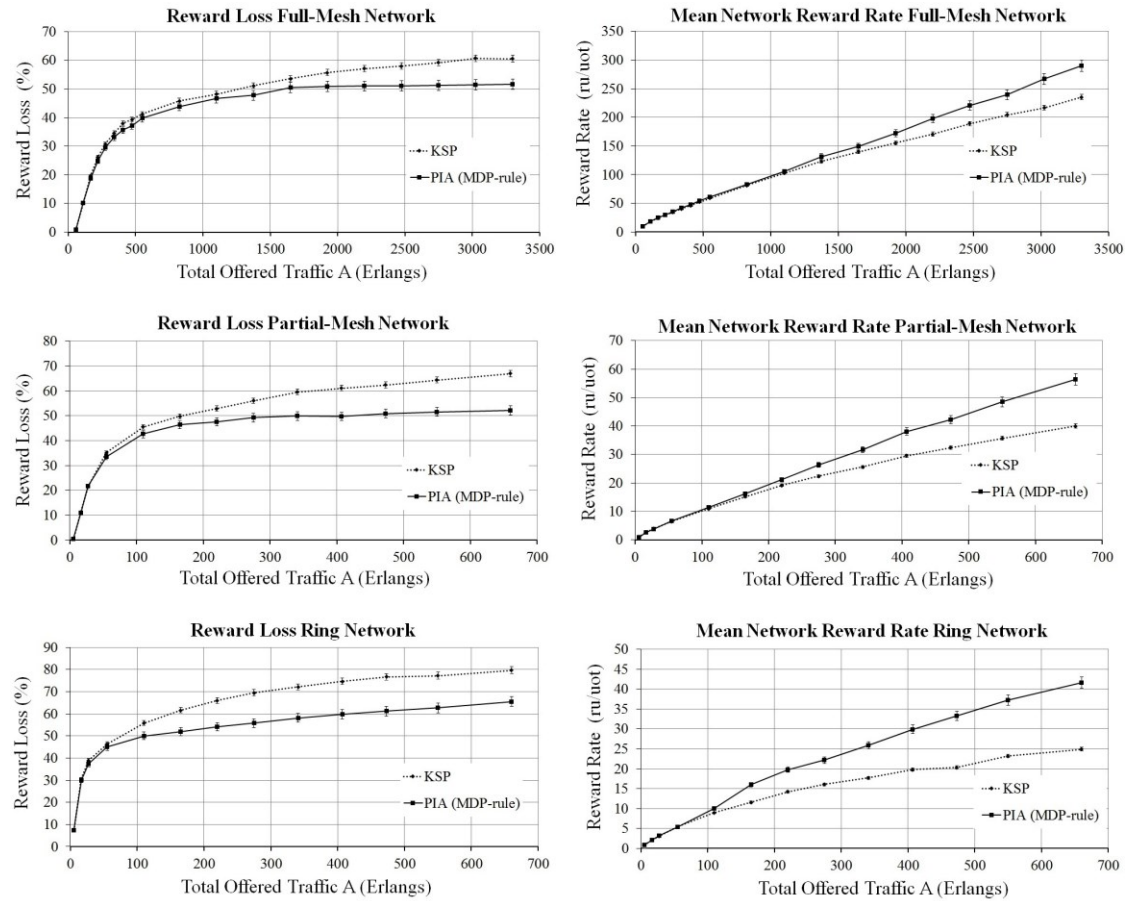


Figure 5.7: Simulation results for the KSP scheme and the PIA with the MDP rule [RB16d].

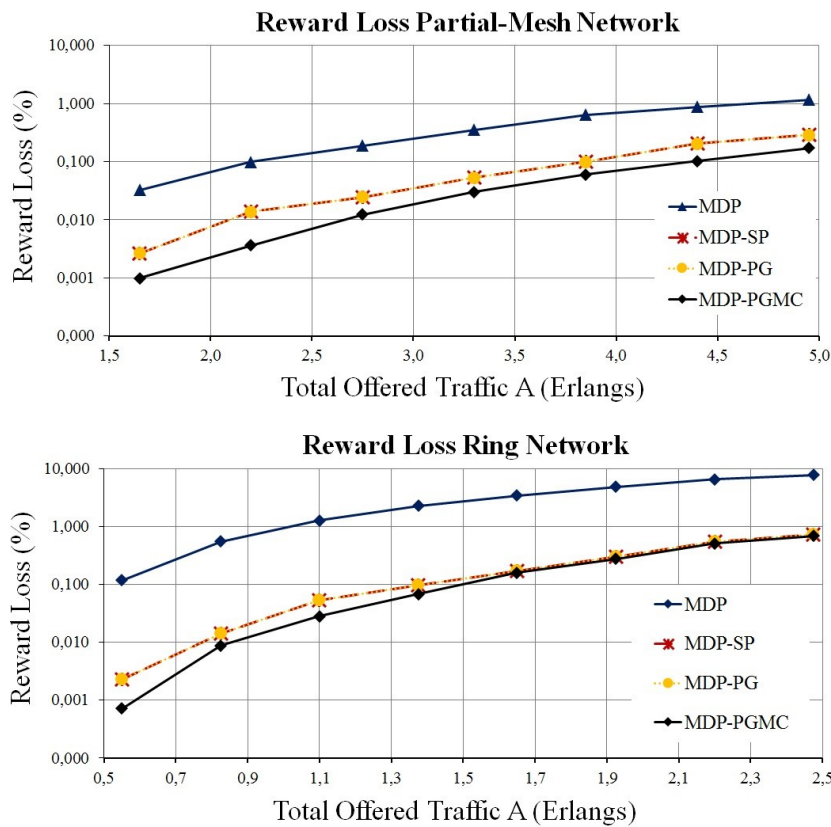
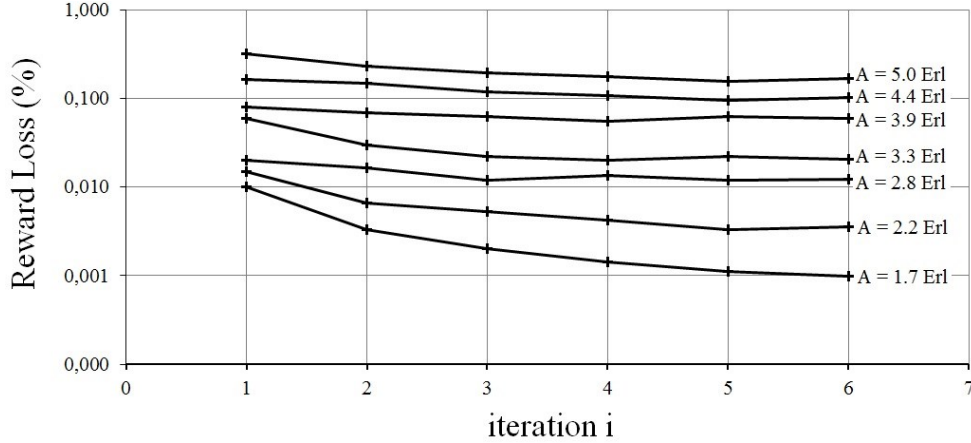


Figure 5.8: Performance comparison of the four PIA admission decision rules [RB17a].

A (Erl)	MDP		MDP-SP		MDP-PG		MDP-PGMC	
	1 <sup>st</sup> (%)	2 <sup>nd</sup> (%)	1 <sup>st</sup> (%)	2 <sup>nd</sup> (%)	1 <sup>st</sup> (%)	2 <sup>nd</sup> (%)	1 <sup>st</sup> (%)	2 <sup>nd</sup> (%)
1.7	13.5	86.5	99.9	0.1	99.9	0.1	82.9	17.1
2.2	15.2	84.8	99.8	0.2	99.8	0.2	80.5	19.5
2.8	15.9	84.1	99.6	0.4	99.6	0.4	79.1	20.9
3.3	16.1	83.9	99.2	0.8	99.2	0.8	77.8	22.2
3.9	16.6	83.4	98.8	1.2	98.8	1.2	77.0	23.0
4.4	18.5	81.5	98.3	1.7	98.3	1.7	76.4	23.6
5.0	18.7	81.3	97.7	2.3	97.7	2.3	75.8	24.2

**Table 5.13:** Percentage of connections admitted on each candidate path [RB17a].

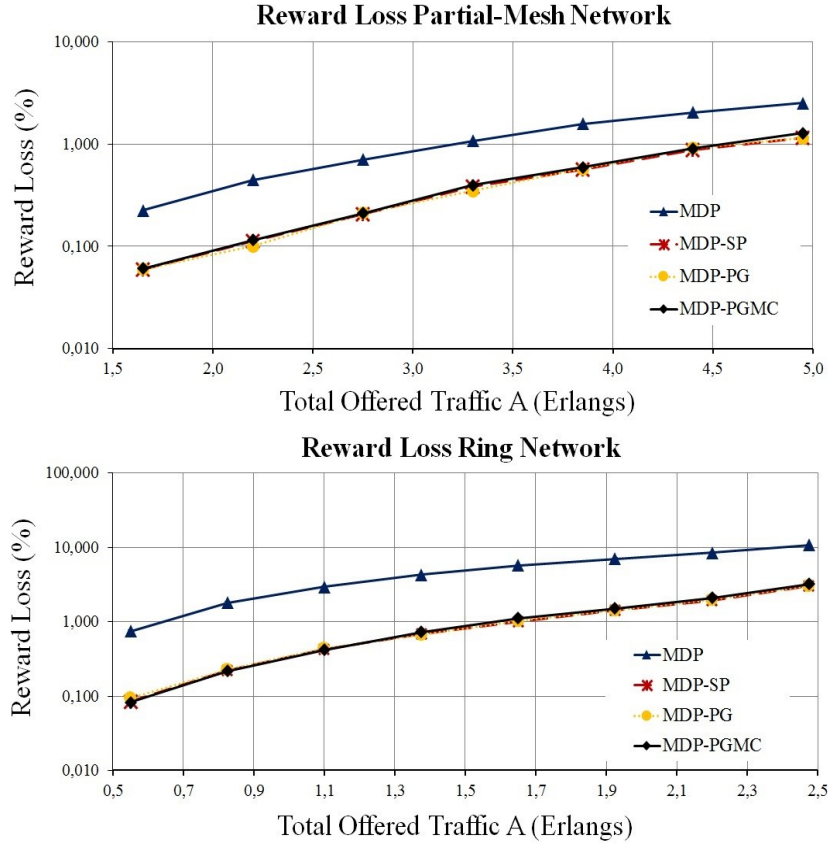


**Figure 5.9:** PIA convergence for the partial-mesh network applying the MDP-PGMC rule.

The differences among the four rules arise because in the simulated traffic load regions the links have low utilizations. As a consequence, the macrostate-dependent link reward-gains are mainly positive. (This statement can be understood by recalling the small single-link network studied in Section 5.2.1. In Tables 5.4-5.9 it is seen that in low traffic loads, the gains are positive and tend to equal the reward parameters.) Thus, in most cases, when a connection arrives, the shortest path in  $\Gamma_j$  has a positive reward gain and exhibits a suitable lightpath (owing to the low utilization). This explains why the MDP-SP and the MDP-PG rules have comparable performance. These two strategies tend to make the same decisions, i.e. placing the connections on the lightpath calculated in the shortest path in  $\Gamma_j$ . On the other hand, since the link gains are mainly positive, there are macrostates for which the second shortest path may yield a higher gain than the shortest path. This renders the MDP rule (in low traffic regions) prone to admit connections on the longest route, thereby increasing the reward losses. The MDP-PGMC rule proves to be the best strategy as it attempts to evenly load the network without impairing the reward rate. To illustrate these arguments better, Table 5.13 shows for the partial-mesh network results in Fig. 5.8, the percentage of connections admitted by each rule on the 1<sup>st</sup> and the 2<sup>nd</sup> shortest path in  $\Gamma_j$ . As can be seen, the MDP rule tends to load the 2<sup>nd</sup> path, whereas the MDP-SP and the MDP-PG strategies place most of the traffic on the shortest path. The MDP-PGMC rule alleviates the load on the 1<sup>st</sup> path by using the 2<sup>nd</sup> route only when it provides higher capacity while guaranteeing a positive reward increment.

For the considered network topologies and the simulated traffic load regions, the higher losses attained by the MDP rule indicate the inaccuracy of the link independence assumption. Although the connections are placed on the path with the highest reward gain, the MDP rule disregards any harmful effect that the connections have on links which are not in the path set  $\Gamma_j$ . However, the results in Fig. 5.8 show that this disadvantage can easily be overcome by adding simple modifications to the MDP rule which reduce the link state correlations. That is the case of the three remaining decision rules which significantly improve the network performance.

For all admission decision rules, the PIA runs the policy  $\Pi_0$  during the first iteration cycle. To show the convergence of the PIA, in Fig. 5.9, for the partial-mesh network, the reward loss attained by the MDP-PGMC rule is depicted as a function of the iteration cycle  $i$ . Each curve in Fig. 5.9 represents the reward loss for a specific traffic load  $A$  (remark: these loads correspond to the values depicted in the first column of Table 5.13). The performance of the KSP (i.e. policy  $\Pi_0$ ) corresponds to the reward losses obtained for



**Figure 5.10:** Performance of the PIA using RF spectrum allocation scheme.

$i = 1$ . As can be seen in Fig. 5.9, the speed of convergence is dependent on  $A$ . In particular, for the partial-mesh and the ring network, it was observed that after  $i = 6$  iterations the policies  $\Pi_i$  calculated by the PIA met the stopping criterion in Equation (5.1). The reward losses shown in Fig. 5.8 correspond to the results obtained for  $i = 6$ . Figure 5.9 also indicates how the policies  $\Pi_i$  calculated by the PIA outperform the KSP policy  $\Pi_0$ .

Although compared to the KSP and the MDP rule, the performance of the MDP-SP, the MDP-PG and the MDP-PGMC is better, it cannot be determined how far or close the policies calculated by these three approaches are from the optimum  $\Pi^*$ . The reason is that the optimum solution provided by the exact PIA cannot be calculated. However, the results indicate that the rules observe a good performance in the considered traffic load regions.

### 5.3.3 Impact of the Spectrum Allocation Method on the PIA

As discussed in Chapter 4, the macrostate-dependent model assigns the same long-term reward gain to all lightpaths routed on the same path. Thus, the approximate PIA only calculates one candidate lightpath on each route that fulfils the contiguity and continuity constraints. However, as argued in Chapters 3 and 4, in reality, on the same path lightpaths may have different gains. (This fact has been observed for the small single-link network studied in Section 5.2.1 - see Tables 5.4-5.9.) For this reason, the spectrum allocation scheme used by the PIA is expected to have an influence on the network performance. To investigate this, we run simulations to determine the reward losses attained by the PIA when random-fit (RF) is used - in the PIR step - as spectrum allocation algorithm. Based on a predetermined probability density function, RF randomly selects, for each route that fulfils the contiguity and continuity constraints, a lightpath from the set of candidate lightpaths available in the route [TAK<sup>+</sup>14]. By using RF with a uniform distribution, in Fig. 5.10 we present the reward losses attained by the four PIA admission decision rules in the partial-mesh and the ring networks. The simulated traffic loads correspond to those evaluated in Fig. 5.8. (Recall that in Fig. 5.8, the PIA uses FF as spectrum allocation algorithm.)

As with FF, the results show that for RF the MDP rule exhibits the worst performance. However, it can also be seen that RF deteriorates the performance of the MDP-PGMC rule, as it is now comparable to the MDP-SP and the MDP-PG strategies. In Table 5.14 and Table 5.15 we present a comparison of the reward losses - expressed in (%) - obtained by each rule in the partial-mesh and the ring networks, respectively. For a given rule, the comparison is made between the reward losses obtained by FF and RF. For that, in both tables we present the percentage change  $\Delta R_L$  of the reward loss that is caused by RF with

A (Erl)	MDP			MDP-SP			MDP-PG			MDP-PGMC		
	FF (%)	RF (%)	$\Delta R_L$ (%)	FF (%)	RF (%)	$\Delta R_L$ (%)	FF (%)	RF (%)	$\Delta R_L$ (%)	FF (%)	RF (%)	$\Delta R_L$ (%)
1.7	0.033	0.225	582	0.003	0.059	1867	0.003	0.060	1900	0.001	0.061	6000
2.2	0.099	0.447	352	0.014	0.112	700	0.014	0.100	614	0.004	0.115	2775
2.8	0.190	0.704	271	0.025	0.206	724	0.025	0.211	744	0.012	0.183	1425
3.3	0.354	1.077	204	0.053	0.383	623	0.053	0.350	560	0.018	0.397	2106
3.9	0.641	1.580	146	0.099	0.559	465	0.099	0.570	476	0.060	0.592	887
4.4	0.885	2.041	131	0.203	0.876	332	0.203	0.908	347	0.102	0.906	788
5.0	1.163	2.530	118	0.287	1.155	302	0.287	1.147	300	0.172	1.277	642

**Table 5.14:** Reward losses obtained by FF and RF algorithms in the partial-mesh network.

A (Erl)	MDP			MDP-SP			MDP-PG			MDP-PGMC		
	FF (%)	RF (%)	$\Delta R_L$ (%)	FF (%)	RF (%)	$\Delta R_L$ (%)	FF (%)	RF (%)	$\Delta R_L$ (%)	FF (%)	RF (%)	$\Delta R_L$ (%)
0.6	0.119	0.747	528	0.002	0.084	4100	0.002	0.093	4550	0.001	0.082	8100
0.8	0.549	1.797	227	0.014	0.222	1486	0.014	0.228	1529	0.009	0.217	2311
1.1	1.271	2.933	131	0.053	0.430	711	0.053	0.436	723	0.028	0.418	1393
1.4	2.287	4.274	87	0.097	0.691	612	0.097	0.669	590	0.068	0.726	968
1.7	3.431	5.707	66	0.170	1.002	489	0.170	1.023	502	0.159	1.111	599
1.9	4.809	6.999	46	0.300	1.441	380	0.300	1.417	372	0.277	1.502	442
2.2	6.481	8.432	30	0.540	1.946	260	0.545	1.975	262	0.510	2.082	308

**Table 5.15:** Reward losses obtained by FF and RF algorithms in the ring network.

respect to the losses attainable by FF. For all traffic loads and network topologies, the results show that the decision rules perform better with FF. When RF is used, the performance is severely deteriorated. The harmful effect is more noticeable under low traffic loads (note that the lower the offered traffic the higher the reward loss change  $\Delta R_L$ ). This is explained by the fact that in low traffic regime, the network paths observe low utilizations, thereby more candidate lightpaths are available for incoming connection requests. In this scenario, with FF the PIA admits connections on the lightpath with the lowest indices in the grid. This avoids transitions to states that may block future arrivals. On the contrary, with RF, the PIA is prone to place connections on lightpaths that block future requests. For higher traffic loads,  $\Delta R_L$  decreases as a higher resource occupation implies that both FF and RF have a reduced set of candidate lightpaths.

Although the exact gains cannot be calculated for each candidate lightpath, the results in Tables 5.14-5.15 allow us to infer that the lightpaths selected by FF yield higher long-term reward gains than those selected by RF. When RF is used, the PIA is more likely to select lightpaths which configure blocking states for class- $j$  connections. Those lightpaths have, in reality, lower long-term reward gains (as it was the case for the single-link network studied in Section 5.2, where it was shown that the approximate PIA tends to overestimate the gain of spectrum configurations that block class- $j$  traffic). Thus, with respect to RF, the performance of the PIA can be improved by using FF as the spectrum allocation method.

### 5.3.4 Grade of Service Control

Regardless of the admission decision rule, the PIA can control the class-specific blocking probabilities  $B_j$  by appropriately setting the parameters  $r_j$ , thereby the policies  $\Pi_i$  calculated by the PIA may achieve a desired GoS for each connection class. This property allows GoS equalization and the prioritization of selected classes. To illustrate this, consider the partial-mesh network in Fig. 5.6 and the PIA using the MDP and the MDP-PGMC rules with FF as spectrum allocation scheme. For each rule, the dependence of the blocking probabilities  $B_j$  on the parameters  $r_j$  is studied in low and high traffic load conditions. This is accomplished by defining the four simulation scenarios depicted in Table 5.16. As seen, the MDP rule is evaluated for  $A = 9.9$  and  $A = 99$ , whereas the MDP-PGMC rule for  $A = 5.0$  and  $A = 8.3$ . In all scenarios, we let all narrowband and wideband connections to have, respectively, the reward parameters  $r_n = r_1 = \dots = r_{45}$  and  $r_w = r_{46} = \dots = r_{90}$ . Furthermore, for all wideband connections (i.e. classes 46-

Parameter	MDP		MDP-PGMC	
	$A = 9.9$	$A = 99$	$A = 5.0$	$A = 8.3$
$C_l$	32 (slots)	32 (slots)	32 (slots)	32 (slots)
$J$	90 (classes)	90 (classes)	90 (classes)	90 (classes)
$\lambda_1 = \dots = \lambda_{90}$	0.11	1.1	0.010	0.016
$\mu_1 = \dots = \mu_{45}$	1.00	1.0	1.0	1.0
$\mu_{46} = \dots = \mu_{90}$	1.00	1.0	0.1	0.1
$b_1 = \dots = b_{45}$	2 (slots)	2 (slots)	2 (slots)	2 (slots)
$b_{46} = \dots = b_{90}$	8 (slots)	8 (slots)	8 (slots)	8 (slots)
$r_n = r_1 = \dots = r_{45}$	variable	variable	variable	variable
$r_w = r_{46} = \dots = r_{90}$	8 (ru)	8 (ru)	1 (ru)	1 (ru)
Policy $\Pi_0$	KSP	KSP	KSP	KSP
Spectrum allocation algorithm	first-fit	first-fit	first-fit	first-fit

**Table 5.16:** Definition of parameters for GoS control simulations [RB16d, RB17a].

90), the reward parameters  $r_w$  are fixed and equal to the values defined in Table 5.16. Then in every scenario, simulations are run to assess the GoS of each connection class for different values of  $r_n$ .

Figure 5.11 and Fig. 5.12 show the simulation results for the considered scenarios. The graphs in both figures depict the blocking probability  $B_n$  (averaged over classes 1-45) of narrowband connections, the blocking probability  $B_w$  (averaged over classes 46-90) of wideband connections, and the overall blocking probability  $B_T$  as function of the narrowband reward parameters  $r_n$ . (Remark: in the graphs, for each  $r_n$  the blocking probabilities are those attained by the PIA in the sixth iteration cycle.) The results illustrate that the PIA can control the GoS by varying the reward parameters. However, it should be noted that changing any individual reward parameter influences the blocking probabilities of all connection classes. In fact, the graphs show that an improper choice of  $r_n$  may deteriorate the GoS of wideband connections. For both rules, in low and high traffic load conditions, low values of  $r_n$  cause narrowband connections to be rejected (as they bring to the network a negligible reward) and priority is only given to wideband traffic. Thus, the blocking  $B_n$  is higher than  $B_w$ . In this case,  $B_w$  is mainly influenced by the traffic offered by wideband connections and by the capacity of the network links. When the parameters  $r_n$  are increased, narrowband connections become more valuable. As a result,  $B_n$  decreases to the detriment of  $B_w$ , which shows the dependence of the class-specific blocking  $B_j$  on the reward parameters of all connection classes.

In addition to the capability of prioritizing traffic classes, the results also indicate that the network may set the reward parameters to equalize  $B_n$ ,  $B_w$  and  $B_T$ . Consider as an example the MDP-PGMC rule. By using the parameters defined in Table 5.16 with  $r_n = r_w = 1$ , we have that  $r_j = 1, \forall j$ , and thus the PIA minimizes the overall blocking probability  $B_T$ . In Table 5.17 we show (with 95% confidence intervals) this blocking as well as  $B_n$  and  $B_w$  for  $A = 5.0$  and  $A = 8.3$ . As seen, the minimization of  $B_T$  does not imply that all connection classes attain the same blocking. For instance, when  $A = 5$ , the minimum blocking is  $B_T = 0.19 \pm 0.03$ . Although the overall losses are minimized, each class suffers a distinct GoS, i.e.  $B_n = 0.06 \pm 0.01$ , whereas  $B_w = 0.31 \pm 0.02$ . However, the results in Fig. 5.12 show that there exist an  $r_n$  for which  $B_n$ ,  $B_w$  and  $B_T$  are equalized. The values of  $r_n$  (obtained via simulations) that equalize these blocking probabilities are presented in Table 5.17. As an example, for  $A = 5$ , the GoS is equalized to  $B_n \cong B_w \cong B_T \cong 0.33$  if the network operator sets  $r_n = 0.0148$  (ru) and  $r_w = 1$  (ru). In general, the values of the parameters  $r_n$  and  $r_w$  for which a GoS equalization is obtained are dependent on  $A$ . If, for example, the traffic load increases to  $A = 8.3$ , equalization is attained by setting  $r_n = 0.0183$  (ru) - as can be seen from Table 5.17 and Fig. 5.12.

Although the analytical relationship between the parameters  $r_j$  and  $B_j$  is not trivial and depends on the traffic load, the results suggest that online and adaptive GoS control algorithms can be investigated so as to achieve (at its best) a desired grade of service for each connection class. Contrary to the exact and the approximate PIA, which have full knowledge of the reward parameters  $r_j$ , any reward-based GoS control scheme must be able to calculate (online and for the current traffic load conditions) the values of  $r_j$  that either cause the lowest possible blocking  $B_j$  or cause GoS equalization. This issue is an open challenge which has also been posed in the literature of reward-based approaches for telephone and packet-switched networks [DM89, DM92, DM94, Dzi97, Nor02]. To the best of our knowledge, the work in [Dzi97] has been the only one which has approached the problem of fairness and equalization (for reward-based schemes) from a game theoretical perspective. Still, further research is required so as to figure out whether those ideas can be applied to the exact and the approximate reward models for dynamic optical networks.

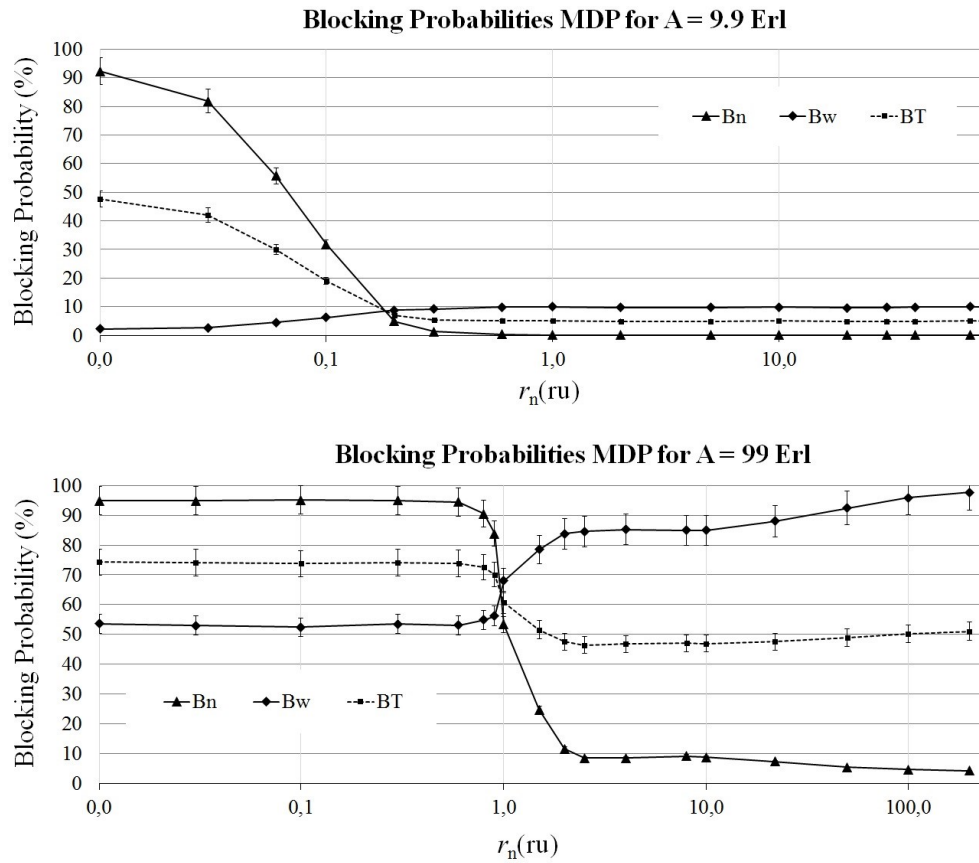


Figure 5.11: Blocking probabilities vs reward parameters MDP rule [RB16d].

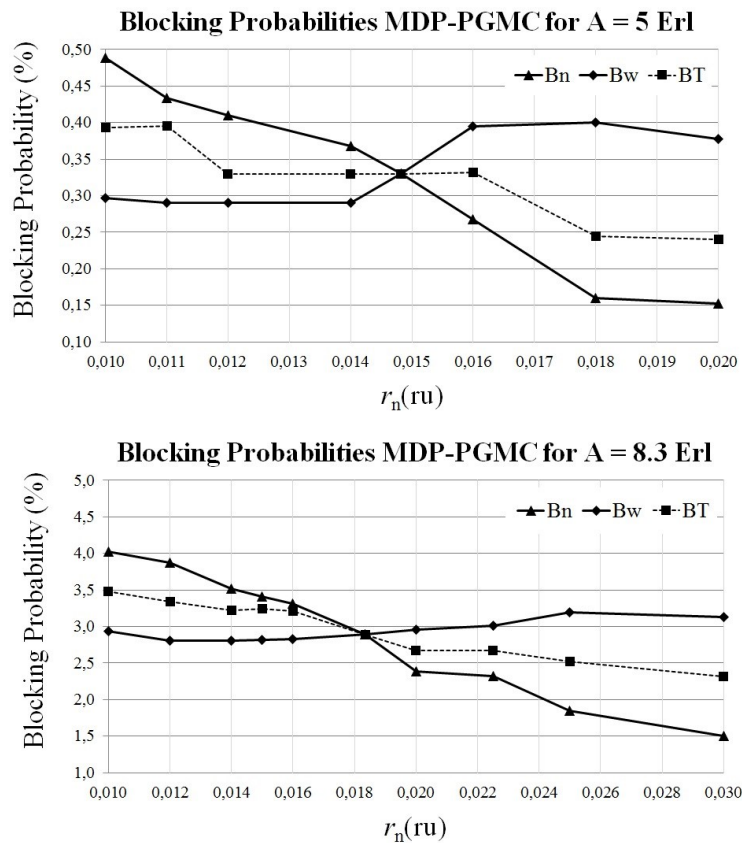


Figure 5.12: Blocking probabilities vs reward parameters MDP-PGMC rule [RB17a].



$A$ (Erl)	$r_n$ (ru)	$r_w$ (ru)	$B_n$ (%)	$B_w$ (%)	$B_T$ (%)
5.0	1	1	0.06±0.01	0.31±0.02	0.19±0.03
	0.0148	1	0.32±0.04	0.33±0.04	0.33±0.05
8.3	1	1	0.56±0.03	2.96±0.05	1.77±0.04
	0.0183	1	2.91±0.04	2.89±0.03	2.90±0.04

**Table 5.17:** Grade of service equalization partial-mesh network MDP-PGMC rule [RB17a].

## 5.4. Chapter Summary

A relevant goal of a reward based approach to online resource allocation is to provide knowledge about the harmful effect that connections may cause in the network. This effect depends on the network state and on the traffic load, and is quantified by the long-term reward gains. The knowledge of these gains is important as it allows the implementation of admission control mechanisms that maximize the reward earned by the network. Connection admission control is applied on the time scale of connection interarrival times, which implies that CAC algorithms must be fast at estimating those gains. However, for most networks, an exact estimation is unfeasible owing to the huge cardinality of the network state-space. To circumvent this problem, the approximate PIA resorts to two strategies: first, a macrostate-dependent approximation of the long-term reward gains; and secondly, the link independence assumption. For the considered performance evaluation scenarios, the results show that the macrostate-dependent long-term reward gains are a reasonable approximation to the exact (state-dependent) gains. The accuracy of the approximation stems from the online procedure that estimates the state-dependent link arrival rates. Moreover, the simulation results for the partial-mesh and the ring networks show that although the link independence assumption is not always accurate, its performance can substantially be improved by adding simple modifications to it. That is the case of the MDP-SP, the MDP-PG and the MDP-PGMC rules. The performance of these rules depends on the spectrum allocation algorithm used by the approximate PIA. The reason for this is explained by the fact that the approximate PIA assumes that the lightpaths routed on the same path have similar gains which - as verified by the results - is not always true. However, with respect to random-fit, the inaccuracy of this assumption can easily be improved by using first-fit as the spectrum allocation method. With this algorithm, the MDP rules place connections on lightpaths that lead to less blocking in the network. From the results we conclude that an exact description of the optical grid configuration is not always necessary to design adaptive and state-dependent resource allocation algorithms. Instead, the stochastic properties of the network can be modelled by the approximate PIA. By this method, the computational complexity of the exact state-dependent PIA is skipped, thereby rendering the online resource allocation problem solvable.

By comparing the performance of the PIA with the KSP scheme (which only rejects connections that violate the contiguity and capacity constraints), it is observed that the PIA yields a lower reward loss. This evinces the advantage of detecting and rejecting adverse connection requests that, despite the availability of resources, would prevent the network from admitting more valuable connections. Furthermore, the PIA has two substantial advantages compared to existing online RSA schemes. First, different reward objectives can be attained (e.g. maximization of carried traffic, revenue, etc.) by properly defining the reward parameters. Secondly, as shown by the results, GoS control can be applied on individual connection classes. However, further research is required to investigate how a reward-based algorithm can be implemented for adaptive GoS control, i.e. for tuning the connection reward parameters to either provide a desired GoS or to provide equalization of the class-specific blocking probabilities.

In the next chapter, we further investigate how an existing connection establishment protocol can be adapted to implement the approximate PIA. Besides the implementation details, an analytical model is proposed to assess the connection set up latency when the PIA is used as the resource allocation method.



# Chapter 6

## Connection Establishment and Online Resource Allocation

To serve connections on demand, dynamic optical networks rely on signalling protocols for connection establishment, maintenance and teardown. In particular, connection establishment protocols define the signalling procedures whereby resources are allocated. This implies that the design of resource allocation algorithms needs to conform to the specifications of the connection setup protocol applied by the network. In this chapter we tackle this issue by studying an implementation of the PIA that uses a 3-way handshake protocol for connection setup. Moreover, we propose an analytical approach to evaluating the connection setup latency. In Section 6.1, the implementation scenario for the PIA is explained, where it is discussed how the protocol is used to perform RSA and CAC. To evaluate the connection setup latency, in Section 6.2 we outline a method to assess the performance of communication protocols by reducible task graphs. Then in Section 6.3 the method is used to formulate an analytical performance evaluation model for the 3-way handshake protocol. To derive the model, the protocol is described by a task graph that represents the signalling latency of the connection setup procedure. By using reduction techniques, the graph is simplified to obtain a performance model that estimates the mean connection setup time. In Section 6.4 the numerical results obtained by the analytical model are compared with simulations of selected scenarios. This chapter includes results published by the author in [RB17a, RB17c, RB18a].

### 6.1. Implementation of the Approximate PIA

Different implementation variants can be proposed for the approximate PIA. They mainly differ in how the information handled by the algorithm is distributed and processed by the network elements. In this section we propose an implementation that employs the path computation element (PCE) [FVA06] to coordinate the PIA iteration cycles. The execution of the VDO is centralized by the PCE, whereas the PIR is executed by following a decentralized approach that relies on a connection establishment protocol.

#### 6.1.1 Centralized Execution of the VDO

In the proposed implementation scenario, three relevant tasks are performed by the PCE: the coordination of the PIA operation, the execution of the VDO and the update of the network state. The first two tasks are depicted by the time-sequence diagram in Fig. 6.1. To coordinate the PIA operation, every  $\Delta T$  time units the PCE commands each network node to start executing the  $i$ th iteration cycle. During that cycle, the nodes locally take online measurements to estimate the link arrival rates. At the end of the cycle, every node sends the measurements to the PCE. With this information the PCE centralizes the execution of the VDO as follows. First, for each link  $l$ , it solves the corresponding system of linear equations for  $R^l(\Pi_i)$  and the transient values  $v(\mathbf{n}^l, \Pi_i)$ . Secondly, from these values, it estimates the long-term reward gains  $g_j^l(\mathbf{n}^l, \Pi_i)$ . Furthermore, since the PCE has full visibility of the network state, it calculates the path sets  $\Gamma_j$  to be used by the PIR in the next cycle. After calculating  $g_j^l(\mathbf{n}^l, \Pi_i)$  and  $\Gamma_j$ , the PCE sends this information to the nodes and initializes the next iteration cycle. (Remark: The PCE only sends to a given node the gains  $g_j^l(\mathbf{n}^l, \Pi_i)$  of the links that connect to the node. Likewise, it sends to the node the sets  $\Gamma_j$  that have at least one candidate path containing the node. This prevents the nodes from receiving and storing information that is irrelevant for the execution of the PIR step.)

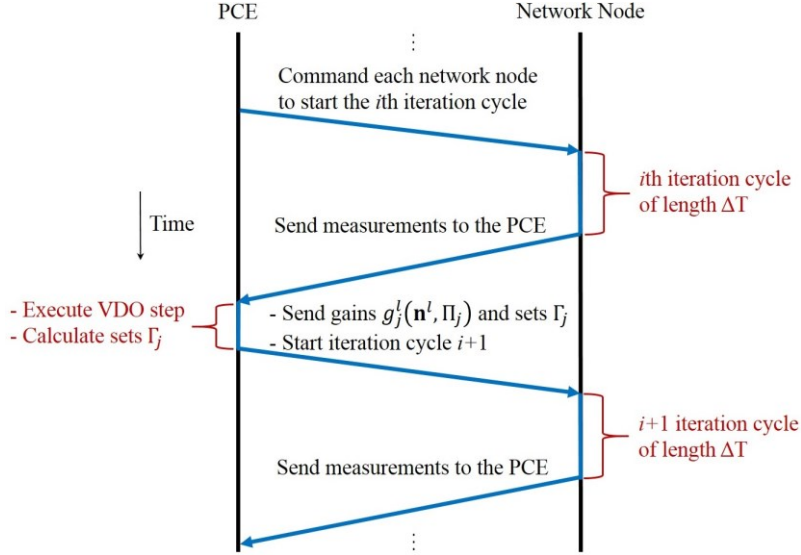


Figure 6.1: Time-sequence diagram centralized implementation of the VDO.

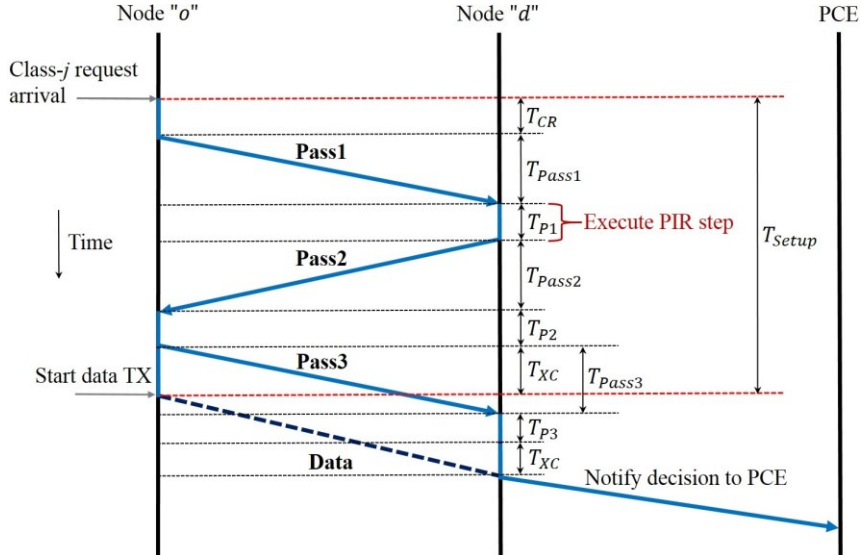


Figure 6.2: Time-sequence diagram implementation of the PIR with the 3WHS protocol.

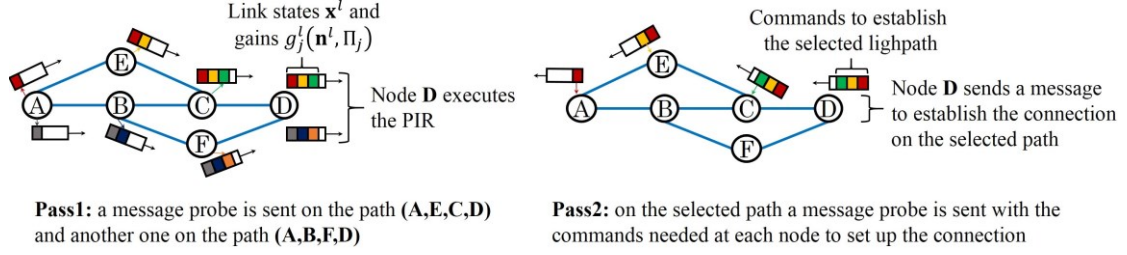
Variable	Description
$T_{CR}$	Processing time of a connection request at node "o"
$T_{Passj}$	Processing time of pass $j$ , $j = 1,2,3$
$T_{P1}$	Processing time at node "d" of the message probes sent by node "o" in Pass1
$T_{P2}$	Processing time at node "o" of the Pass2 signalling message sent by node "d"
$T_{P3}$	Processing time at the node "d" of the Pass3 signalling message
$T_{XC}$	Time to perform a cross-connection in an optical node

Table 6.1: Variables describing the 3WHS protocol.

### 6.1.2 Decentralized Execution of the PIR

To enable fast connection setup times in dynamic optical networks (e.g. in the range of milliseconds to seconds), a 3-way handshake signalling (3WHS) protocol was proposed in [SN09, CCC<sup>+</sup>12, SCG<sup>+</sup>12, SGK<sup>+</sup>14]. In Fig. 6.2 we propose an adaptation of that protocol that enables the execution of the PIR step during the connection establishment procedure. (Table 6.1 defines the variables depicted in Fig. 6.2.)

The PIR is implemented as a decentralized probing mechanism that consists of three passes. To explain them, assume that all network nodes have received from the PCE the corresponding gains  $g_j^l(\mathbf{n}^l, \Pi_i)$  and



**Figure 6.3:** Example of the course of a Pass1 and a Pass2 signalling message probes.

the path sets  $\Gamma_j$ . Let us further assume that a connection request of class- $j$  arrives at the node “ $o$ ” in  $(o, d)_j$  - see Fig 6.2. This node takes a time  $T_{CR}$  to process the request. After this, it starts a Pass1 by sending on each route in  $\Gamma_j$  a message probe towards node “ $d$ ”. The probes are sent out simultaneously by the node “ $o$ ”. As a message probe travels towards “ $d$ ”, it collects the states  $\mathbf{x}^l$  and the reward gains  $g_j^l(\mathbf{n}^l, \Pi_j)$  of the links in the path it traverses. (Remark:  $g_j^l(\mathbf{n}^l, \Pi_j)$  is the gain that the class- $j$  request would bring to link  $l$  if it is accepted in the macrostate  $\mathbf{n}^l$  defined by the current link state  $\mathbf{x}^l$ .) Thus, every message probe must be processed at all intermediate nodes in the path. This processing includes both reading the message and writing in it the link states and the reward gains needed to execute the PIR step. The processing time  $T_{Pass1}$  of Pass1 (see Fig. 6.2) is completed when all message probes arrive at “ $d$ ”. All probes need not arrive at the same time as each path may have different propagation delays and number of nodes.

With the information provided by the message probes, node “ $d$ ” takes a time  $T_{P1}$  to execute the PIR step. Therefore, after  $T_{P1}$  time units, node “ $d$ ” knows whether the class- $j$  request must be admitted or rejected. In case of admission, it also knows the lightpath on which the connection needs to be established. With this knowledge, node “ $d$ ” triggers the resource allocation procedure. For that, it starts a Pass2 by sending a signalling message towards “ $o$ ” on the path selected by the PIR. This message has the commands that the nodes in the path need to cross-connect the lightpath. When this message arrives at an intermediate node, it is read and immediately forwarded to the next upstream node while the cross-connection process is locally triggered. (For all nodes we denote the cross-connection time as  $T_{XC}$ .) The processing time of Pass2, namely  $T_{Pass2}$ , ends when the message arrives at “ $o$ ”. This node takes a time  $T_{P2}$  to process this message. After this, it sends a Pass3 message towards “ $d$ ” to confirm that the lightpath has been configured and, at the same time, it starts the cross-connection to the client ports. The connection is set up when the cross-connection at “ $o$ ” is completed (see  $T_{Setup}$  in Fig. 6.2). If after Pass1 the PIR decides to reject the connection, then Pass2 is used to inform node “ $o$ ” about the decision. In this case, the Pass2 signalling message travels back to “ $o$ ” without triggering any cross-connection in the path it traverses. In this case, upon reception of the Pass2 message, node “ $o$ ” informs the client that the request has been rejected.

The processing time  $T_{Pass3}$  of Pass3 ends when the Pass3 message arrives at “ $d$ ”. This node takes a time  $T_{P3}$  to process this message. After this, it starts the cross-connection to the client ports. Furthermore, node “ $d$ ” locally stores the decision and sends it to the PCE in order to update the network state.

**Example 6.1** Figure 6.3 depicts the course of a Pass1 and a Pass2 message probes in a network with six nodes and seven links. The probes correspond to a class- $j$  request for which  $(o, d)_j = (A, D)$  and  $\Gamma_j = \{(A, E, C, D), (A, B, F, D)\}$ . Since the request arrives at node A, then this node triggers the connection setup process by sending two message probes: one over the path (A, E, C, D) and the other over (A, B, F, D). Each probe collects information about the link states and the long-term link reward gains. Once the two messages arrive at D, this node executes the PIR step. Figure 6.3 shows the Pass2 message when the PIR decides to admit the connection over the path (A, E, C, D). This message triggers the resource allocation process.

The connection setup process in Fig. 6.2 is solely controlled by the node pair  $(o, d)_j$ . The process is always triggered by the node that receives the connection request, whereas the destination node executes the PIR and triggers the resource allocation procedure. Note that although the PCE is not directly involved in the process, it is periodically updating (every  $\Delta T$  time units) the information that the nodes need to establish connections on demand. By this approach, the connection setup process is decentralized, thereby avoiding the exchange of signalling messages between the nodes  $(o, d)_j$  and the PCE. Those messages might cause delays that may lead to large connection setup times. As demonstrated by the results in [SN09, CCC<sup>+</sup>12, SCG<sup>+</sup>12, SGK<sup>+</sup>14], by avoiding the exchange of messages with the PCE, the 3WHS protocol can provide setup times less than the round trip fibre delay ( $RTD$ ) + 50 ms. This feature is useful to provide services with stringent setup time requirements. Moreover, since the 3WHS probes all the routes in  $\Gamma_j$  simultaneously, it provides the information needed by the destination node to calculate decisions on CAC and RSA. This avoids the collection of global information to perform resource allocation, and thus, the decision and connection setup processes are accelerated. In [SCG<sup>+</sup>12] it is shown that to provide setup times less than  $RTD + 50$  ms, the ROADMs would require a signalling processing time less than 580  $\mu$ s,

which is feasible with today's technology. The 3WHS protocol together with the PIA (as seen in Fig. 6.2) can be realized as an extension of the GMPLS [Man04] control plane or included into a software-defined networking (SDN) architecture. The PIR step could also be realized in a purely centralized approach, e.g. within an SDN controller. However, in this case, centralization may impose severe restrictions on the minimum connection setup times attainable by the network.

The time-sequence diagram in Fig. 6.2 shows that the connection setup time is given by:

$$T_{Setup} = T_{CR} + T_{Pass1} + T_{P1} + T_{Pass2} + T_{P2} + T_{XC} \quad (6.1)$$

In general,  $T_{Setup}$ ,  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$ ,  $T_{Pass1}$ ,  $T_{Pass2}$  and  $T_{XC}$  are random variables, and hence, from Equation (6.1) it is not possible to directly calculate an estimate of the connection setup latency. In fact, an accurate estimation of the latency needs to consider the statistical properties of the variables, and has to properly quantify their impact on the protocol performance. To address this issue in more detail, in Section 6.2 we outline a generic method that can be used to estimate the latencies incurred by communication protocols. Then in Section 6.3 the method is applied to formulate a performance model that estimates the latency of the connection setup procedure in Fig. 6.2.

## 6.2. Performance Evaluation of Communication Protocols by Stochastic Task Graphs

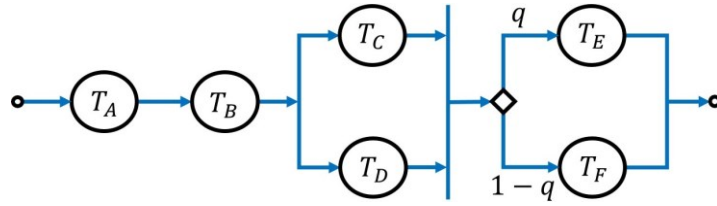
The design of a communication protocol comprises the specification, verification and implementation of the system of rules that define the protocol. An efficient design has to rely on performance evaluations that provide insight into the parameters that determine the protocol behaviour. Such evaluations can be made through experiments, simulations or analytical models. In some cases, depending on the complexity of the protocol design, experiments and simulations can be costly and time-consuming. This justifies the use of analytical models that provide valid descriptions of the protocol performance. If these models give precise approximations, they can be used to perform analyses and improvements on the protocol design. In this section, a method to calculate analytical performance evaluation models for communication protocols is outlined. In the method, protocols are modelled as task graphs which consist of the series of actions (or tasks) executed by a protocol. These tasks are described by stochastic processes that represent the work load imposed by the protocol on the network. The approach is motivated by the work in [Küh14], where a novel task graph reduction strategy is formulated to analyse parallel processing in data centre environments. Such a strategy, as discussed in [Küh16], is also applicable to the analysis of distributed network control mechanisms. In particular, [Küh16] shows that by representing a protocol as a reducible task graph, it is possible to find the probability density function that describes the stochastic properties of the protocol. We will use this result to define a performance evaluation model for the 3WHS protocol.

### 6.2.1 Task Graph Representation of Communication Protocols

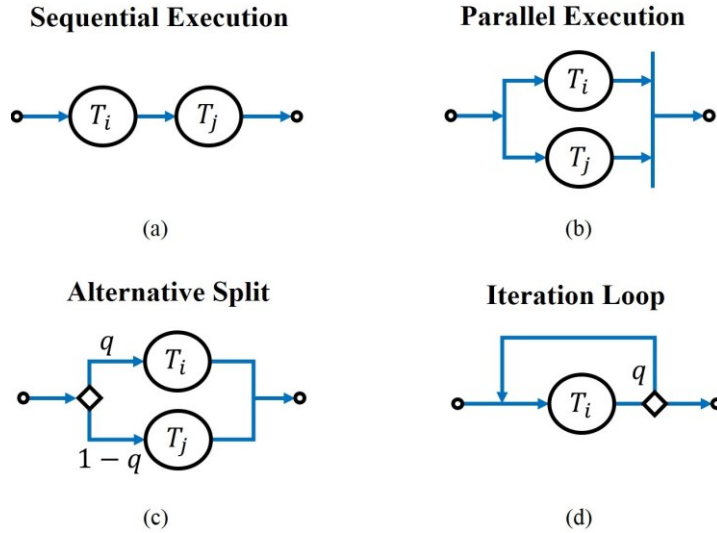
A communication protocol is a system of rules that defines the series of tasks or steps taken to achieve a specific goal, such as the realization of a network control or management function. All tasks consume processing and storage resources in the network, and they may interact with each other by exchanging information messages. The protocol performance is given by the expected value  $E[T]$  of its delay (or execution time)  $T$ , which depends on the stochastic properties of the task execution times and on the interrelationships among tasks. These dependencies can be represented by modelling the protocol as a task graph, which is a diagram made up of three building blocks: circles, arrows and diamonds. The circles denote tasks characterized by random variables  $T_i$  which represent the processing time of a task  $i$ . The arrows are used to illustrate the sequential order in the execution of the tasks they interconnect (i.e. tasks connected to the beginning of an arrow must be executed before those connected to the end). The diamonds represent possible splits during the process of protocol execution.

**Example 6.2** Consider the task graph depicted in Fig. 6.4, which represents a protocol that consists of six tasks characterized by the random variables  $T_A$ ,  $T_B$ ,  $T_C$ ,  $T_D$ ,  $T_E$  and  $T_F$ . The tasks are represented by the circles in the graph, and the arrows define the sequential order whereby the tasks are executed. Task  $A$  is executed first and before task  $B$ ; after  $B$ , tasks  $C$  and  $D$  are executed in parallel. Upon completion of these two tasks there is a protocol split represented by the diamond, which indicates that with probability  $q$ , the protocol performs task  $E$  instead of  $F$ .

In order to calculate the average delay  $E[T]$  for any given communication protocol, the corresponding task graph has to first be built as follows:



**Figure 6.4:** Example of a task graph for a protocol made up of six tasks.



**Figure 6.5:** The four basic task graph components.

1. Given the set of rules defining the protocol, the tasks and their relationships must be defined. This can be obtained from time-sequence diagrams that specify the protocol behaviour. (The diagrams explicitly depict the message flows between the network entities responsible for the execution of tasks.) As a result, a task graph can be built where arrows are used to depict the temporal flow in the process of protocol execution.
2. For every task  $i$ , the stochastic properties of the task processing time  $T_i$  have to be determined. These are: the probability density function (PDF)  $f_i(t)$ , the cumulative distribution function (CDF)  $F_i(t) = P(T_i \leq t)$ , and the expected value  $E[T_i]$ . These properties are given either by analytical models which are known to represent the random variable  $T_i$ , or by statistical analyses made on observations of  $T_i$ .

The task graph represents the workload handled by the network when the protocol is running. With the aid of suitable reduction techniques (provided that the graph is directed and acyclic), the graph can be simplified to a single task whose processing time  $E[T]$  equals the mean delay of the protocol it represents.

### 6.2.2 Task Graph Reduction Methodology

In [Küh14] an approach for the analysis of parallel processing in data centre environments is presented. Therein, a novel mathematical method for the reduction of task graphs is outlined. In the following, we adopt that method to reduce protocol graphs by calculating an equivalent task for any pair of interrelated tasks. For that, in [Küh14] it is shown that a directed acyclic graph is reducible by using the four basic graph components defined in Fig. 6.5.

Two tasks  $i$  and  $j$  which are sequentially executed (see Fig. 6.5a) are equivalent to a single task  $k$  with processing time  $T_k = T_i + T_j$ . The stochastic properties of  $k$  follow from its PDF, which is given by the convolution of  $f_i(t)$  and  $f_j(t)$ :

$$f_k(t) = f_i(t) \otimes f_j(t) = \int_0^t f_i(\tau) \cdot f_j(t - \tau) \cdot d\tau \tag{6.2}$$

When two tasks  $i$  and  $j$  are executed in parallel (see Fig. 6.5b), two cases are distinguished. First, if  $i$  and  $j$  need to be completed before processing any subsequent task, the equivalent task  $k$  has execution time  $T_k = \max(T_i, T_j)$  with PDF given by:

$$f_k(t) = f_i(t) \cdot F_j(t) + f_j(t) \cdot F_i(t) \quad (6.3)$$

Otherwise, if the protocol may execute subsequent tasks once either  $i$  or  $j$  is completed first, then we have  $T_k = \min(T_i, T_j)$  with PDF:

$$f_k(t) = f_i(t) \cdot [1 - F_j(t)] + f_j(t) \cdot [1 - F_i(t)] \quad (6.4)$$

Alternative split, see Fig. 6.5c, refers to the case where two tasks  $i$  and  $j$  are mutually exclusive, this is depicted in the figure by the two branches emerging from the diamond. If tasks  $i$  and  $j$  are performed with probabilities  $q$  and  $(1 - q)$ , respectively, the equivalent task  $k$  has a processing time  $T_k$  with PDF:

$$f_k(t) = q \cdot f_i(t) + (1 - q) \cdot f_j(t) \quad (6.5)$$

Another interesting case occurs when upon completion, a task  $i$  can be re-executed with probability  $q$ . This forms an iteration loop (see Fig. 6.5d) represented by a task  $k$  with a PDF obtained by iteratively applying Equation (6.2):

$$f_k(t) = \sum_{i=0}^{\infty} q^i \cdot (1 - q) \cdot f_i(t) \otimes [f_i(t) \otimes \dots \otimes f_i(t)] \quad (6.6)$$

For each of the four aforementioned cases, the expected value  $E[T_k]$  (or mean processing time) of task  $k$  is calculated as:

$$E[T_k] = \int_0^{\infty} t \cdot f_k(t) \cdot dt \quad (6.7)$$

For a given task graph, the four elements in Fig. 6.5 are used to calculate the PDF  $f(t)$  of the protocol delay  $T$ . From this PDF the stochastic properties of the protocol are derived. For this, a stepwise reduction of tasks is performed on the graph by implementing the following procedure:

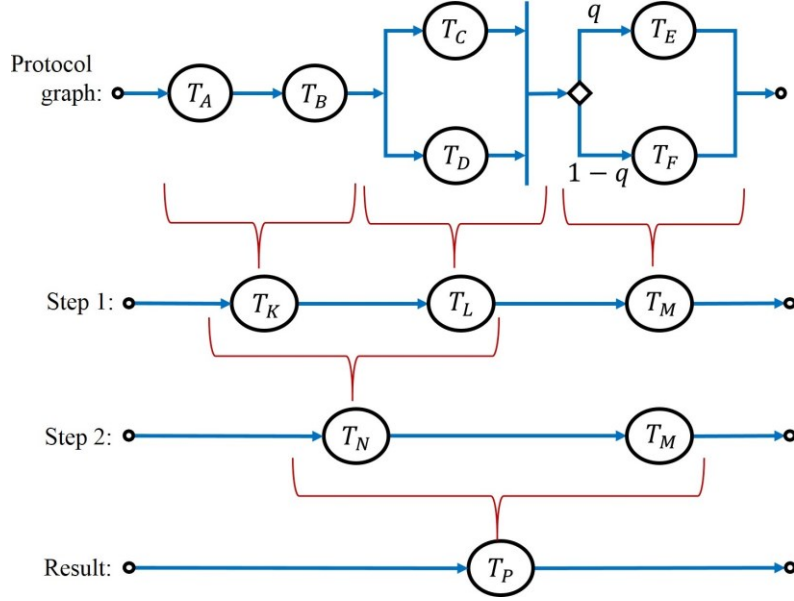
1. Define the task graph for the protocol. This includes the definition of the PDF and the CDF for each task processing time  $T_i$ .
2. Aggregate pairs of interrelated tasks by using Equations (6.2)-(6.6). As a result, each aggregated pair is represented by a task  $k$  with PDF  $f_k(t)$  and expected delay  $E[T_k]$ . This procedure defines a reduced equivalent graph for the protocol.
3. If the equivalent graph obtained in the previous step has a single task  $k$  with processing time  $T_k$ , stop the graph reduction procedure, and define  $f(t) = f_k(t)$  as the PDF of the protocol delay. Otherwise, go back to step two, i.e. start a new graph reduction step.

This method, as formulated in [Küh14], guarantees that if the task graph is directed and acyclic, then a single task is found whose PDF represents the statistical properties of the original graph.

**Example 6.3** In Fig. 6.6 we show the stepwise reduction for the task graph in Fig. 6.4. The six tasks have PDFs  $f_A(t)$ ,  $f_B(t)$ ,  $f_C(t)$ ,  $f_D(t)$ ,  $f_E(t)$  and  $f_F(t)$ . By knowing them, in a first reduction step the task-pairs  $(A, B)$ ,  $(C, D)$  and  $(E, F)$  are, respectively, aggregated into the tasks  $K$ ,  $L$  and  $M$ . From Equation (6.2) the PDF of task  $K$  is calculated as  $f_K(t) = f_A(t) \otimes f_B(t)$ . Assuming that tasks  $C$  and  $D$  need to be completed before continuing with the subsequent tasks, then according to Equation (6.3) task  $L$  has PDF  $f_L(t) = f_C(t) \cdot F_D(t) + f_D(t) \cdot F_C(t)$ . Furthermore, task  $M$  has PDF  $f_M(t) = q \cdot f_E(t) + (1 - q) \cdot f_F(t)$  as it is the split between tasks  $E$  and  $F$ . As a result, an equivalent graph with three tasks, i.e.  $K$ ,  $L$  and  $M$ , is obtained (see Fig. 6.6). In the second step, the tasks  $(K, L)$  are aggregated into task  $N$  with PDF  $f_N(t) = f_K(t) \otimes f_L(t)$ . Then the protocol is now described by a graph with two tasks  $N$  and  $M$  which are further simplified (in the last step) to a task  $P$  with PDF  $f_P(t) = f_N(t) \otimes f_M(t)$ . The statistical properties of the protocol delay are fully described by  $f_P(t)$ . For example, after calculating  $f_P(t)$  through the reduction process, the mean delay is determined from Equation (6.7) as  $E[T_P] = \int_0^{\infty} t \cdot f_P(t) \cdot dt$ .

Communication protocols are event driven processes whose execution is triggered by external requests. Upon arrival of a request, the protocol executes the tasks defined in its graph. If requests arrive according to a general and independently distributed arrival process GI with rate  $\lambda$  (requests/time unit) and





**Figure 6.6:** Task graph reduction for a protocol with six tasks.

the service time (which is the protocol delay  $T$ ) is characterized by the PDF  $f(t)$ , according to [Küh14], the protocol can be represented by a GI/G/1 queuing model. Therefore, the mean protocol delay  $E[T]$  is the expected service time of a request that triggers the protocol execution. In what follows, we use these results to derive an analytical performance evaluation model to analyse connection setup in dynamic optical networks.

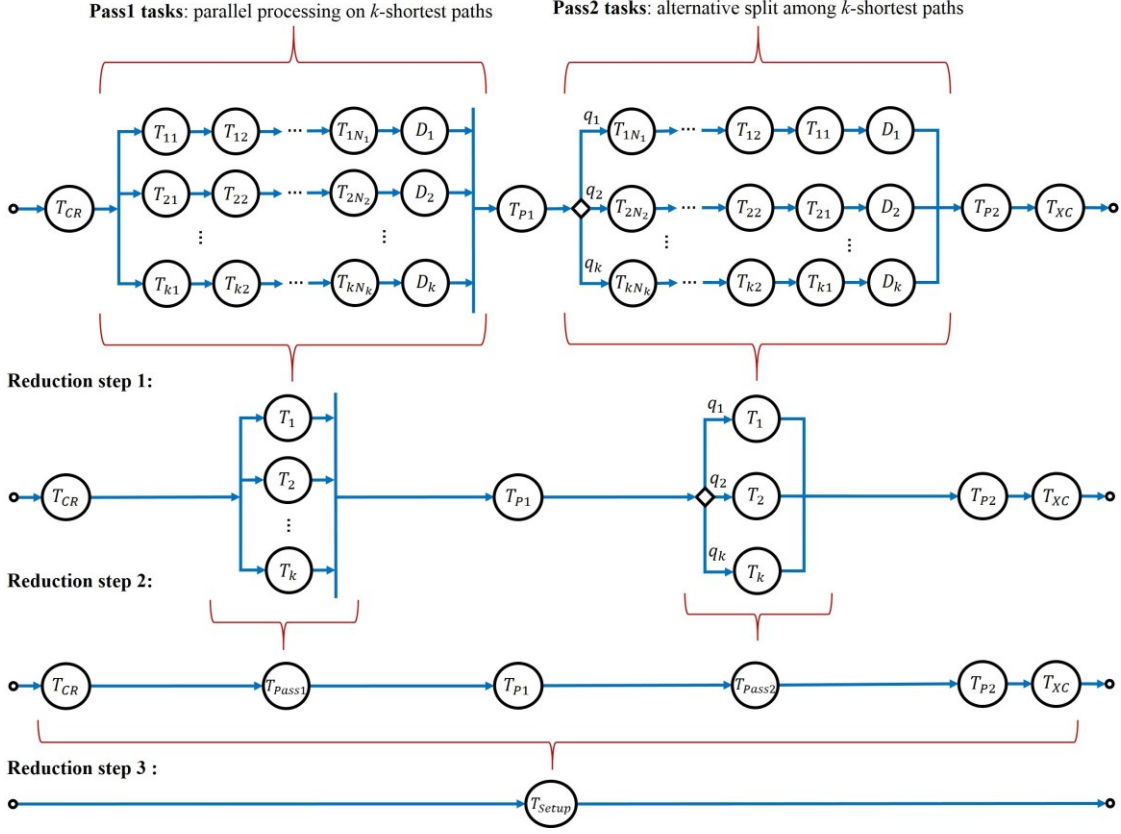
### 6.3. Connection Setup Latency in Dynamic Optical Networks

The graph at the top of Fig. 6.7 represents the task graph for the 3WHS protocol as derived from the time-sequence diagram in Fig. 6.2. From left to right, the graph defines the sequence of protocol steps (or tasks) that set up a connection between a node-pair  $(o, d)_j$ . These two nodes are connected by a set  $\Gamma_j$  of  $k$ -shortest paths. Pass1 and Pass2 are modelled as tasks whose durations, i.e.  $T_{Pass1}$  and  $T_{Pass2}$ , are given by the delays incurred in the paths traversed by their signalling messages. Those delays consist of node processing times and propagation delays. To represent these two delay contributors in Fig. 6.7, let us define  $N_i$  as the number of intermediate nodes (i.e. excluding the nodes “ $o$ ” and “ $d$ ”) in path  $i$ . Therefore, in Fig. 6.7, for the tasks modelling Pass1 and Pass2,  $T_{ij}$  is a random variable that denotes the processing time of a signalling message at the intermediate node  $j$  ( $j = 1, \dots, N_i$ ) which is in the path  $i$  ( $i = 1, \dots, k$ ). Moreover,  $D_i = L_i/C$  is the propagation delay in path  $i$ , with  $L_i$  and  $C$  being the path length and the speed of light in the fibre, respectively. In the graph, Pass1 is modelled as the parallel processing of  $k$ -message probes, where each probe is processed by a shortest path. Furthermore, Pass2 is modelled by an alternative split among the  $k$ -shortest paths, as during this pass, signalling takes place on a selected path. This split is represented by the diamond in the graph.

The properties of the 3WHS protocol are fully characterized by the PDF  $f_{T_{Setup}}(t)$  of the connection setup time  $T_{Setup}$ . From this PDF, the protocol performance can be defined as the mean connection setup latency  $E[T_{Setup}]$ . To calculate  $f_{T_{Setup}}(t)$ , let the random variables  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$ ,  $T_{ij}$ ,  $D_i$  and  $T_{XC}$  have PDFs  $f_{T_{CR}}(t)$ ,  $f_{T_{P1}}(t)$ ,  $f_{T_{P2}}(t)$ ,  $f_{T_{ij}}(t)$ ,  $f_{D_i}(t)$  and  $f_{T_{XC}}(t)$ , respectively. Let us further assume these PDFs to be known. Thus,  $f_{T_{Setup}}(t)$  is obtained by reducing the task graph through the three steps depicted in Fig. 6.7 [RB17c]. In the first reduction step, an equivalent task is found for each of the  $k$ -shortest paths that connects the nodes  $(o, d)_j$ . For path  $i$  ( $i = 1, \dots, k$ ), this task is represented by the random variable  $T_i$ . In Fig. 6.7, for Pass1 and Pass2,  $T_i$  is the processing time of a signalling message through all intermediate nodes  $j$  in the path  $i$  plus the propagation delay  $D_i$ :

$$T_i = \sum_j^{N_i} T_{ij} + D_i, \quad i = 1, \dots, k \quad (6.8)$$

3WHS task graph:



**Figure 6.7:** Task graph for the 3WHS protocol and its intermediate reduction steps [RB17c].

This means that in path  $i$ , a signalling message is processed by  $N_i + 1$  tasks which are sequentially executed. By performing successive aggregations of pairs of tasks in path  $i$ , from Equation (6.2) we have that  $T_i$  has PDF given by:

$$f_{T_i}(t) = f_{T_{i1}}(t) \otimes f_{T_{i2}}(t) \otimes \dots \otimes f_{T_{iN_i}}(t) \otimes f_{D_i}(t) \quad , i = 1, \dots, k \quad (6.9)$$

After calculating  $f_{T_i}(t)$  for all  $i$ , the protocol task graph simplifies to the graph shown in the first reduction step in Fig. 6.7. In the second step, the goal is to calculate the PDFs of  $T_{Pass1}$  and  $T_{Pass2}$ . From Fig. 6.7, it is seen that  $T_{Pass1}$  is a random variable that represents the parallel processing of  $k$  tasks which have to be completed before starting  $T_{P1}$ , i.e. before executing the PIR step. (Recall that in Pass1, the  $k$ -message probes must arrive at the destination node before the PIA may run the PIR.) Hence we have:

$$T_{Pass1} = \max(T_1, T_2, \dots, T_{k-1}, T_k) \quad (6.10)$$

By successive aggregations of pairs of tasks with Equation (6.3), the PDF of  $T_{Pass1}$  is found to be:

$$f_{T_{Pass1}}(t) = \frac{d}{dt} \prod_{i=1}^k F_{T_i}(t) \quad (6.11)$$

where  $F_{T_i}(t)$  is the CDF of  $T_i$ , with  $f_{T_i}(t) = dF_{T_i}(t)/dt$ . Equation (6.11) assumes that the  $k$  variables  $T_i$  are independent. Regarding Pass2, from Fig. 6.7, we have that  $T_{Pass2}$  is a random variable that models the alternative split among the  $k$ -shortest paths. The reason is that in this pass, signalling only takes place on the path over which the connection is to be established. Therefore, if  $q_i$  is the probability that the PIR decides to route a connection on the path  $i$ , then by applying Equation (6.5) we have:

$$f_{T_{Pass2}}(t) = \sum_{i=1}^k q_i \cdot f_{T_i}(t) \quad (6.12)$$

with  $\sum_{i=1}^k q_i = 1$ . The probabilities  $q_i$  can be obtained from statistical analyses made on the routing decisions made by the PIA. Having calculated  $f_{T_{Pass1}}(t)$  and  $f_{T_{Pass2}}(t)$ , the six tasks resulting from step

two are sequentially aggregated in the third reduction step. Thus, with Equation (6.2) it is found that the connection setup time  $T_{Setup}$  has a probability distribution function defined by:

$$f_{T_{Setup}} = f_{T_{CR}}(t) \otimes f_{T_{Pass1}}(t) \otimes f_{T_{P1}}(t) \otimes f_{T_{Pass2}}(t) \otimes f_{T_{P2}}(t) \otimes f_{T_{XC}}(t) \quad (6.13)$$

Therefore, the mean connection setup latency between the nodes  $(o, d)_j$  is:

$$E[T_{Setup}] = \int_0^{\infty} t \cdot f_{T_{Setup}} \cdot dt \quad (6.14)$$

which from Equation (6.13) simplifies to [RB17c]:

$$E[T_{Setup}] = E[T_{CR}] + E[T_{Pass1}] + E[T_{P1}] + E[T_{Pass2}] + E[T_{P2}] + E[T_{XC}] \quad (6.15a)$$

where:

$$E[T_{CR}] = \int_0^{\infty} t \cdot f_{T_{CR}}(t) \cdot dt \quad (6.15b)$$

$$E[T_{Pass1}] = \int_0^{\infty} t \cdot \frac{d}{dt} \prod_{i=1}^k F_{T_i}(t) \cdot dt \quad (6.15c)$$

$$E[T_{P1}] = \int_0^{\infty} t \cdot f_{T_{P1}}(t) \cdot dt \quad (6.15d)$$

$$E[T_{Pass2}] = \sum_{i=1}^k q_i \cdot \int_0^{\infty} t \cdot f_{T_i}(t) \cdot dt \quad (6.15e)$$

$$E[T_{P2}] = \int_0^{\infty} t \cdot f_{T_{P2}}(t) \cdot dt \quad (6.15f)$$

$$E[T_{XC}] = \int_0^{\infty} t \cdot f_{T_{XC}}(t) \cdot dt \quad (6.15g)$$

Equations (6.15) define a performance evaluation model that estimates the connection setup latency of the 3WHS protocol. For class- $j$  connections established between a pair of nodes  $(o, d)_j$ , the equations are used as follows:

1. Define the PDFs and the mean values of the variables  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$ ,  $T_{ij}$  and  $T_{XC}$ . Moreover, for each path  $i$  in the set  $\Gamma_j$ , calculate the propagation delay as  $D_i = L_i/C$ .
2. For each path  $i$ , use the PDFs  $f_{T_{ij}}(t)$  and  $f_{D_i}(t)$  to calculate, with Equation (6.9), the PDF  $f_{T_i}(t)$  of the path processing time  $T_i$ . Observe that  $f_{D_i}(t) = \delta(t - D_i)$  since the propagation delay is deterministic, with  $\delta$  being the Dirac delta function.
3. Use  $f_{T_i}(t)$ ,  $i = 1, \dots, k$ , to calculate  $E[T_{Pass1}]$  with Equation (6.15c). For this, the CDFs  $F_{T_i}(t)$  are determined by solving the differential equation  $f_{T_i}(t) = dF_{T_i}(t)/dt$ .
4. Use  $f_{T_i}(t)$ ,  $i = 1, \dots, k$ , to calculate  $E[T_{Pass2}]$  with Equation (6.15e).
5. Calculate  $E[T_{Setup}]$  with equation (6.15a).

This calculation procedure shows that, besides the path propagation delays,  $E[T_{Pass1}]$  and  $E[T_{Pass2}]$  are strongly dependent on the stochastic properties of the node processing times  $T_{ij}$ . (This can be understood by noticing that  $f_{T_i}(t)$  is determined by  $f_{T_{ij}}(t)$  and  $f_{D_i}(t)$ .) In order to illustrate this dependence better, let us study two examples where  $E[T_{Setup}]$  is calculated for the case of random and deterministic node processing times  $T_{ij}$ .

### 6.3.1 Example of a Model with Stochastic Task Processing Times

To illustrate the applicability of the performance model for the 3WHS protocol, consider the case where the expected values  $E[T_{CR}] = \bar{T}_{CR}$ ,  $E[T_{P1}] = \bar{T}_{P1}$ ,  $E[T_{P2}] = \bar{T}_{P2}$  and  $E[T_{XC}] = \bar{T}_{XC}$  are known from their corresponding PDFs. The path propagation delays  $D_i = L_i/C$  are deterministic and known as well.

Let us calculate  $E[T_{Setup}]$  for the case of exponentially distributed processing times  $T_{ij}$  with mean  $E[T_{ij}] = \bar{T}_{SP}$ . Therefore, the PDF of  $T_{ij}$  is defined as:

$$f_{T_{ij}}(t) = \frac{1}{\bar{T}_{SP}} \exp\left(-\frac{t}{\bar{T}_{SP}}\right), \quad \forall i, j \quad t \geq 0 \quad (6.16)$$

Then to estimate  $E[T_{Setup}]$  it is necessary to calculate  $E[T_{Pass1}]$  and  $E[T_{Pass2}]$ . For that, the first step is to determine the PDF and the CDF of the random variables  $T_i$ . By plugging Equation (6.16) into Equation (6.9), the distribution of the processing time  $T_i$  of a Pass1/Pass2 signalling message on path  $i$  is:

$$f_{T_i}(t) = \begin{cases} 0, & t < D_i \\ \frac{[(t-D_i)/\bar{T}_{SP}]^{N_i-1}}{\bar{T}_{SP} \cdot (N_i-1)!} \exp\left(-\frac{(t-D_i)}{\bar{T}_{SP}}\right), & t \geq D_i \end{cases} \quad (6.17)$$

with a corresponding CDF given by:

$$F_{T_i}(t) = \begin{cases} 0, & t < D_i \\ \int_{D_i}^t \frac{[(y-D_i)/\bar{T}_{SP}]^{N_i-1}}{\bar{T}_{SP} \cdot (N_i-1)!} \exp\left(-\frac{(y-D_i)}{\bar{T}_{SP}}\right) \cdot dy, & t \geq D_i \end{cases} \quad (6.18)$$

which shows that  $T_i$  is a gamma distributed random variable. As a result, from Equation (6.17) a signalling message has on path  $i$  a mean processing delay equal to:

$$E[T_i] = \int_{D_i}^{\infty} t \cdot f_{T_i}(t) \cdot dt = D_i + N_i \cdot \bar{T}_{SP} \quad (6.19)$$

Furthermore, by plugging Equation (6.18) into Equation (6.15c), it is verified that  $E[T_{Pass1}]$  is given by:

$$E[T_{Pass1}] = \int_{\max(D_1, \dots, D_k)}^{\infty} t \cdot \frac{d}{dt} \prod_{i=1}^k F_{T_i}(t) \cdot dt \quad (6.20)$$

whereas from Equations (6.17) and Equation (6.15e), we find that  $E[T_{Pass2}]$  is equal to:

$$E[T_{Pass2}] = \sum_{i=1}^k q_i \cdot [D_i + N_i \cdot \bar{T}_{SP}] \quad (6.21)$$

Thus, if the node processing times  $T_{ij}$  are exponentially distributed with mean  $\bar{T}_{SP}$ , then between the nodes  $(o, d)_j$  a connection is expected to have a setup latency:

$$E[T_{Setup}] = \bar{T}_{CR} + \int_{\max(D_1, \dots, D_k)}^{\infty} t \cdot \frac{d}{dt} \prod_{i=1}^k F_{T_i}(t) \cdot dt + \bar{T}_{P1} + \sum_{i=1}^k q_i \cdot [D_i + N_i \cdot \bar{T}_{SP}] + \bar{T}_{P2} + \bar{T}_{XC} \quad (6.22)$$

Observe that this latency depends on the number  $k$  of shortest paths in the set  $\Gamma_j$ .

### 6.3.2 Example of a Model with Deterministic Task Processing Times

Consider the case where the variables  $T_{CR} = \bar{T}_{CR}$ ,  $T_{P1} = \bar{T}_{P1}$ ,  $T_{P2} = \bar{T}_{P2}$ ,  $T_{XC} = \bar{T}_{XC}$ ,  $D_i = L_i/C$  and  $T_{ij} = \bar{T}_{SP}$  are deterministic. Thus, on path  $i$ , a signalling message has a processing delay  $T_i$  equal to:

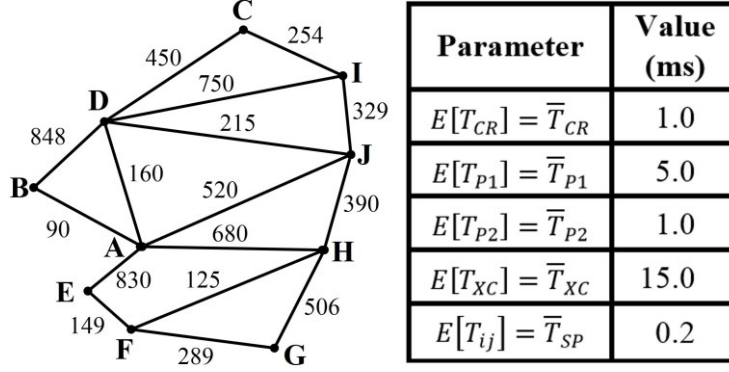
$$T_i = D_i + N_i \cdot \bar{T}_{SP} \quad (6.23)$$

which is deterministic as well. Then from Equation (6.15c) it follows that  $E[T_{Pass1}]$  is:

$$E[T_{Pass1}] = \max(T_1, \dots, T_i, \dots, T_k) \quad (6.24)$$

Likewise, according to Equation (6.15e), we find that  $E[T_{Pass2}]$  is given by:

$$E[T_{Pass2}] = \sum_{i=1}^k q_i \cdot [D_i + N_i \cdot \bar{T}_{SP}] \quad (6.25)$$



**Figure 6.8:** Optical network with distances in Km and definition of parameters [RB17c].

Therefore, if the task processing times are deterministic, then between the nodes  $(o, d)_j$  a connection is expected to have a setup latency:

$$E[T_{Setup}] = \bar{T}_{CR} + \max(T_1, \dots, T_k) + \bar{T}_{P1} + \sum_{i=1}^k q_i \cdot [D_i + N_i \cdot \bar{T}_{SP}] + \bar{T}_{P2} + \bar{T}_{XC} \quad (6.26)$$

Note that although the task processing times are deterministic,  $T_{Setup}$  is a random variable. The reason is that unlike  $T_{Pass1}$ , the delay  $T_{Pass2}$  is random since its actual value (for a given connection request) depends on the path selected by the PIR to allocate resources. Thus, in Equation (6.26) Pass2 has a latency estimated by  $E[T_{Pass2}]$ , which is the weighted average of the path processing delays  $T_i$ . The weights are the probabilities  $q_i$ .

The performance evaluation models in Equation (6.22) and Equation (6.26) are not alike. This shows that the statistical properties of the task processing times are key to the estimation of the connection setup latency. In the next section, simulations are used to study how well these two models estimate  $E[T_{Setup}]$ .

## 6.4. Evaluation of the Analytical Performance Model for Connection Setup Latency

In this section we assess the analytical performance model defined by Equations (6.15). For that, consider the optical network in Fig. 6.8, which has 10 nodes and 16 links with capacities of  $C_l = 32$  slots. The 10 network nodes define 45 node-pairs, each one serving two connection classes, namely, a narrowband and a wideband class that request  $b_j = 2$  and  $b_j = 8$  adjacent slots, respectively. Thus, the network serves 90 connection classes, all of them with reward parameters  $r_j = 1$  (ru). The holding-times are exponentially distributed with mean  $\mu_j^{-1} = 1$  s for narrowband and  $\mu_j^{-1} = 10$  s for wideband traffic. Connection requests arrive following a Poisson process with mean rate  $\lambda_j = 0.01$  (con/s). Therefore, the network is offered a traffic load of  $A = \sum_{j=1}^{90} A_j \cong 5.0$  Erlangs. At the end of Pass1, lightpaths are calculated by the PIA with the MDP-PGMC rule. For the 45 node-pairs, the sets  $\Gamma_j$  contain  $k = 2$  shortest paths. Since the network is offered a traffic of  $A \cong 5.0$  Erlangs, from the performance evaluation results in Table 5.13 - Chapter 5, we have that for the PIA applying the MDP-PGMC rule,  $q_1 \cong 0.76$  and  $q_2 \cong 0.24$ .

The connection setup latency is assessed in two cases, namely, for exponentially distributed and for deterministic task processing times. In the former case, the analytical model is given by Equation (6.22), whereas in the latter by Equation (6.26). In both cases, the variables  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$ ,  $T_{XC}$  and  $T_{ij}$  have the mean values shown in Fig 6.8, which were originally proposed in [CCC<sup>+</sup>12, SCG<sup>+</sup>12, SN09]. Furthermore, to investigate the effect of the propagation delays, both cases are evaluated for four different network sizes. Let  $D_i$  be the path propagation delays calculated from the link lengths in Fig. 6.8 by assuming the speed of light in a fibre link as  $C = 250000$  km/s. We then evaluate the latencies when the propagation delays scale as  $D_i \times 0$ ,  $D_i \times 0.1$ ,  $D_i \times 1$  and  $D_i \times 10$ . (For the sake of clarity,  $D_i \times \alpha$  represents a scenario where all link lengths are multiplied by the factor  $\alpha$ .)

A further remark is needed regarding the analytical model for the case of exponentially distributed task processing times. For  $k = 2$  shortest paths, the integral in Equation (6.22) that defines  $E[T_{Pass1}]$  is solved by the PDF:

$$f_{T_{Pass1}}(t) = \frac{[(t-D_1)/\bar{T}_{SP}]^{N_1-1}}{\bar{T}_{SP} \cdot (N_1-1)!} \exp\left(-\frac{(t-D_1)}{\bar{T}_{SP}}\right) \cdot \frac{\gamma\left(N_2, \frac{t-D_2}{\bar{T}_{SP}}\right)}{(N_2-1)!} + \frac{[(t-D_2)/\bar{T}_{SP}]^{N_2-1}}{\bar{T}_{SP} \cdot (N_2-1)!} \exp\left(-\frac{(t-D_2)}{\bar{T}_{SP}}\right) \cdot \frac{\gamma\left(N_1, \frac{t-D_1}{\bar{T}_{SP}}\right)}{(N_1-1)!} \quad (6.27)$$

which is obtained by plugging Equation (6.18) into Equation (6.11). Furthermore, in Equation (6.27),  $\gamma\left(N_i, \frac{t-D_i}{\bar{T}_{SP}}\right)$  is the lower incomplete gamma function given by:

$$\gamma\left(N_i, \frac{t-D_i}{\bar{T}_{SP}}\right) = \int_0^{(t-D_i)/\bar{T}_{SP}} y^{N_i-1} \cdot \exp(-y) \cdot dy \quad (6.28)$$

The analytical result for  $E[T_{Pass1}]$  is not shown here, as it is too voluminous and can easily be derived from Equation (6.15c) by solving the integral  $E[T_{Pass1}] = \int_0^\infty t \cdot f_{T_{Pass1}}(t) \cdot dt$ .

For each case (and node-pair), discrete event simulations were carried out to determine the mean values and 95% confidence intervals of  $T_{Setup}$ . The results are compared with the latencies predicted by Equation (6.22) and Equation (6.26). All simulations were performed in a simulation framework implemented in R [Rpr17, LDL13], running on a 6-core Intel X86 server with 64 GB RAM.

#### 6.4.1 Analytical and Simulated Connection Setup Latencies

In Fig. 6.9 and Fig. 6.10 we present the results for the case of deterministic and exponentially distributed task processing times, respectively. For each node-pair, the figures compare the mean connection setup latency obtained by simulations (plotted with 95% confidence intervals) with the numerical results given by the analytical models. As seen, for all considered network sizes (or equivalently propagation delays), the analytical evaluations give precise estimates of  $E[T_{Setup}]$ . As expected, the latency increases with the network size. However, note that in both cases, independently of the network size, the task processing times  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$  and  $T_{XC}$  contribute 22 ms to the connection setup time. (Recall that the mean values of these variables - as defined in Fig. 6.8 - are the same regardless of their PDFs, and thus,  $E[T_{CR}] + E[T_{P1}] + E[T_{P2}] + E[T_{XC}] = 1 + 5 + 1 + 15 = 22$  ms in Equation (6.22) and Equation (6.26).) Since these four delays are independent of the path propagation delays  $D_i$ , we have that the network size only influences  $E[T_{Setup}]$  through the latencies of Pass1 and Pass2 (which use the network links to transport their signalling messages). In Fig. 6.11 and Fig. 6.12 we depict the simulated and the analytical  $E[T_{Pass1}]$  and  $E[T_{Pass2}]$  for the latencies in Fig. 6.9 and in Fig. 6.10, respectively. For all node-pairs, it is observed that  $E[T_{Pass1}] \geq E[T_{Pass2}]$ , which shows that Pass1 is the pass that impacts  $E[T_{Setup}]$  the most.

To gain a more comprehensive insight into the properties of the connection setup latency, consider the stacked bar charts in Fig 6.13 and Fig 6.14. These charts depict the percentage contribution of the protocol tasks to the latencies shown in Fig. 6.9 and Fig. 6.10, respectively. From the charts we draw the following observations common to both considered cases:

1. When the propagation delays are low, i.e. for short link lengths (see in the charts the results for  $D_i = 0$  and  $D_i \times 0.1$ ), the dominant delay contributor is the cross-connection time  $T_{XC}$ . (Recall that, as seen in Fig. 6.2,  $T_{XC}$  is the time that the node "o" needs to cross-connect to the client ports.) This delay represents approximately 65% of the total connection setup latency. The second contributor is  $T_{P1}$ , i.e. the time needed to process the Pass1 signalling messages and to execute the PIR. This delay accounts (on average) for 22% of the total latency. On the other hand, the effect of  $T_{Pass1}$  and  $T_{Pass2}$  is marginal. However, these delays are not zero because although the path propagation delays are negligible, the Pass1/Pass2 messages are processed by the nodes in the paths they traverse. (Actually, in the charts for  $D_i = 0$ , the bars depicting  $T_{Pass1}$  and  $T_{Pass2}$  solely include the sum of the processing times  $T_{ij}$  of the intermediate path nodes.)
2. As the network size grows (see the bar charts for  $D_i \times 1$  and  $D_i \times 10$ ),  $T_{Pass1}$  and  $T_{Pass2}$  become dominant (with  $T_{Pass1}$  being higher than  $T_{Pass2}$ ). Hence, in networks with large links, the path propagation delays are the determinant factors in the connection setup latency. It is worth noting that even for large network sizes, the cross-connection time  $T_{XC}$  is not negligible. For instance, when the propagation delays scale as  $D_i \times 10$ ,  $T_{XC}$  accounts (on average) for 17% of the overall latency.

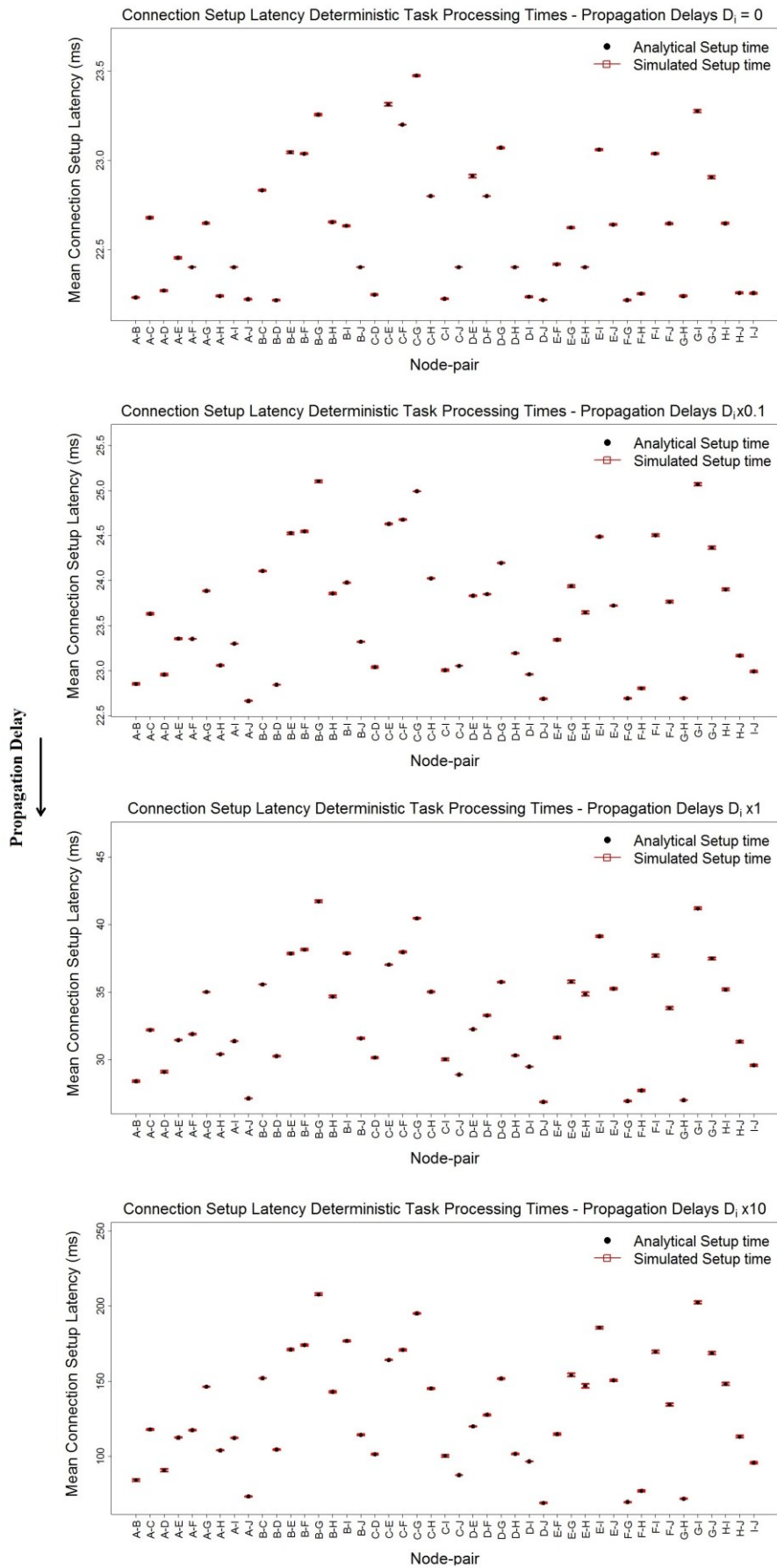


Figure 6.9: Connection setup latencies for the case of deterministic task processing times.

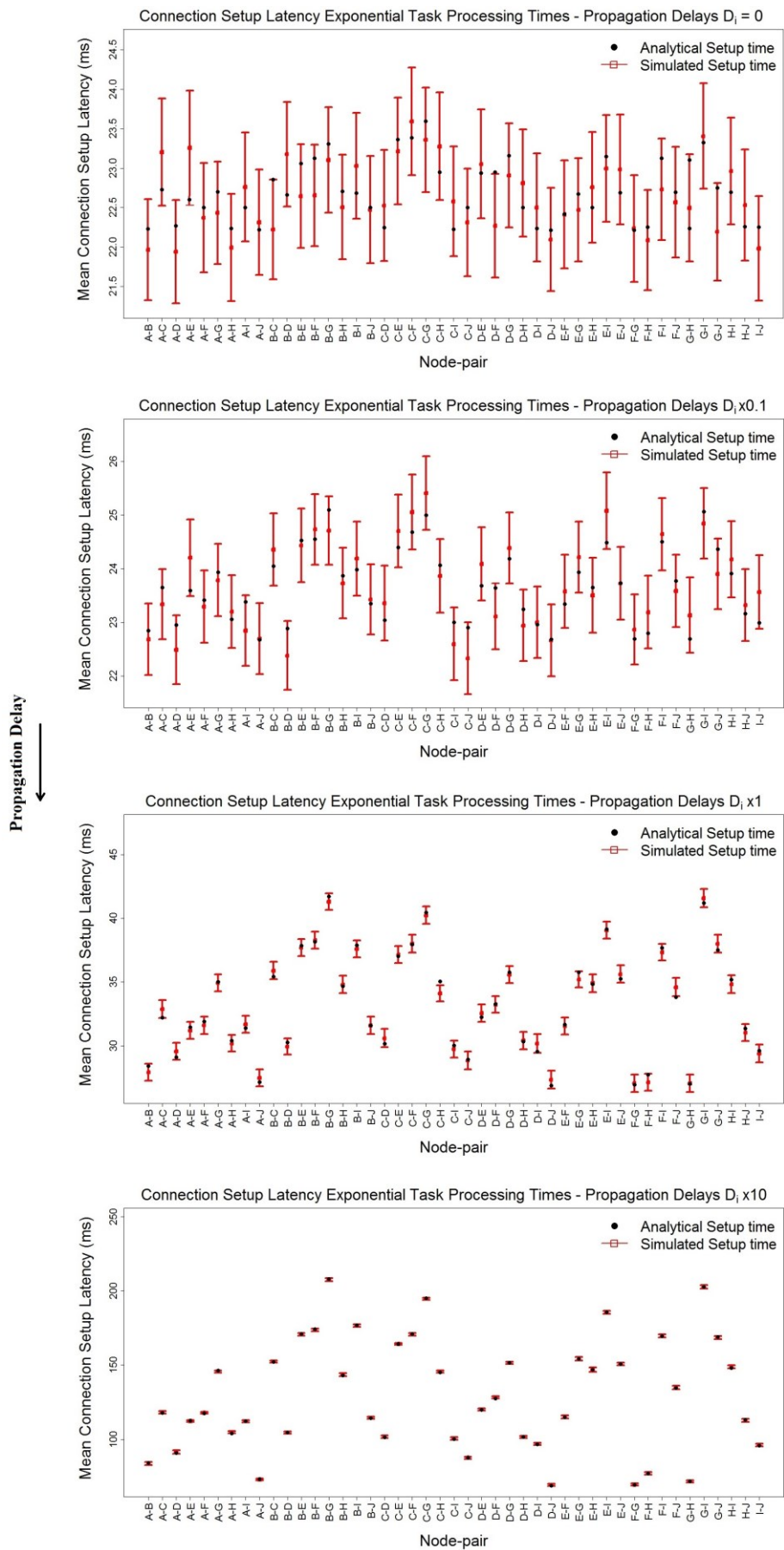


Figure 6.10: Connection setup latencies for the case of exponential processing times.



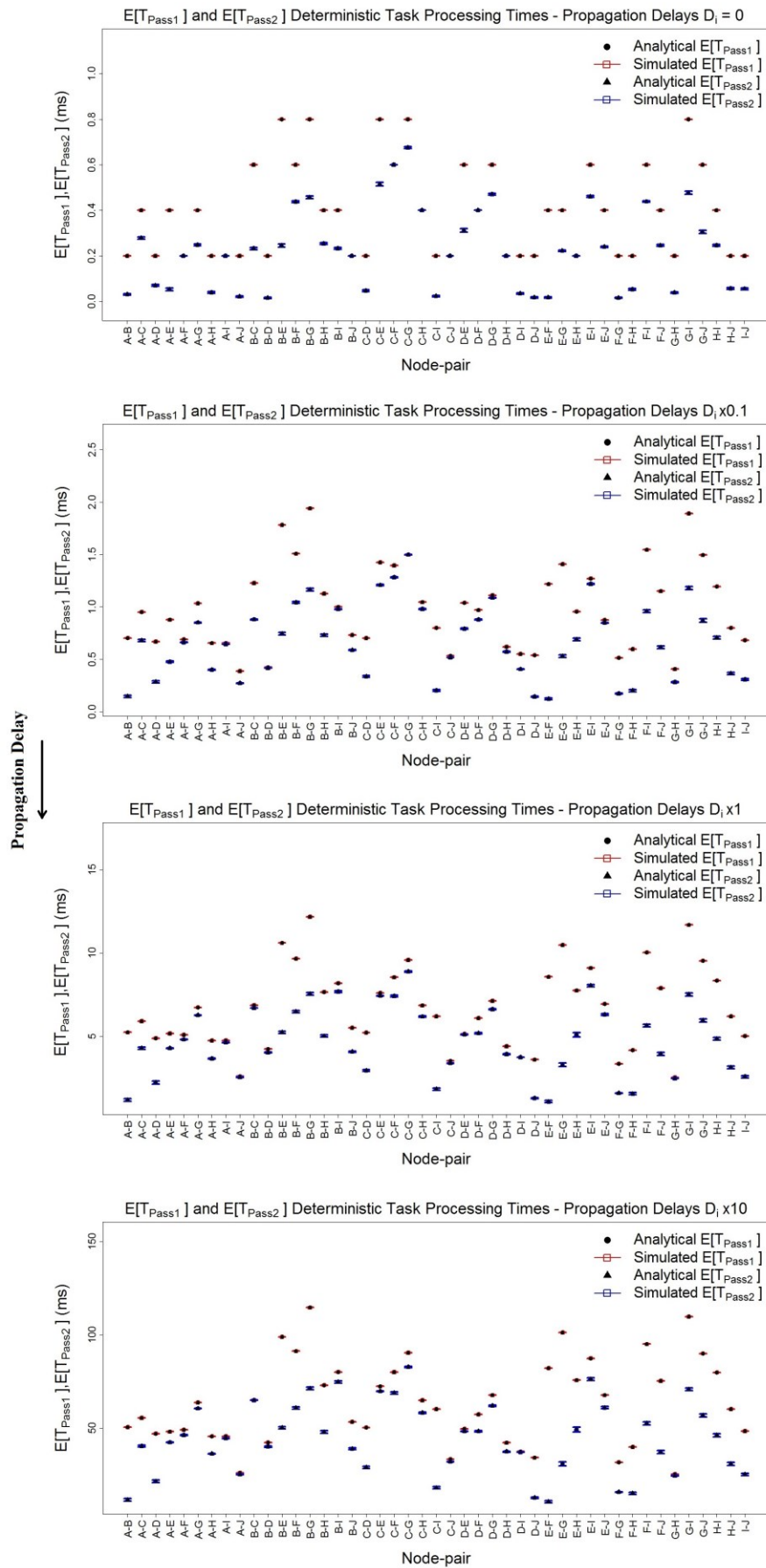


Figure 6.11: Pass1 and Pass2 latencies for the case of deterministic task processing times.

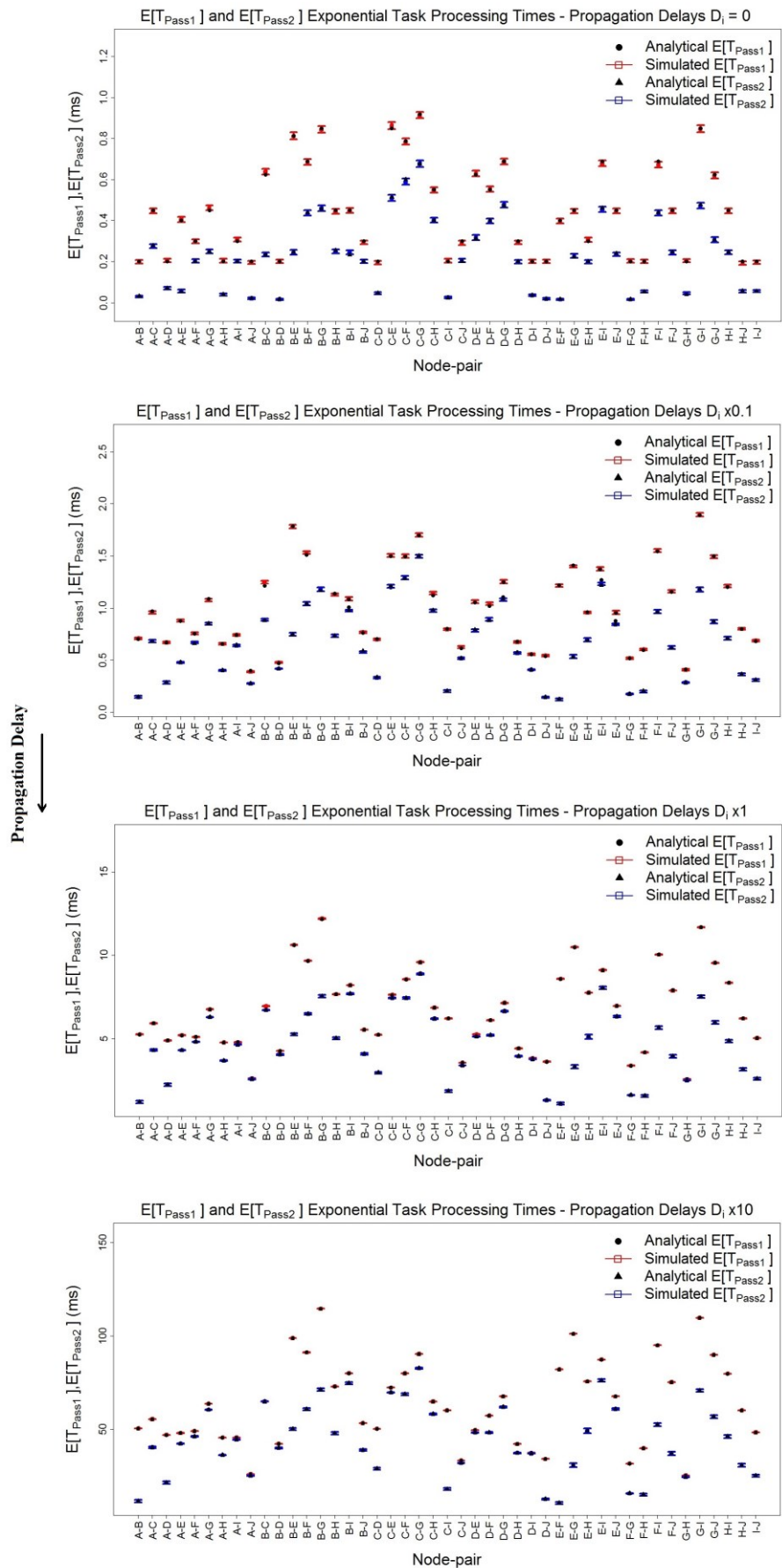


Figure 6.12: Pass1 and Pass2 latencies for the case of exponential processing times.

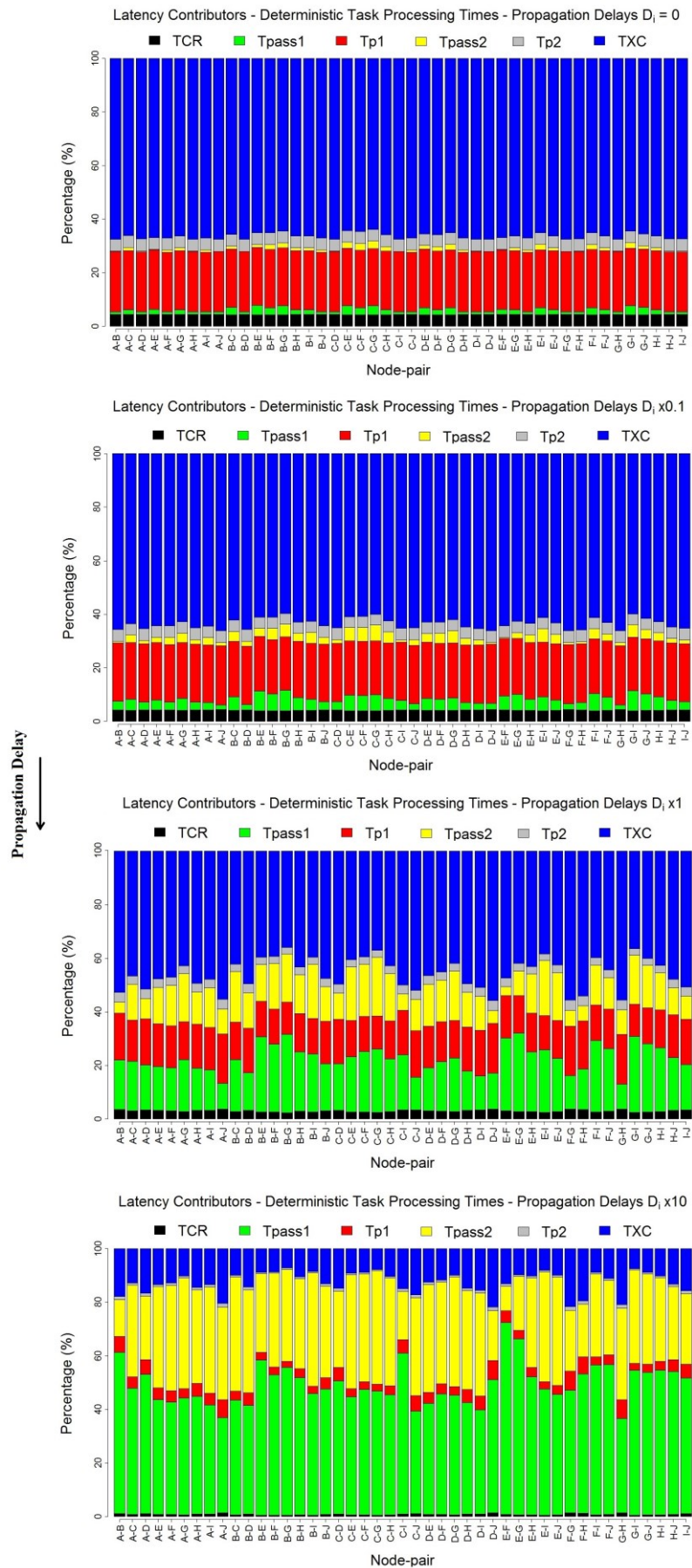


Figure 6.13: Latency contributors for the case of deterministic task processing times.

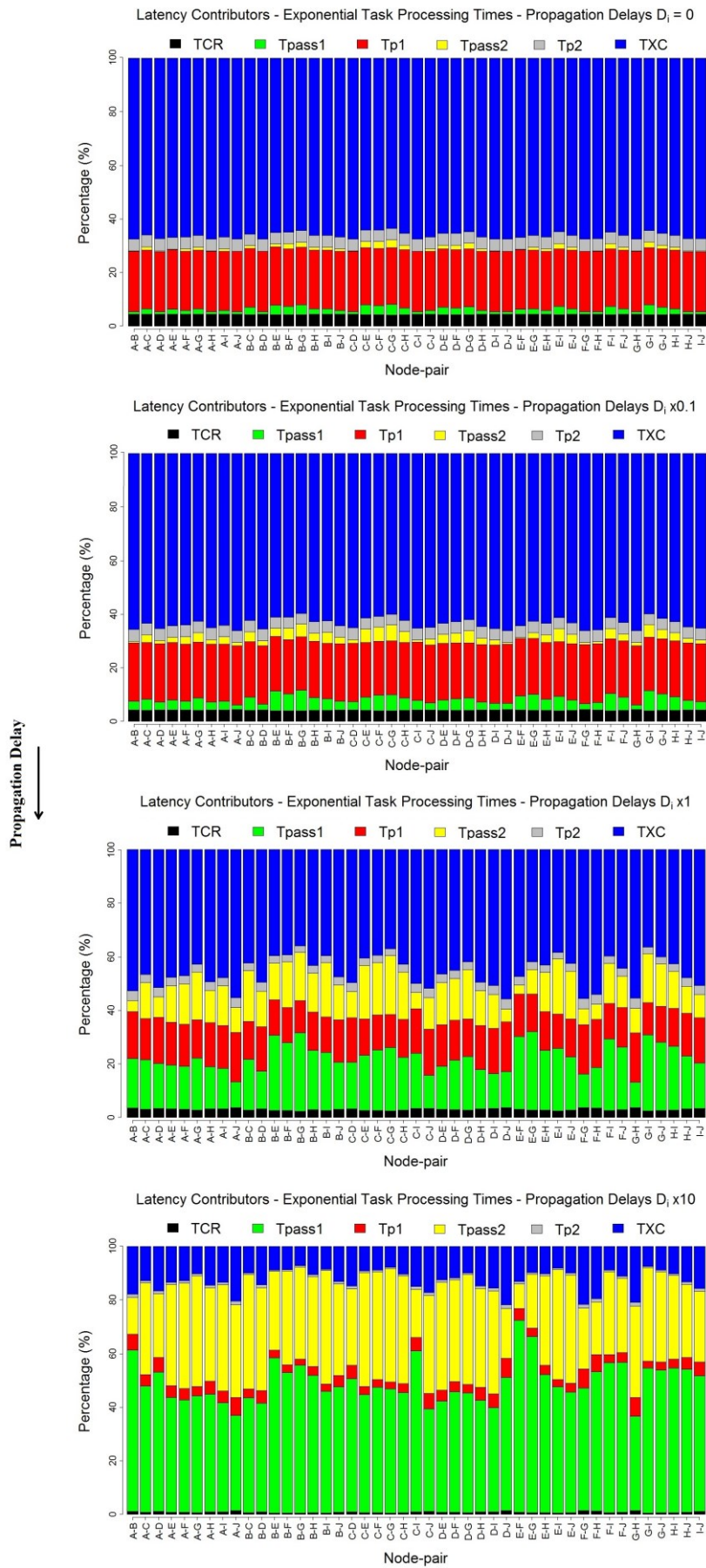


Figure 6.14: Latency contributors for the case of exponential processing times.

From the results, on the one hand, it can be seen that key properties of the connection setup protocol are not strongly dependent on the PDFs of the task processing times. On the other hand, for the exponentially distributed and deterministic task processing times the analytical results match well to the simulation results. This supports the hypothesis that the task-graph reduction method can be conveniently applied to determine the connection setup latency for arbitrary PDFs of  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$ ,  $T_{ij}$  and  $T_{XC}$ .

## 6.4.2 Estimating Boundaries on the Protocol Parameters

Besides estimating the connection setup latency, the analytical model is useful to study how the protocol parameters influence the performance. For example, it may quantify the requirements needed from the task processing times in order to meet a given performance objective. Let  $X$  be a desired upper bound on the connection setup latency  $T_{Setup}$ , and let  $\rho$  be the probability that  $T_{Setup} \leq X$ . Then from the PDF of  $T_{Setup}$  we have:

$$\rho = P\{T_{Setup} \leq X\} = \int_0^X f_{T_{Setup}} \cdot dt \quad (6.29)$$

For a given  $\rho$  and  $X$ , Equation (6.29) can be used to determine the variables  $T_{CR}$ ,  $T_{P1}$ ,  $T_{P2}$ ,  $T_{ij}$ , and  $T_{XC}$  so as to meet more stringent setup times (as defined by  $X$ ). As an example, assume the analytical model for deterministic task processing times in Equation (6.26), and assume the mean values  $\bar{T}_{CR}$ ,  $\bar{T}_{P1}$ ,  $\bar{T}_{P2}$  and  $\bar{T}_{XC}$  to be known. From Equation (6.29), for a given node-pair, the maximum node processing times  $T_{ij} = \bar{T}_{SP}$  for which  $T_{Setup} \leq X$ , with probability  $\rho = 1$ , are obtained by solving the inequality:

$$\bar{T}_{CR} + T_{Pass1} + \bar{T}_{P1} + T_{Pass2} + \bar{T}_{P2} + \bar{T}_{XC} \leq X \quad (6.30)$$

where  $T_{Pass1} = \max(T_1, \dots, T_k)$ . Since  $\rho = 1$ , the inequality must be solved for the worst-case scenario, which occurs when a connection is established on the largest path in  $\Gamma_j$ . In that case,  $T_{Pass2} = T_{Pass1} = \max(T_1, \dots, T_k)$ , thereby from Equation (6.8) Pass1 and Pass2 have latencies equal to:

$$T_{Pass2} = T_{Pass1} = D_{max} + N_{max} \cdot \bar{T}_{SP} \quad (6.31)$$

with  $D_{max}$  and  $N_{max}$  being, respectively, the propagation delay and the number of intermediate nodes of the path with the maximum processing time  $T_i$ . Equation (6.30) then simplifies to:

$$\bar{T}_{CR} + \bar{T}_{P1} + 2 \cdot (D_{max} + N_{max} \cdot \bar{T}_{SP}) + \bar{T}_{P2} + \bar{T}_{XC} \leq X \quad (6.32)$$

Therefore, to guarantee that  $T_{Setup} \leq X$ , with probability  $\rho = 1$ , the node processing times must fulfil:

$$\bar{T}_{SP} \leq \frac{X - (\bar{T}_{CR} + \bar{T}_{P1} + \bar{T}_{P2} + \bar{T}_{XC}) - 2 \cdot D_{max}}{2 \cdot N_{max}} \quad (6.33)$$

In [Sal06, Sal07, CCC+12, SCG+12, SN09] it is argued that, in a future, optical connections with very fast setup times might need to fulfil  $T_{Setup} \leq RTD + 50$  ms, with  $RTD$  being the round trip fibre propagation delay. Thus, with the mean values  $\bar{T}_{CR}$ ,  $\bar{T}_{P1}$ ,  $\bar{T}_{P2}$  and  $\bar{T}_{XC}$  in Fig. 6.8, if  $X = RTD + 50$  ms and  $RTD = 2 \cdot D_{max}$ , from Equation (6.33):

$$\bar{T}_{SP} \leq \frac{RTD + 50 - (1 + 5 + 1 + 15) \cdot RTD}{2 \cdot N_{max}} \quad (6.34)$$

$$\bar{T}_{SP} \leq \frac{14}{N_{max}} \quad (6.35)$$

In [SCG+12] a basic latency model is derived for the 3WHS protocol from which it is concluded that for paths with  $N_{max} = 25$ ,  $\bar{T}_{SP} \leq 0.58$  ms. This result approximates that obtained by Equation (6.35), i.e.  $\bar{T}_{SP} \leq \frac{14}{25} = 0.56$  ms. However, it is worth pointing out that the latency analysis presented in [SCG+12] oversimplifies the calculation of  $T_{Setup}$ . It assumes that the network node-pairs are connected by a single shortest path. Furthermore, it disregards any effect that the PDFs of the task processing times have on

$T_{Setup}$ . More specifically, from the analysis in [SCG<sup>+</sup>12], given that  $T_{Setup}$  is distributed as  $f_{T_{Setup}}$ , it is not possible to estimate  $\bar{T}_{SP}$  for a desired  $\rho$  and  $X$ .

Recall that the results in Fig. 6.9-6.14 are derived by setting  $\bar{T}_{SP} = 0.2$  ms (see Fig. 6.8). This value has been suggested as a reasonable estimate of  $T_{ij}$  by the results in [SN09]. With that estimate, we have that the connection setup latencies in Fig. 6.9-6.10 fulfil  $T_{Setup} \leq RTD + 50$  ms, i.e. they satisfy Equation (6.35). The reason is that in the network, the node-pairs with the largest count of intermediate nodes have  $N_{max} = 5$ , thereby  $\bar{T}_{SP} \leq \frac{14}{5} = 2.8$  ms, which is verified by the estimate  $\bar{T}_{SP} = 0.2$  ms.

Equation (6.33) can also be reformulated to answer the question: for a given estimate of  $\bar{T}_{SP}$ , what is the maximum number  $N_{max}$  for which  $T_{Setup} \leq X$ , with probability  $\rho = 1$ ? The answer follows from:

$$N_{max} \leq \frac{X - (\bar{T}_{CR} + \bar{T}_{P1} + \bar{T}_{P2} + \bar{T}_{XC}) - 2 \cdot D_{max}}{2 \cdot \bar{T}_{SP}} \quad (6.36)$$

If  $X = RTD + 50$  ms and  $RTD = 2 \cdot D_{max}$ , with the values in Fig. 6.8, every node-pair must have  $N_{max} \leq 70$ . This inequality is fulfilled by the partial-mesh network in Fig. 6.8. (This analysis has been originally tackled in [SN09].)

Equations (6.33) and (6.36) are two examples that illustrate how to derive boundaries on the protocol parameters from Equation (6.29), for a given  $f_{T_{Setup}}$ ,  $\rho$  and  $X$ .

## 6.5. Chapter Summary

In this chapter a hybrid implementation approach has been proposed for the approximate PIA. The VDO and the PIR steps are realized, respectively, by centralized and decentralized signalling mechanisms. In particular, given that connection admission control must be applied on the time scale of connection interarrival times (which implies that CAC algorithms must be fast), we have studied an implementation of the PIR that uses the 3WHS protocol for connection setup. This protocol has recently been proposed in the literature as an alternative (or extension) to GMPLS that guarantees the stringent connection setup latencies envisioned for dynamic optical networks. Unlike GMPLS, upon arrival of a connection request, the 3WHS protocol is able to probe different candidate paths simultaneously, and hence, it provides the network nodes with all the information needed to execute the PIR step. These capabilities render the 3WHS protocol faster than GMPLS, which justifies its suitability to perform online resource allocation with the PIA. Furthermore, the protocol fits well as an implementation alternative for resource allocation algorithms that use the link independence assumption. Under the assumption, to solve the RSA and CAC problems, it is only necessary to know the states of the  $k$ -shortest paths that connect the source/destination nodes. The fast collection of such information is what the protocol performs by its decentralized probing mechanism.

To study the properties of the 3WHS protocol, a task graph reduction methodology was proposed for the analytical evaluation of the protocol latency. By using as input the PDFs of the task processing times, the methodology calculates the PDF of the protocol delay. From this PDF, all statistical properties of the connection setup latency can be determined. The simulations show that the analytical performance model gives precise estimates of the mean connection setup latencies. Therefore, the approach is appropriate for the validation of the protocol design specifications. As an example, it has been shown how the analytical model can be used to quantify the requirements from the task processing times in order to meet a desired performance objective (e.g. a desired upper bound on the connection setup latency).



# Chapter 7

## Modelling Infrastructure Costs in Dynamic Optical Networks

The life-cycle of a telecommunication network is driven by techno-economic evaluations that rely on network planning and cost calculation methods. For a given network, starting from its initial deployment, network planning performs the dimensioning of the network infrastructure. Then the CAPEX and OPEX of this infrastructure are calculated so as to determine the financial feasibility of the network deployment plan. In this chapter, this issue is tackled by outlining a bottom-up framework for the calculation of the CAPEX and OPEX of telecommunication networks. The framework is further applied to the definition of a CAPEX model for dynamic optical networks, with focus on the network infrastructure costs. The goal is to provide insight into the main cost drivers that influence the deployment of this type of networks in the market. Section 7.1 describes the bottom-up framework for CAPEX and OPEX calculation within the context of the network life-cycle. In Section 7.2 the framework is used to formulate a CAPEX model for dynamic optical networks. Section 7.3 presents cost evaluation results. This chapter is based on the cost evaluation models proposed and published by the author in [REB15, RB17b, RAB18].

### 7.1. Bottom-Up Calculation of the Total Cost of the Network Ownership

In Chapter 2 - Section 2.3, a four-step method was outlined for the techno-economic evaluation of telecommunication networks. The method defines a bottom-up approach to dimensioning the network infrastructure and to calculating the corresponding TCO. As explained in Chapter 2 (recall Fig. 2.5), the first step of the method (denoted as input information processing) is responsible for forecasting the service demand over a planning period  $T_i$ . For this predicted demand, the second step (denoted as cost and revenue calculation) applies network planning methods to dimension the infrastructure and to calculate the TCO. The third step calculates the network financial metrics, and the last step assesses the financial feasibility of the network deployment plan. In this chapter, we focus our attention on the first two steps to formulate a bottom-up approach to calculating the network TCO. The approach will then be used to determine the CAPEX for dynamic optical networks.

To start assume that at time  $t_i$  an operator needs to estimate the TCO of a network design that has to serve an expected demand over an interval  $T_i$ . The target design may represent a new network deployment or the upgrade/migration of an existing infrastructure. Forecasting is used to estimate the service demand within  $T_i$ . Having estimated this demand, the network TCO is then calculated as the sum of the total CAPEX and OPEX. This calculation relies on network planning in order to design an infrastructure that minimizes the TCO. To this end, offline optimization algorithms are used for topology planning, demand routing and network dimensioning. These algorithms take into account the existing infrastructure at  $t_i$  and the predicted demands. The purpose is to design a cost-effective network while providing QoS guarantees to the customers. Topology planning involves the definition (or redefinition in the case of an upgrade or migration) of the network nodes and links. Routing and dimensioning determines the resources needed by the network nodes and links, as well as the traffic routing. The resulting design provides a list of materials that specifies the infrastructure to deploy within  $T_i$ . With this information the network CAPEX and OPEX are determined with the bottom-up calculation framework shown in Fig. 7.1.



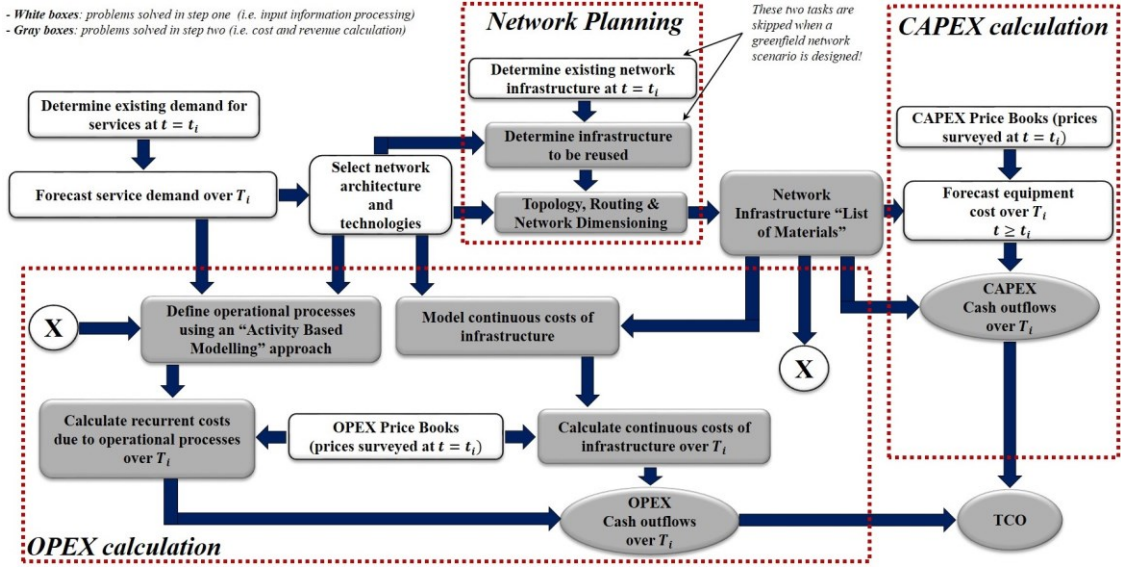


Figure 7.1: Bottom-up framework for TCO calculation [RB17b].

The CAPEX is the cost to buy, upgrade and install the infrastructure, and is a cash outflow stream in  $T_i$ . This cost is calculated from the list of materials provided by the network planning algorithms. For that, the operator needs to define price books with the costs of network components, software, installation, etc. Request for Quotation (RFQ) and Request for Proposals (RFP) can be sent to vendors to obtain this information. The cost depreciation of the network equipment over  $T_i$  has to be regarded in the calculation. For the network design, a CAPEX outflow event at time  $t \in T_i$  may comprise three contributors, and is calculated as [REB15, RB17b]:

$$CAPEX(t) = C_{Inf}(t) + C_{Ins}(t) + C_{ntsi}(t), \quad t \in T_i \quad (7.1)$$

where  $C_{Inf}(t)$  are the network infrastructure costs (which includes network software),  $C_{Ins}(t)$  are the installation costs, and  $C_{ntsi}(t)$  are non-telco specific infrastructure costs (e.g. buildings for personnel). Installation costs are considered as CAPEX, since they are non-recurring charges. Besides, infrastructure costs can be either a one-time expenditure or a cost spread over  $T_i$ . The total CAPEX over  $T_i$  is the sum of the CAPEX cash outflow events in this interval:

$$CAPEX = \sum_{t \in T_i} CAPEX(t) \quad (7.2)$$

As for OPEX calculation, a complete study on the definition, classification and estimation of these expenditures is found in [VPW<sup>+</sup>05]. OPEX are the costs to keep the network operational and are split into continuous costs of infrastructure, costs of operational processes and non-telco specific operational costs. Since OPEX are ongoing costs, it is essential to define before their calculation a time frame, e.g. annual, monthly, over which the expenditures are estimated. Thus, OPEX can be seen as a periodic cash outflow stream within  $T_i$ . The continuous costs of infrastructure referred to in Fig. 7.1, are cash outflows for a network in failure free operation [VPW<sup>+</sup>05]. This expenditure at a time  $t$  in  $T_i$  is given by:

$$O_{CC}(t) = O_{EC}(t) + O_S(t) + O_L(t), \quad t \in T_i \quad (7.3)$$

where  $O_{EC}(t)$  is the cost due to the energy consumption of the network components,  $O_S(t)$  is the cost for leasing space to host network equipment, fibre links, etc., and  $O_L(t)$  is the cost for leasing network infrastructure (e.g. leasing of a copper line in order to provide a transport service). The continuous costs of infrastructure are strongly dependent on the network technologies and on the type of services offered by the operator. Besides these expenditures, we have the costs of operational processes. The network operator has to implement and run processes in order to guarantee full operation and continuity of the business. As seen in [VPW<sup>+</sup>05], these processes are: service provisioning, operational network planning, marketing, maintenance and reparation, as well as pricing and billing. The activity based modelling approach in [CW03, VPW<sup>+</sup>05] can be used to estimate the cost of any process over the predetermined time frame. Therefore, the OPEX cost  $O_{OP}(t)$  due to operational processes is the sum of the costs contributions from each process. An OPEX cash outflow event at time  $t \in T_i$ , is calculated as follows:

Transponder Type	Bit-rate (Gbps)	Reach (Km)
1	40	2500
	100	2000
2	100	2500
	200	2000
	400	500
3	500	2000
	1000	500

**Table 7.1:** Typical technical specifications of flex-grid optical transponders.

$$OPEX(t) = O_{CC}(t) + O_{OP}(t) + O_{ntsi}(t), \quad t \in T_i \quad (7.4)$$

where the total OPEX over  $T_i$  is the sum of the OPEX cash outflow events in this interval:

$$OPEX = \sum_{t \in T_i} OPEX(t) \quad (7.5)$$

$O_{ntsi}(t)$  are non-telco specific ongoing costs (e.g. energy consumption and leasing of premises for personnel). In Equations (7.4)-(7.5) time  $t$  may represent the end of a time frame, e.g. a month, a year. OPEX is not subject to depreciation, and thus, its expected growth must be forecast over  $T_i$ . As with CAPEX, price books for electricity and leasing costs, salaries for personnel, etc., are needed for OPEX evaluation. From Equations (7.2) and (7.5), we have that the TCO of the network design is then calculated as:

$$TCO = \sum_{t \in T_i} (CAPEX(t) + OPEX(t)) \quad (7.6)$$

## 7.2. Calculation of Infrastructure Costs in Dynamic Optical Networks

The novel architectural concepts brought about by dynamic optical networking have a direct influence on the network TCO. In this section we tackle this issue by studying how the three technology drivers of dynamic optical networking (i.e. flex-grid WDM, flexible ROADMs and bandwidth variable and tunable transponders) define the infrastructure cost  $C_{Inf}(t)$  of the CAPEX in Equations (7.1)-(7.2). Thus, in the following, installation costs  $C_{Ins}(t)$  and non-telco specific costs  $C_{ntsi}(t)$  are not regarded in the analysis.

Let us assume that at time  $t_i$  an operator deploys and pays for a network infrastructure that copes with an expected demand over a period  $T_i$ . Since the infrastructure cost is assumed to be a one-time payment incurred at  $t_i$ , we omit the dependence on the time variable  $t$ , and calculate this cost as [REB15, RB17b]:

$$C_{Inf} = C_T + C_{ROADMs} + C_{OLAs} \quad (7.7)$$

where  $C_{Inf}$  comprises three contributors: the costs due to optical transponders  $C_T$ , the ROADM costs  $C_{ROADMs}$ , and the costs of the optical inline amplifiers (OLA)  $C_{OLAs}$ . (The OLAs are installed along the fibre links to compensate the attenuations suffered by the optical signals.) If the network has  $N$  nodes and  $L$  links, the cost contributors can be calculated as:  $C_T = \sum_{n=1}^N C_T(n)$ ,  $C_{ROADMs} = \sum_{n=1}^N C_{ROADM}(n)$  and  $C_{OLAs} = \sum_{l=1}^L C_{OLA}(l)$ , where  $C_T(n)$  and  $C_{ROADM}(n)$  are, respectively, the cost of the transponders and the cost of the ROADM installed at the network node  $n$ . (Recall from Chapter 2 that the transponders are provisioned in shared pools so that they are assigned on demand to connection requests.) Moreover,  $C_{OLA}(l)$  is the cost of the inline amplifiers provisioned on link  $l$ . Equation (7.7) can then be expressed as:

$$C_{Inf} = \sum_{n=1}^N [C_T(n) + C_{ROADM}(n)] + \sum_{l=1}^L C_{OLA}(l) \quad (7.8)$$

To calculate the cost  $C_T(n)$  of the transponders required in the pool at node  $n$ , it is important to note that transponders are available with different specifications such as the supported bit-rates, the required bandwidth and the transparent reach (i.e. the maximum attainable transmission distance without requiring signal regeneration). Table 7.1 presents typical transponder types and their specifications as published in

[RKD<sup>+</sup>13]. These transponders may operate in the C-band (which spans the wavelength range 1530 nm - 1565 nm) and may be tuned at any frequency defined by the flex-grid ITU-T standard G.694.1 [ITU12].

**Example 7.1** In Table 7.1 a transponder of type 2 is a device that can be installed to provide line bit-rates of 100, 200 or 400 Gbps. In all cases, this capacity is provisioned on channels that can be tuned at any carrier frequency defined in the C-band. Note that the transparent reach of the transponder decreases as the offered line bit-rate increases.

In general, the pool of transponders at a node  $n$  may comprise different transponder types. Therefore, if  $m_i$  is the number of transponders of type  $i$  installed in the pool, then  $C_T(n)$  is calculated as:

$$C_T(n) = \sum_i m_i \cdot C_i \quad (7.9)$$

where  $C_i$  is the cost a transponder of type  $i$ . The sum is over all transponder types provisioned in the node. The dimensioning rules used to size the shared pool of transponders determine  $m_i$ .

To calculate the ROADM cost  $C_{ROADM}(n)$ , it is important to select a ROADM design and to perform its dimensioning according to the expected demand. In Chapter 2 - Section 2.2, three ROADM designs were discussed, namely, basic, colorless and colorless & directionless (recall Fig. 2.4). In principle, any of these three variants can be used to design a dynamic optical network, each one provides a solution with a distinct level of flexibility at a given cost. To evaluate that cost, let us determine  $C_{ROADM}(n)$  for each ROADM design. For that, consider Fig. 7.2, which describes the three ROADM types by using as a reference an optical node with degree two (i.e. with two bidirectional links connected to the ROADM). As explained in Chapter 2, the main building blocks of a ROADM are: arrayed waveguide gratings (AWG) - used to implement WDM multiplexers and demultiplexers - passive optical splitters and wavelength selective switches (WSS). In industry the most common WSS variants are WSS(1x9), WSS(1x20) and WSS(9x9), where  $m \times k$  denotes two separate groups of ports. The WSS takes a WDM signal from one of the ports in the group of  $m$  ports and switches each of its constituent channels to ports in the group of  $k$  ports. (The WSS can also switch channels from the group of  $k$  ports to the group of  $m$  ports, i.e. the WSS switching capability is bidirectional [RKD<sup>+</sup>13].) Before studying each design in detail, it is worth mentioning that, for a given ROADM, an add/drop section can be installed for each node degree (i.e. for each bidirectional link attached to the ROADM). This guarantees that the ROADM is able to drop traffic coming from, and add traffic onto, a given ROADM link. This does not imply that pools of transponders are separately provisioned for each node degree. Instead, a single pool suffices and its transponders are connected to their corresponding add/drop sections.

## 7.2.1 Calculation of the Cost of a Basic ROADM

To understand the basic ROADM, let us study the configuration of the components that handle the traffic served by a ROADM link. As seen in Fig. 7.2, each link is bidirectional, and hence, a link consists of an incoming and an outgoing fibre. The incoming fibre of a link carries a WDM signal entering the ROADM. That signal contains connections which may individually be dropped at the node or switched to a desired outgoing fibre. For that, the incoming fibre connects to a pre-amplifier that compensates the attenuation suffered by the signal. Then a splitter directs copies of that signal to the drop section of the link and to WSSs installed at outgoing fibre links. This allows the ROADM to select the traffic to be dropped and switched from the incoming WDM signal. On the other hand, the outgoing fibre of a link is fed by a WSS. This switch is responsible for generating the WDM signal that exits the ROADM through the link. To accomplish this, the WSS receives (from the splitters) copies of the WDM signals that enter the ROADM from other links. It also receives from the add section of the link a WDM signal with connections originating at the node. The WSS then switches to the outgoing fibre the channels (from the incoming signals) that must exit the ROADM. A booster amplifier is placed between the WSS and the fibre to give the signal the appropriate transmission power. The WSSs are of the type WSS(1xk), where  $k$  is the number of input ports of the switch. (Notice that the WSSs only have an output port, which is that connected to the booster amplifier.) With the aid of Fig. 7.2, it is easy to observe that, for a basic ROADM, the number of input ports used at each WSS equals the node degree of the ROADM, i.e. one port is seized by an add section, and one port is seized by each splitter installed at other incoming fibre links.

**Example 7.2** If the basic ROADM in Fig. 7.2 implements switches of type WSS(1x9), only two input ports are used per switch. For instance, the switch connected to link  $A$  assigns one port for the add section of the link and one port to receive the incoming traffic from link  $B$ . This ROADM could be upgraded to support up to 9 bidirectional links without changing the installed WSSs.

In the basic ROADM, the add/drop sections are implemented by AWGs. Today a typical AWG device can be used to install a multiplexer or a demultiplexer with 40 channels capacity [RKD<sup>+</sup>13]. To increase this capacity, an interleaver is used to combine the WDM signals from two multiplexers into a signal with higher channel density (see Fig. 7.2). The multiplexers need to operate at frequencies that do not overlap

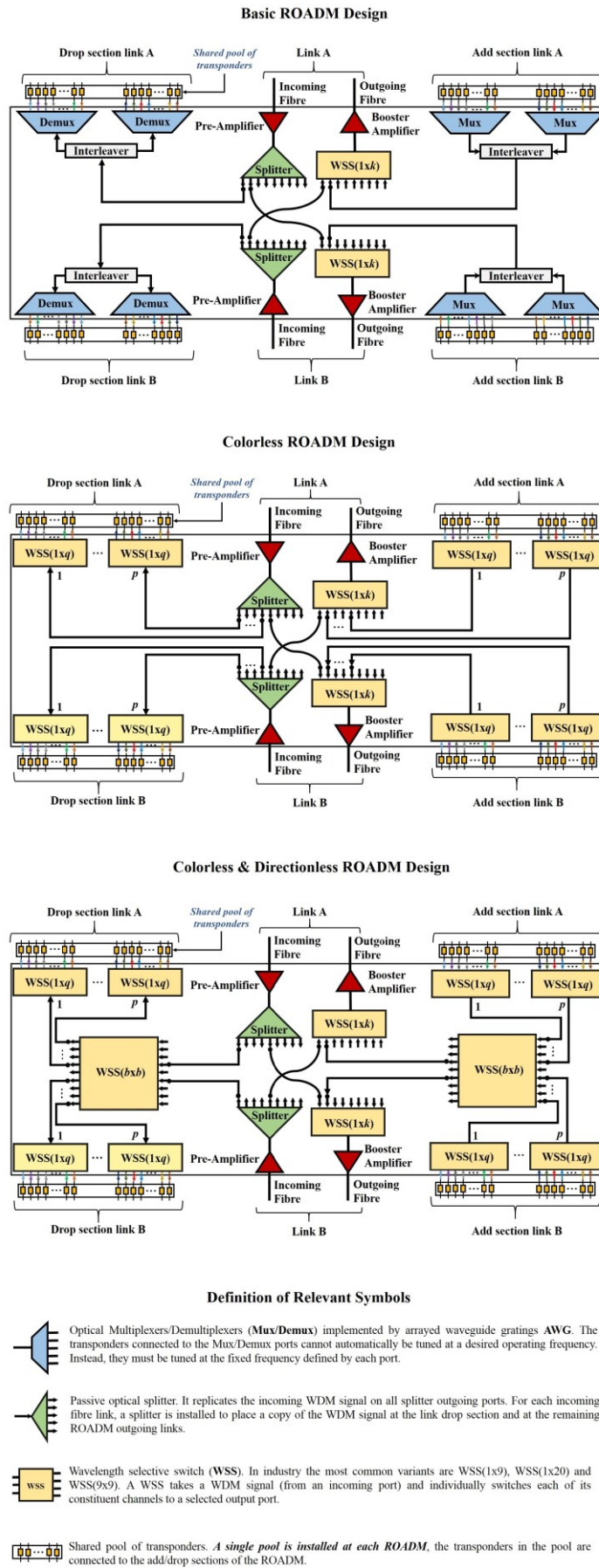


Figure 7.2: Design of ROADM architectures (a two-degree node is shown).

with each other, thereby the interleaver combines both signals without interference. A typical application of this solution is the combination of two WDM signals, each one with 40 channels spaced at 100 GHz. The interleaver creates a new signal with a spectrum grid consisting of 80 channels with 50 GHz spacing. If an interleaver is used at every add section of the ROADM, another one is needed at every drop section for the demultiplexers to be able to separate the WDM channels.

By using Fig. 7.2 as a reference, the cost  $C_{ROADM}(n)$  of a basic ROADM with node degree  $D$  is:

$$C_{ROADM}(n) = D \cdot [C_{WSS(1xk)} + C_{Sp} + C_{Pamp} + C_{Bamp} + 2(i+1) \cdot C_{AWG} + 2i \cdot C_{Int}] \quad (7.10)$$

with  $i = 0,1$ . The term in square brackets contains the costs of the components that support a node degree. These costs are, the cost  $C_{WSS(1xk)}$  of a WSS(1xk) switch, the splitter cost  $C_{Sp}$ , the pre-amplifier cost  $C_{Pamp}$ , the booster amplifier cost  $C_{Bamp}$ , the interleaver cost  $C_{Int}$ , and the cost of a single multiplexer or demultiplexer  $C_{AWG}$ . (The abbreviation AWG is adopted given that multiplexers and demultiplexers are implemented by arrayed waveguide gratings.) The constant  $i$  is introduced to evaluate two different ROADM designs. If  $i = 0$ , the case is considered where each add/drop section has a multiplexer and demultiplexer with capacity of 40 channels, i.e. only two AWG devices are needed without interleavers. Otherwise, if  $i = 1$ , two multiplexers and two demultiplexers are required so as to serve 80 channels per link. In this case two interleavers are needed, one for the add and one for the drop section. The type of WSS to install depends on the node degree  $D$ . If  $2 \leq D \leq 9$ , a WSS(1x9) suffices at each outgoing fibre. Otherwise, if  $10 \leq D \leq 20$ , the variant WSS(1x20) is to be used.

The major drawback of the basic ROADM architecture is that the multiplexers and demultiplexers provide coloured add/drop ports. This implies that the transponders connected to the add/drop sections cannot dynamically tune their frequencies. Instead, they have to be tuned at the frequencies predefined by the ports of the AWG devices. This limitation is overcome by the colorless ROADM design.

## 7.2.2 Calculation of the Cost of a Colorless ROADM

Compared with the basic ROADM, the colorless design (as seen in Fig. 7.2) differs in that the add/drop sections employ WSSs rather than AWG multiplexers and demultiplexers. These WSSs are of the type WSS(1xq), where the number of ports  $q$  determines the add/drop capacity handled by the switch. The per link add/drop capacity is given by the product  $p \times q$ , where  $p$  is the number of switches installed at the add and at the drop sections of the link. Therefore, in a colorless ROADM, the switches of the add sections seize  $p$  input ports from the WSS(1xk) switches that feed the outgoing fibre links. Likewise, the switches in the drop sections seize  $p$  output ports from the splitters connected to the incoming fibres.

**Example 7.3** Assume that the colorless design in Fig. 7.2 uses WSS(1x20) switches to implement the add/drop sections. To support an add/drop capacity of 80 channels (or connections) per link, we have that  $p = 4$ , i.e. four switches need to be installed at both the add and the drop sections of the ROADM links. Moreover, each add section uses four input ports from its corresponding WSS(1xk) switch, whereas each drop section uses four output ports from its link splitter.

By using Fig. 7.2 as a reference, the cost  $C_{ROADM}(n)$  of a colorless ROADM with node degree  $D$  and with per link add/drop capacity  $p \times q$  is given by [RAB18]:

$$C_{ROADM}(n) = D \cdot [C_{WSS(1xk)} + C_{Sp} + C_{Pamp} + C_{Bamp} + 2p \cdot C_{WSS(1xq)}^{add/drop}] \quad (7.11)$$

The term in square brackets contains the costs of the components that support a single node degree. These costs are, the cost  $C_{WSS(1xk)}$  of the WSS(1xk) switch, the splitter cost  $C_{Sp}$ , the pre-amplifier cost  $C_{Pamp}$ , the booster amplifier cost  $C_{Bamp}$ , and the cost  $C_{WSS(1xq)}^{add/drop}$  of the WSS(1xq) switch used to implement the add/drop sections. The constant  $2p$  accounts for the number of WSS(1xq) switches employed to add/drop connections at each ROADM link. In Equation (7.11) the inequality:  $D - 1 + p \leq k$  must be verified. It guarantees that at most the  $k$  input ports of each WSS(1xk) switch are used. Based on Fig. 7.2, we see that, in a WSS(1xk) switch,  $D - 1$  of its input ports are seized by the WDM signals coming from  $D - 1$  splitters, and  $p$  input ports are used by the switches of its add section. Hence, the WSS(1xk) switches must be selected so that they can provide at least  $D - 1 + p$  input ports.

In the colorless ROADM design, the transponders are connected to the ports of the WSS(1xq) switches. Given that these ports are colorless, the transponders can dynamically tune their operating frequencies by software control. This represents an advantage over the basic ROADM design. However, the colorless ROADM constrains the transponders to serve a predetermined network link. For example, in Fig. 7.2, the transponders connected to the add/drop section of link  $A$  are free to adjust their frequencies on demand, but they can only serve traffic carried over that link. If the link fails, the traffic added by the transponders

cannot automatically be re-routed over link  $B$ . This type of deficiency is overcome by using a colorless & directionless design which allows all transponders in the pool to access any ROADM link.

### 7.2.3 Calculation of the Cost of a Colorless & Directionless ROADM

In Fig. 7.2 it is shown how the directionless property can be added to a colorless ROADM. Instead of directly connecting the add/drop sections to the splitters and to the WSS(1xk) switches, two additional switches of the type WSS(bxb) are added to the ROADM. One switch connects all the add sections to all the WSS(1xk) switches installed in the node. The second switch connects all the splitters to all the drop sections. By this approach, both switches guarantee that any transponder can drop/add traffic from/to any network link.

By using Fig. 7.2 as a reference, the cost  $C_{ROADM}(n)$  of a colorless & directionless ROADM with node degree  $D$ , and with per link add/drop capacity  $p \times q$  is given by [RAB18]:

$$C_{ROADM}(n) = D \cdot \left[ C_{WSS(1xk)} + C_{Sp} + C_{Pamp} + C_{Bamp} + 2p \cdot C_{WSS(1xq)}^{\frac{add}{drop}} \right] + 2 \cdot C_{WSS(bxb)} \quad (7.12)$$

This expression is Equation (7.11) plus the cost of the two WSS(bxb) switches, where a single switch has a cost  $C_{WSS(bxb)}$ . Unlike the colorless design, the inequality  $D - 1 + p \leq k$  does not apply as the add sections are not directly connected to the WSS(1xk) switches. Instead, the size of the WSS(1xk) and WSS(bxb) switches must be dimensioned so as to guarantee that any add/drop port can be connected to any node degree. (If this design criterion is verified in practice, the ROADM said to be contentionless [GBS<sup>+</sup>10, RKD<sup>+</sup>13].)

### 7.2.4 Bottom-Up Calculation of the Network Infrastructure Cost

In Table 7.2 and Table 7.3 we summarize the infrastructure cost model for dynamic optical networks as defined by Equations (7.7)-(7.12). The application of the model (to a specific network design) requires the adoption of a ROADM type common to all network nodes. With that, the cost calculation is performed by the following bottom-up methodology [RAB18]:

1. **Dimensioning of the shared pools of transponders and the add/drop capacity:** based on the traffic demand, determine the number and the type of the transponders required at each network node. Besides, define the add/drop capacity (in connections) to be handled by each ROADM.
2. **Calculation of the transponder costs:** for each node  $n$ , calculate the transponder costs  $C_T(n)$  with Equation (7.9). Then calculate the overall transponder cost as  $C_T = \sum_{n=1}^N C_T(n)$ .
3. **Calculation of the ROADM costs:** for each node  $n$ , calculate the ROADM cost  $C_{ROADM}(n)$ . For that, use the add/drop capacity calculated for the ROADM to dimension its add/drop sections, splitters and WSS(1xk) switches. The goal is to define a list of materials that provides the number of components for the ROADM with their size (e.g. the number of input/output ports for each WSS). With the list of materials, the cost  $C_{ROADM}(n)$  is determined by using the equation that corresponds to the selected ROADM design. Finally, the overall ROADM costs are calculated as  $C_{ROADMs} = \sum_{n=1}^N C_{ROADM}(n)$ .
4. **Calculation of the OLA costs:** from the link lengths and the amplification span (i.e. the distance between adjacent OLAs), determine the number  $x_l$  of OLAs required on link  $l$ . Then calculate  $C_{OLA}(l) = x_l \cdot C_{OLA}$ , where  $C_{OLA}$  is the cost of a single amplifier. The overall amplification cost is calculated as  $C_{OLAs} = \sum_{l=1}^L C_{OLA}(l)$ .
5. **Calculation of the network infrastructure cost:** use Equation (7.8) to calculate  $C_{Inf}$ .

It is worth clarifying that in practice different implementation alternatives are valid for the three ROADM designs. In this section we have outlined three possibilities which are plausible with the components (i.e. WSSs, AWGs, splitters, etc.) available today. Our approach is motivated by the cost models proposed in [RKD<sup>+</sup>13] for multilayer transport networks (which includes models for fixed-grid and flex-grid WDM infrastructures). Moreover, in [RKD<sup>+</sup>13] a price book is provided with reference costs for the components used to implement the ROADM designs. Such information will be used in the next section to apply the cost model in Tables 7.2-7.3 to the evaluation of infrastructure costs for selected network scenarios.

Component Cost	Definition
$C_i$	Cost of a transponder of type $i$
$C_{OLA}$	Cost of an optical inline amplifier
$C_{Sp}$	Cost of an optical splitter
$C_{Pamp}$	Cost of a pre-amplifier
$C_{Bamp}$	Cost of a booster amplifier
$C_{AWG}$	Cost of an AWG device (i.e. either a multiplexer or a demultiplexer)
$C_{Int}$	Cost of an interleaver
$C_{WSS(1xk)}$	Cost of a WSS(1xk) switch used to feed outgoing fibre links
$\frac{add}{drop}$ $C_{WSS(1xq)}$	Cost of a WSS(1xq) switch used to implement add/drop sections
$C_{WSS(bxb)}$	Cost of a WSS(bxb) switch used to implement directionless ROADMs

**Table 7.2:** Definition of relevant component costs for dynamic optical networks.

Cost Type	Cost Calculation
Infrastructure cost for a network with $N$ nodes and $L$ bidirectional links	$C_{Inf} = \sum_{n=1}^N [C_T(n) + C_{ROADM}(n)] + \sum_{l=1}^L C_{OLA}(l)$
Cost of the transponders installed at node $n$	$C_T(n) = \sum_i m_i \cdot C_i$ . Total number of transponders in the pool: $\sum_i m_i$ , where $m_i$ transponders of type $i$ are used
Basic ROADM with degree $D$ , with per link add/drop capacity of 40 ( $i = 0$ ) or 80 channels ( $i = 1$ )	$C_{ROADM}(n) = D \cdot [C_{WSS(1xk)} + C_{Sp} + C_{Pamp} + C_{Bamp} + 2(i+1) \cdot C_{AWG} + 2i \cdot C_{Int}]$
Colorless ROADM with degree $D$ , with per link add/drop capacity $p \times q$	$C_{ROADM}(n) = D \cdot [C_{WSS(1xk)} + C_{Sp} + C_{Pamp} + C_{Bamp} + 2p \cdot C_{WSS(1xq)}^{add/drop}]$
Colorless & directionless ROADM with degree $D$ , with per link add/drop capacity $p \times q$	$C_{ROADM}(n) = D \cdot [C_{WSS(1xk)} + C_{Sp} + C_{Pamp} + C_{Bamp} + 2p \cdot C_{WSS(1xq)}^{add/drop} + 2 \cdot C_{WSS(bxb)}]$

**Table 7.3:** Calculation of infrastructure costs for dynamic optical networks.

### 7.3. Evaluation of Network Infrastructure Costs

In this section we assess the infrastructure costs for the partial-mesh and the ring networks in Fig. 7.3. Both networks consist of 10 nodes connected by links with capacities of  $C_l = 32$  slots, and define 45 node-pairs, each one serving two connection classes: a narrowband and a wideband class that request, respectively,  $b_j = 2$  and  $b_j = 8$  adjacent slots. Sets of  $k = 2$  shortest paths are defined (for every node-pair) to route connections. The performance of these networks was evaluated in Chapter 5 for different traffic loads  $A$ . (The results are summarized in Section 5.3, Figures 5.7-5.10 and Tables 5.13-5.15.) In this section, for each network and traffic load  $A$ , we perform the dimensioning and the cost evaluation of the infrastructure that provides the performance assessed in Chapter 5. Three solutions are assessed for each network, which are based on the implementation of basic, colorless and colorless & directionless ROADMs.

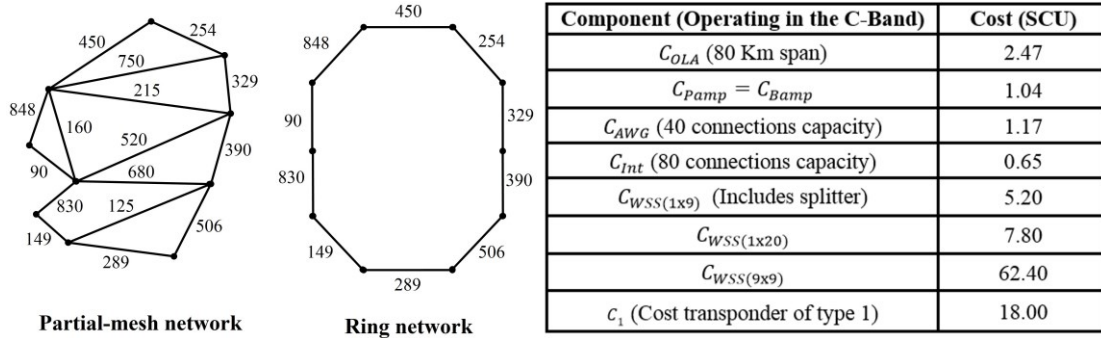


Figure 7.3: Network topologies with link lengths in Km and definition of component costs.

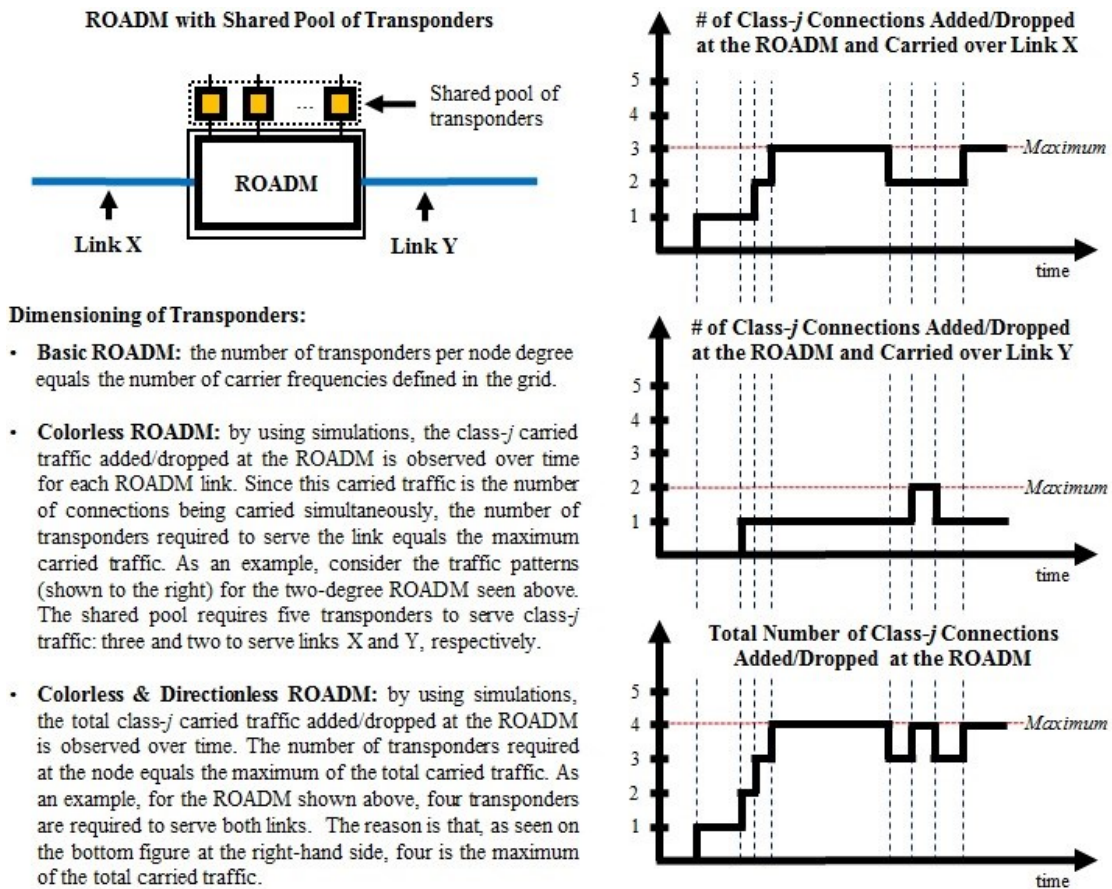


Figure 7.4: Dimensioning of the number of transponders to install at a network node.

To calculate the infrastructure costs, we follow the methodology outlined in Section 7.2.4 that uses the cost model in Tables 7.2-7.3. As price book for the cost model, we use the component costs defined in [RKD<sup>+</sup>13] for booster/pre-amplifiers, transponders, interleavers, OLAs, WSSs and AWGs. Such costs are shown in Fig. 7.3 and are expressed in strongest cost units (SCU). As defined in [RKD<sup>+</sup>13], an SCU is the cost of a 10 Gbps transponder with transparent reach of 750 km in the year 2012.

### 7.3.1 Network Infrastructure Dimensioning

In general, for any given network, the dimensioning of the shared pools of transponders and the add/drop capacity depends on the ROADM type used. For a basic ROADM, since the multiplexers/demultiplexers provide coloured ports, then for each ROADM link, the number of transponders is equal to the number of carrier frequencies in the spectrum grid. This number is 32 for the partial-mesh and the ring networks in Fig. 7.3, and it determines the add/drop capacity per link. Therefore, a node  $n$  with degree  $D$  requires  $32 \times D$  transponders. To illustrate this, consider the two-degree ROADM in Fig. 7.4. In a basic design, such a node requires 64 transponders (32 per link). In order to provide this capacity, each link add/drop section



is implemented by AWG multiplexers/demultiplexers with capacity of 40 channels (eight ports are free at each AWG device). The cost of this component is listed in Fig. 7.3. Thus, to implement the partial-mesh and the ring networks with basic ROADMs, there is no need for interleavers as the capacity of a single AWG supports the grid with 32 slots. By this design, every ROADM add/drop section provides the carrier frequencies that may be required by a connection request.

The dimensioning of transponders differs in colorless and colorless & directionless ROADM designs. For these two ROADM types, the transponders connected to the add/drop ports may re-tune their carrier frequencies on demand, and therefore, the number of transponders (per ROADM link) needs not equal the grid size. Instead, it suffices to install the transponders required to cope with the actual carried traffic. To achieve this, we use a simulation-based method for transponder dimensioning similar to that in [SW10, CCC<sup>+</sup>12, SCG<sup>+</sup>12]. The idea is to determine the number of transponders from the carried traffics observed in the simulations run in Chapter 5 to assess the performance of both networks. (Recall that, as mentioned in Section 5.3, the simulations were run under the assumption that all network nodes had enough resources to serve connections.) For a given network and offered traffic load  $A$ , the method works as follows. In a colorless ROADM, the number of transponders needed to add/drop class- $j$  connections over a ROADM link equals the maximum class- $j$  carried traffic (observed over time) which is added/dropped over the link. As an example, consider the two-degree ROADM in Fig. 7.4, which is depicted with its per link carried traffics. It is seen that according to the traffic carried over link X, three transponders suffice at the add/drop section of the link. Similarly, two transponders are needed for link Y. Thus, the ROADM (in a colorless configuration) would require five transponders to add/drop class- $j$  connections. (Recall that the traffic carried at a given instant is the number of connections being carried simultaneously. Therefore, the number of transponders - for each link add/drop section - equals the maximum number of concurrent connections observed over time.) For a colorless & directionless ROADM the dimensioning method varies. In this case (unlike the basic and colorless designs) the transponders are not limited to serve a specific ROADM link. Therefore, it suffices to observe the overall class- $j$  carried traffic added/dropped at the ROADM, thereby the number of transponders equals the maximum overall carried traffic. (Note that the overall traffic is the sum of the carried traffics added/dropped over all ROADM links.) For the ROADM in Fig. 7.4, the overall class- $j$  carried traffic (shown at the bottom of the figure) is the sum of the class- $j$  traffic added/dropped over links X and Y. In a colorless & directionless configuration, such a ROADM would only require four transponders to add/drop class- $j$  traffic (i.e. by adding the directionless property to a colorless ROADM, the number of transponders is reduced.) The proposed simulation-based dimensioning method tailors (for a selected ROADM design) the number of transponders to an expected traffic load  $A$ . This method cannot be applied to basic ROADMs because (regardless of the traffic) the add/drop sections need a transponder for each carrier frequency in the grid.

For the partial-mesh and the ring networks, the add/drop sections of the colorless and the colorless & directionless designs are implemented by WSS(1x $q$ ) switches, which (according to the typical variants available on the market) can be of type WSS(1x9) or WSS(1x20). From the cost model in Tables 7.2-7.3, each add/drop section in a ROADM has capacity  $p \times q$  connections. Based on this, we use the following rule for the dimensioning of the add/drop capacity of a network ROADM. Let  $k$  be the number of transponders (as determined by the simulation-based method) required to serve the ROADM link that handles the highest (add/drop) carried traffic across all ROADM links. Then the add/drop sections of the ROADM are implemented by the minimum number  $p$  of switches of type WSS(1x $q$ ) such that  $k \leq p \times q$ , where  $q = \underset{q \in \{9,20\}}{\operatorname{argmin}} [p \cdot C_{WSS(1xq)}]$  is the switch type that yields the minimum cost. As an example, for the ROADM in Fig. 7.4, we have that  $k = 3$  (as link X carries the highest traffic load). Then the minimum cost is obtained if  $p = 1$  and  $q = 9$ , i.e. a WSS(1x9) switch is used at each add and drop section at a cost of  $C_{WSS(1x9)} = 5.20$  (SCU) - see price book in Fig. 7.3. This dimensioning rule guarantees that, for a given ROADM, all links have the same add/drop capacity at minimum cost. The results obtained by this rule are applicable to both the colorless and the colorless & directionless versions of the ROADM (as seen in the previous section, these designs, for a given ROADM, have the same add/drop sections). Depending on the calculated switch type, in the cost model in Tables 7.2-7.3, we have that  $C_{WSS(1xq)}^{add/drop}$  can be equal to either  $C_{WSS(1x9)}$  or  $C_{WSS(1x20)}$ . These component costs are defined in Fig. 7.3.

Two additional comments are needed regarding the dimensioning of the ROADMs. First, for the three ROADM designs, the WSS(1x $k$ ) switches are of type WSS(1x9), given that the nodes of the partial-mesh and the ring networks have degrees  $D < 9$ . Therefore, in the cost model in Tables 7.2-7.3, we use  $C_{WSS(1xk)} = C_{WSS(1x9)}$ . This cost (see Fig. 7.3) includes the cost  $C_{Sp}$  of the splitter used by the incoming fibre of the link served by the switch. Secondly, in the colorless & directionless design, the WSS( $b \times b$ ) switches are of type WSS(9x9), as they are available in the market and fulfil the capacity requirements for the networks in Fig. 7.3.

The relationship between the class- $j$  bandwidth  $b_j$  (expressed in slots) and the connection bit-rate  $R_j$  (in Gbps) is defined as  $b_j = 2 \cdot \lceil (R_j/I + G)/12.5 \rceil$ , where  $I$  is the spectral efficiency (in Gbps/GHz) of the modulation format applied by the transponders, and  $G$  is the guard band (in GHz) [KRV<sup>+</sup>13, REB15,

Partial-Mesh Network		Ring Network	
A (Erl)	$R_L$ (%)	A (Erl)	$R_L$ (%)
1.7	0.001	0.6	0.001
2.2	0.004	0.8	0.009
2.8	0.012	1.1	0.028
3.3	0.018	1.4	0.068
3.9	0.060	1.7	0.159
4.4	0.102	1.9	0.277
5.0	0.172	2.2	0.510

**Table 7.4:** Summary PIA performance with the MDP-PGMC rule (first-fist spectrum allocation).

RB17b]. The constant  $12.5 = 2 \times 6.25$  has units of GHz and it accounts for the fact that, in a flex-grid network, the bandwidth  $b_j$  consists of an integer number of spectrum blocks with 12.5 GHz bandwidth. Each block is built by two 6.25 GHz adjacent slots [ITU12]. By using  $I = 2$  Gbps/GHz, which is a typical modulation efficiency envisioned for flex-grid systems [Sal07], and by setting  $G = 0$  GHz, we have that  $R_j = 20$  Gbps for narrowband ( $b_j = 2$ ) and  $R_j = 100$  Gbps for broadband ( $b_j = 8$ ) connections. Therefore, from the transponder types listed in Table 7.1, the transponder of type 1 suffices to serve both connection classes in the partial-mesh and the ring networks. In particular, to serve a narrowband connection, the transponder is used to offer the bit rate of 20 Gbps, assuming a transparent reach of 2500 Km. For a broadband connection, a bit-rate of 100 Gbps is used, thereby attaining a 2000 Km reach (see Table 7.1).

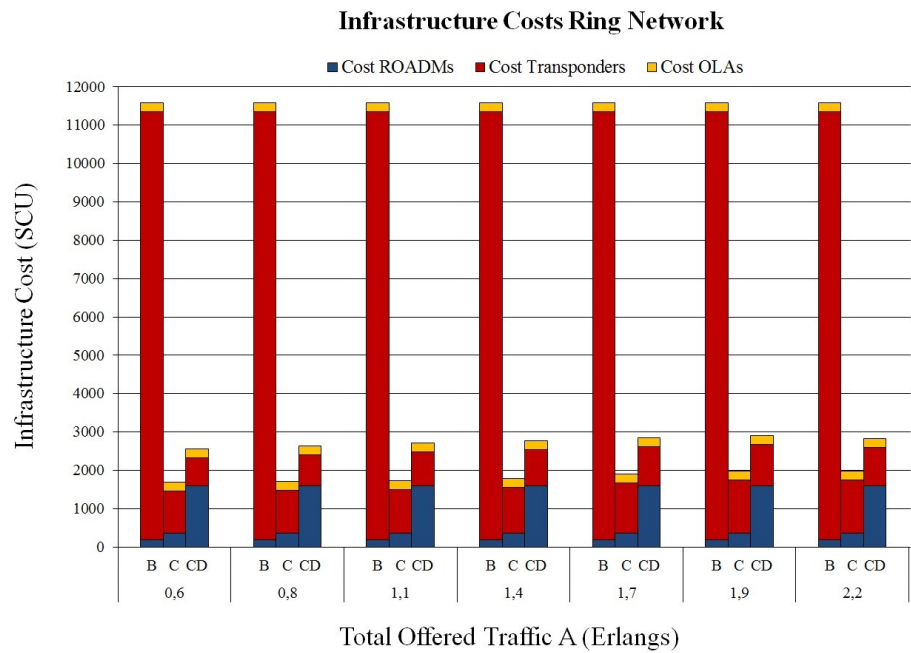
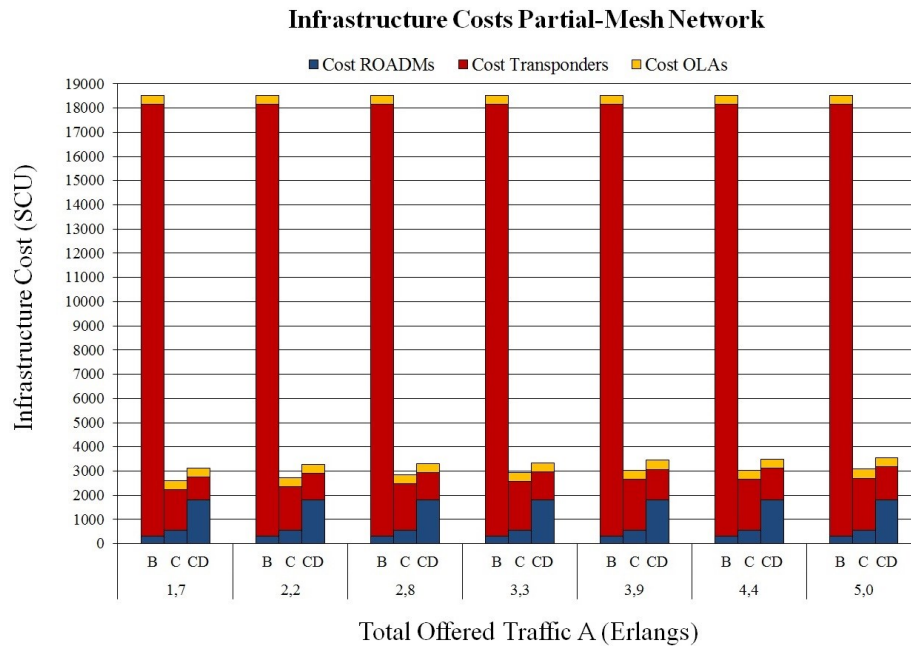
In both networks, regenerators are installed (if needed) for connections that use paths with lengths that exceed the transparent reach of the transponders. A regenerator is implemented as a pair of transponders (back to back connected) installed at the pool of transponders of a network node. Thus, if a regenerator is required, it has a cost of  $2 \times 18 = 36$  (SCU), where 18 is the cost a type 1 transponder (see Fig. 7.3). The regenerator costs are included in the transponder costs  $C_T$ . As for inline optical amplification, according to the OLA specifications in Fig. 7.3, every 80 Km an amplifier is installed on each fibre link.

### 7.3.2 Numerical Results

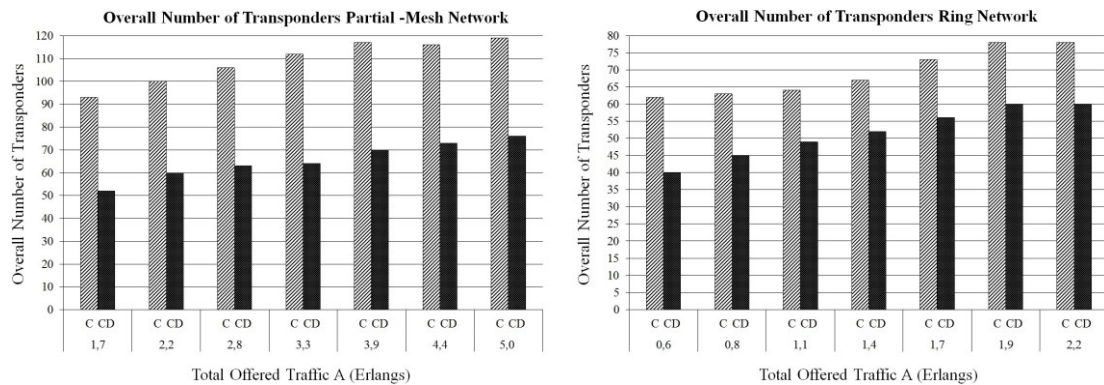
In this section the acronyms B, C and CD are used in figures and tables to refer to the basic (B), colorless (C) and colorless & directionless (CD) solutions. Moreover, the results are presented for the traffic loads  $A$  considered in Chapter 5 for the partial-mesh and the ring networks. In particular, for a network serving a traffic  $A$ , the infrastructure is designed for the three ROADM types on condition that the three solutions have the same performance. Such a performance (i.e. the reward losses  $R_L$  assessed in Chapter 5 - Section 5.3) and the corresponding traffic loads  $A$  are summarized in Table 7.4. The reward losses are those observed (in both networks) when the PIA is run with the MDP-PGMC rule using first-fit as spectrum allocation method. (The information in Table 7.4 has been taken from Tables 5.14-5.15 in Chapter 5.)

Figure 7.5 shows the infrastructure costs calculated for the partial-mesh and the ring networks. The costs are split into their three components, namely,  $C_T$ ,  $C_{ROADMs}$  and  $C_{OLAs}$ . As seen, for both networks in all traffic loads, the solutions based on basic ROADMs are the most costly, whereas the lowest costs are provided by the colorless solutions. The higher costs of the basic solutions stem from the large number of transponders required to provide connectivity at all frequencies in the grid. This evinces that implementing coloured add/drop sections renders the basic solution prohibitively expensive. On the other hand, Fig. 7.5 shows that, in both networks, the infrastructure costs for the colorless and the colorless & directionless solutions depend on (and therefore, vary with) the traffic load. For the considered traffic loads, the cost component more sensitive to traffic variations is the overall transponder costs  $C_T$  (see Fig. 7.5). These costs are driven by the number of transponders installed in the network. This number, as shown in Fig. 7.6, grows with the traffic load  $A$ .

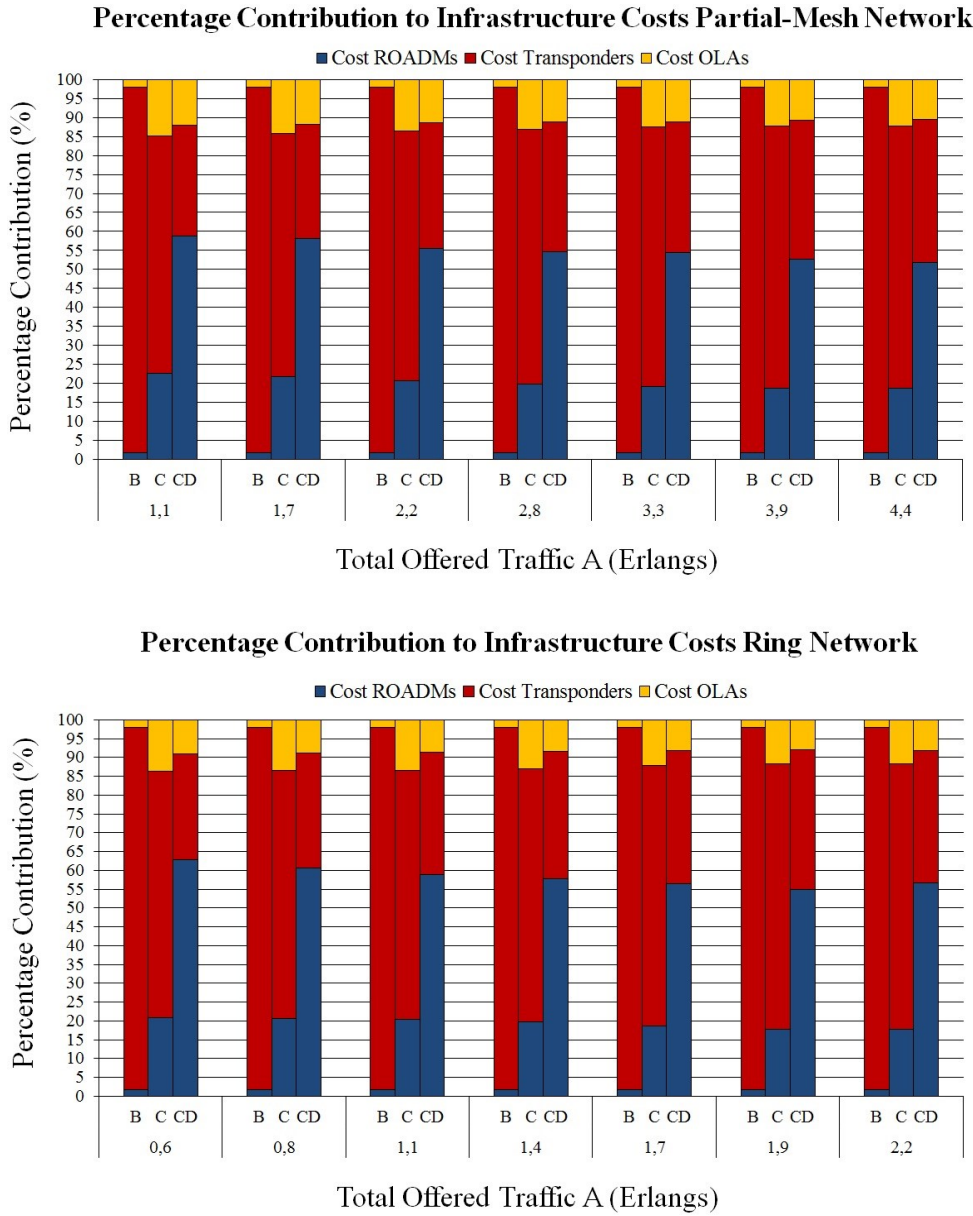
Based on Fig 7.6, we have that on average, with respect to the colorless & directionless solutions, there is an increment in the colorless designs of 70% and of 35% in the number of transponders for the partial-mesh and the ring networks, respectively. This shows that the directionless property substantially reduces the size of the transponder pools, and therefore, the infrastructure cost component  $C_T$  (note that in Fig. 7.5 the transponder costs in the colorless & directionless solutions are less than in the colorless case). By using directionless ROADMs, the transponders are better utilized as idle transponders are always available for connections routed over any ROADM link. In contrast, in the colorless solutions, idle transponders can only serve connections over a fixed ROADM link. This constraint increments the number of transponders required to provide the same performance as the directionless-based designs. Another disadvantage of this



**Figure 7.5:** Infrastructure costs vs traffic load for the three ROADM designs.



**Figure 7.6:** Transponders required by the colorless and the colorless & directionless solutions.



**Figure 7.7:** Percentage contribution of ROADMs, transponders and OLAs to the overall cost.

constraint is that in case of link failures, the colorless ROADMs cannot automatically switch the add/drop traffic onto another ROADM link. This limitation is easily overcome by the directionless designs as by software control, the affected traffic can be re-routed to selected back-up paths. Therefore, although more costly than the colorless solutions, the colorless & directionless designs are more suitable to cope with the stringent requirements of dynamic optical networks. More specifically, the infrastructure costs for the colorless & directionless solutions in the partial-mesh network (as seen in Fig. 7.5) are on average 17% higher than the colorless approaches. For the ring network, the cost increment is on average 50%. The colorless & directionless solutions are more expensive due to the WSS(9x9) switches that are needed to add the directionless capability.

In Fig. 7.7 the percentage contributions to the infrastructure costs of the cost components  $C_T$ ,  $C_{ROADMs}$  and  $C_{OLAs}$  are shown. As seen, in the basic and colorless solutions the main cost driver (or contributor) is the overall cost of the transponders. This cost represents more than 95% and 60% of the infrastructure cost in the basic and colorless designs, respectively. In contrast, in the colorless & directionless solutions, the ROADM costs are the major contributors, accounting for more than 50% of the total cost. In all three solutions, the amplification costs  $C_{OLAs}$  represent the minimum cost contributor. It is important to note that the aforementioned percentage contributions tend to be similar in the partial-mesh and the ring networks in all considered traffic loads  $A$ .

## 7.4. Chapter Summary

The cost evaluations of this chapter have shown that to serve a traffic demand (while guaranteeing a desired performance) the implementation of colorless ROADMs provides the lowest infrastructure costs. However, this solution suffers a severe technical limitation such as the inability to automatically re-route connections in case of link failures. This deficiency can be overcome by adding the directionless property to colorless ROADMs. Although the resulting solution is more costly, it allows more efficient utilization of the network resources. More specifically, a colorless & directionless architecture, substantially reduces the number of transponders required to serve an expected traffic demand, which represents a remarkable advantage over the basic and the purely colorless-based approaches.

Regardless of the ROADM type, the results show that the transponders represent a relevant cost driver, especially in the case of basic and colorless designs. This cost is reduced by the directionless ROADM designs in both the partial-mesh and the ring topologies considered in the study.



# Chapter 8

## Conclusion

The emergence of grid computing and cloud-based services is imposing new connectivity requirements on the optical layer. This has raised awareness on the need for a changeover to dynamic optical networks. Instead of providing long-term installed constant bit-rate connectivity, these networks are envisioned to serve connections with different bit-rate requirements, with random interarrival and holding times, and with low connection setup latencies. To efficiently utilize their capacity, dynamic optical networks require algorithms for online resource allocation that are fast, adaptive and state-dependent. In this thesis, this issue has been addressed by approaching online resource allocation as a reward-based Markov decision process. By modelling the network as a large-scale stochastic system, it has been shown that the decision process can be implemented by an approximate PIA. The purpose of the PIA is to calculate a resource allocation policy that aims at maximizing the reward earned by the network. Compared to existing online RSA algorithms, the PIA has two relevant advantages: first, any desired performance objective can be defined by properly setting the connection reward parameters; secondly, adaptive GoS control can be performed on individual connection classes, thereby the PIA is not limited to the minimization of overall blocking (as it is the case of existing online RSA algorithms). These PIA capabilities are validated by the performance evaluation results for the selected full-mesh, partial-mesh and ring network topologies.

The adoption of a reward-based MDP approach is advantageous as it provides a mechanism to estimate the impact that carried connections cause in the network. This impact is state, policy and traffic load dependent, and can be quantified by the long-term reward gains. The purpose of the approximate PIA is to assign connection requests to lightpaths that cause positive long-term reward gains. For most networks, the exact estimation of these gains is unfeasible given the huge cardinality of the network state-space. It has been shown, however, that this limitation is circumvented by the approximate PIA, which resorts to the link independence assumption, and to a macrostate-dependent estimation of the gains. From the considered performance evaluation scenarios, it can be argued that although the link independence assumption is not always accurate, the performance of the approximate PIA can be improved by defining simple admission decision strategies, such as the MDP-SP, the MDP-PG and the MDP-PGMC rules. The performance of these rules depends on the spectrum allocation algorithm used to calculate lightpaths. In particular, it was observed that, compared to random-fit, the rules perform better when first-fit is used as the spectrum allocation method. In fact, with first-fit, the MDP rules admit connections on lightpaths that tend to reduce the spectrum fragmentation, thereby leading to less blocking in the network. The best performance is attained by the PIA variant that uses the MDP-PGMC rule with first-fit as spectrum allocation method. It is worth noting that the differences in the PIA performance for first-fit and random-fit prove that the reward gains need not be alike for lightpaths routed on the same path.

From the performance evaluation results, it can be concluded that an exact description of the configuration of the optical grid is not always necessary to perform adaptive/state-dependent resource allocation. Instead, the stochastic properties of the network can be described by the macrostate-dependent approach used by the approximate PIA. This method not only reduces computational complexity, but provides a mechanism to detect and reject adverse connection requests that, despite the availability of resources, would prevent admission of more valuable connections. The proposed PIA extends the applicability of MDP theory to the control of stochastic networks subject to contiguity and continuity constraints. Besides, it demonstrates the benefits of modelling resource allocation as a decision process that is adaptive to changing traffic load conditions.

In dynamic optical networks connections can be rapidly set up and torn down. Ongoing research has shown that in some cases connections might require setup times in the range of milliseconds to seconds. To meet these stringent requirements, signalling protocols are needed to enable low setup latencies without manual intervention. Given that resource allocation is part of the connection setup process, the design of resource allocation algorithms must consider the connection setup signalling procedures. This issue has

been tackled by investigating an implementation of the approximate PIA with a 3WHS protocol which is known for providing very fast connection setup. The proposed implementation is enabled by the link independence assumption which assumes that a connection only modifies the reward rates of the links that it uses. The implications of this assumption, from a performance perspective, were discussed before (i.e. it reduces computational complexity while introducing - in some cases - inaccuracies in the estimation of the reward gains which are mitigated by MDP-based admission decision rules). From an implementation point of view, the assumption has another implication: it substantially reduces the signalling workload in the network. The reason is that upon arrival of a connection request, it is only necessary to know the state of the  $k$ -shortest paths that connect the source/destination nodes. The collection of the actual states of these paths is easily accomplished by the probing mechanism of the 3WHS protocol. (Note that this probing mechanism, which simultaneously collects information from  $k$ -shortest paths, is not enabled by current GMPLS connection setup procedures.) Therefore, the link independence assumption, as embraced by the approximate PIA, reduces the information (or equivalently, the exchange of signalling messages) needed to perform online resource allocation. It is worth noting that for the exact PIA (which is computationally intractable for large-size networks), neither GMPLS nor the 3WHS protocol are suitable to provide fast connection setup. The reason is that to establish a connection, the exact PIA must know the actual network state. The collection of such information in real time may cause large signalling delays that might preclude fast connection setup. This statement is true for any online resource allocation algorithm that needs global information for the provisioning of resources. The proposed implementation shows that the approximate PIA can be used with the 3WHS protocol so as to attain low connection setup latencies.

To provide a thorough analysis of the connection setup latency, an analytical method was proposed to evaluate the performance of the 3WHS protocol. In the method, the protocol is modelled as a task graph that represents the signalling delays incurred in the connection establishment phase. By using reduction techniques, the graph is simplified so as to obtain a performance model that estimates the mean connection setup latency. The results show that the model provides accurate estimates of the protocol performance. For the proposed PIA implementation, it has been found that, regardless of the network size, the cross-connection times of the ROADMs are relevant contributors to the setup latency. For large-size networks, on the other hand, the connection setup latencies are mainly driven by the propagation delays of the signalling messages on the  $k$ -shortest paths. The advantage of the analytical method is that it can be extended to study the latencies incurred by any connection setup protocol and resource allocation algorithm.

Network planning plays a predominant role in the deployment of dynamic flex-grid optical networks. It guarantees that sufficient infrastructure is installed to cope with an expected traffic demand. The cost study performed on selected network scenarios indicates that, although not the least costly, colorless & directionless network infrastructures are the best solution alternatives in terms of flexibility. More specifically, from the study, it was observed that by adding the directionless property to colorless ROADMs, the number of transponders in the network is substantially reduced (compared to purely colorless-based solutions). Furthermore, this property allows the transponders installed at a given network ROADM to add/drop connections over any ROADM link. With this feature, the network is able to automatically re-route traffic by software control, while providing quick recovery from link failures. These capabilities justify the additional costs incurred by the directionless property w.r.t. colorless-based network designs.

The research contributions in this thesis can be further extended in different directions. First, regarding the approximate PIA, an open research question is how to tune the connection reward parameters (during network operation) to either provide a desired GoS or to perform equalization of the class-specific blocking probabilities. Also it would be interesting to investigate different solution methods for the systems of linear equations defined by the approximate PIA. In this thesis, an existing VIA has been used, however, it is necessary to study in more detail the performance of this and other solution methods for different network scenarios (e.g. very-large capacity networks, large number of connection classes, more stringent connection setup latencies, etc.). Regarding the implementation of the approximate PIA, the solution based on the 3WHS protocol is a plausible alternative, but not the only one. This motivates further research on SDN-based implementation scenarios. In this case, an important issue to address is the definition of the control plane procedures and the signalling messages that would support the execution of the PIA. In particular, the signalling mechanisms need to guarantee that (like the 3WHS protocol) fast connection establishment is provided when needed. For this analysis, the task graph reduction methodology can be used to determine the statistical properties of the connection setup latencies.

In dynamic optical networks the arrival and departure of connections renders the spectral resources prone to reach a fragmentation state. This means that, on each link, the spectrum contains idle and non-contiguous slots which cannot be allocated to future connection requests. The increment in the connection blocking probability is a result of a fragmented spectrum. From the perspective of the PIA, it would be interesting to investigate admission decision rules to detect and reject connection requests that can increase the spectrum fragmentation. For this, it is necessary to study the relationship between the long-term reward gains and the fragmentation state of the network. (A highly fragmented spectrum is expected to be defined



by link states with negative gains, this effect can easily be detected by the PIA.) Besides, the reward gains could be used as a decision threshold to trigger a spectrum defragmentation procedure in the network. For example, negative reward gains suggest that the blocking is high (which has a direct correlation with the fragmentation state of the spectrum). Therefore, reference values for the reward gains can be defined to decide when to re-accommodate carried connections so as to reach network states with higher gains.

Another open research challenge is the study of the OPEX in dynamic optical networks. The CAPEX analysis presented in this thesis has shown that a colorless & directionless infrastructure brings flexibility and configurability by software control. This flexibility has an impact on the operational processes defined by the operator to keep its network up and running. More specifically, maintenance and reparation, service provisioning and operational network planning are processes that must be redefined in a dynamic scenario. Therefore, it is a must to investigate how these processes change and which are their corresponding cost contributions to the network TCO. This can be accomplished by applying the bottom-up cost calculation framework defined in this thesis.



# List of Acronyms

AWG	Arrayed Waveguide Grating
BoD	Bandwidth on Demand
CAC	Connection Admission Control
CAPEX	Capital Expenditures
CDF	Cumulative Distribution Function
CPE	Customer Premises Equipment
CSP	Communication Service Provider
DEMUX	Demultiplexer
DWDM	Dense Wavelength Division Multiplexing
FF	First-Fit
FXC	Fibre Crossconnect
GMPLS	Generalized Multiprotocol Label Switching
GoS	Grade of Service
ILP	Integer Linear Programming
IP/MPLS	Internet Protocol/Multiprotocol Label Switching
IRR	Internal Rate of Return
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
KSP	K-Shortest Paths
MDP	Markov Decision Process
MDP-PG	Markov Decision Process-Positive Gain
MDP-PGMC	Markov Decision Process-Positive-Gain-Maximum Capacity
MDP-SP	Markov Decision Process-Shortest Path
MILP	Mixed Integer Linear Programming
MUX	Multiplexer
NPV	Net Present Value
OFDM	Orthogonal Frequency Division Multiplexing
OLA	Optical Inline Amplifier
OPEX	Operational Expenditures
OTN	Optical Transport Network
PB	Payback Period
PCE	Path Computation Element
PDF	Probability Density Function
PIA	Policy Iteration Algorithm
PIR	Policy Improvement Routine
POP	Point of Presence
QoS	Quality of Service
RF	Random-Fit
ROADM	Reconfigurable Optical Add/Drop Multiplexer
ROI	Return on Investment
RSA	Routing and Spectrum Allocation
RWA	Routing and Wavelength Assignment
SA	Spectrum Allocation
SDH	Synchronous Digital Hierarchy
SDN	Software Defined Networking
SLA	Service Level Agreement
SONET	Synchronous Optical Network
TCO	Total Cost of Ownership
TDM	Time Division Multiplexing
VDO	Value Determination Operation

VIA	Value Iteration Algorithm
WDM	Wavelength Division Multiplexing
WSS	Wavelength Selective Switch
3WHS	3-Way Handshake Signalling Protocol

# List of Figures

Figure 2.1: Transport services and technology layers that define the core network. ....	7
Figure 2.2: Dynamic optical network architecture. ....	8
Figure 2.3: Comparison between fixed-grid and flex-grid WDM. ....	9
Figure 2.4: Typical ROADM configurations (a two-degree node is depicted as an example). ....	10
Figure 2.5: Techno-economic analysis for network deployment planning [RB17b]. ....	11
Figure 2.6: The network life-cycle as driven by techno-economic evaluations. ....	12
Figure 3.1: Example of a flex-grid optical network serving 12 classes. ....	19
Figure 3.2: Example of a flex-grid optical network that serves two connection classes. ....	20
Figure 3.3: Possible macrostates (carried traffics) for the network in Fig. 3.2. ....	21
Figure 3.4: Example of two possible resource allocation policies for the network in Fig. 3.2. ....	22
Figure 3.5: Resource allocation for class-1 and class-2 arrivals when two distinct policies are used. ....	22
Figure 3.6: Possible decisions for a class-1 request arriving in state $x_2$ . ....	24
Figure 3.7: Possible decisions when either a class-1 or a class-2 arrival occurs in state $x_6$ . ....	24
Figure 3.8: Probabilistic trees to calculate the rewards $V(x, \Pi, n)$ . ....	27
Figure 3.9: Expected reward $V(x, \Pi, n)$ earned by a network until time $t \gg t_0$ after $n$ state transitions. ....	29
Figure 3.10: Calculation of the network reward rates $R(\Pi_1)$ and $R(\Pi_2)$ in Example 3.11. ....	31
Figure 3.11: Linear system for the network in Fig. 3.2 when policy $\Pi_1$ is used. ....	33
Figure 3.12: Flow chart for the policy iteration algorithm (PIA). ....	34
Figure 3.13: Temporal diagram for the policy iteration algorithm (PIA). ....	35
Figure 3.14: Lightpaths defined by the decisions in Table 3.5. ....	38
Figure 4.1: Flex-grid optical network with four nodes, five links and 12 classes [RB16d]. ....	44
Figure 4.2: Flex-grid optical link. ....	45
Figure 4.3: State and macrostate spaces for a single-link network that serves two connection classes. ....	48
Figure 4.4: System of linear equations in the state-dependent model for the link in Fig. 4.3. ....	50
Figure 4.5: System of linear equations in the macrostate-dependent model for the link in Fig. 4.3. ....	51
Figure 4.6: Definition of the policy and macrostate-dependent arrival rates for the link in Fig. 4.3. ....	58
Figure 4.7: Approximate policy iteration algorithm (PIA). ....	59
Figure 4.8: Decision calculation algorithm executed by the PIA in the PIR step. ....	61
Figure 5.1: Definition of the policy, the state and the macrostate spaces for the single-link network. ....	68
Figure 5.2: Definition of the exact and the approximate reward gains for the single-link network. ....	71
Figure 5.3: Simulation results for the basic CAC scheme, the exact and the approximate PIA. ....	74
Figure 5.4: Analytical and simulated average link reward rate vs. offered traffic [RB16b]. ....	75
Figure 5.5: Reward loss vs offered traffic for the basic CAC scheme and the PIA [RB16b]. ....	76
Figure 5.6: Network topologies and definition of connection class parameters [RB16d]. ....	77
Figure 5.7: Simulation results for the KSP scheme and the PIA with the MDP rule [RB16d]. ....	79
Figure 5.8: Performance comparison of the four PIA admission decision rules [RB17a]. ....	79
Figure 5.9: PIA convergence for the partial-mesh network applying the MDP-PGMC rule. ....	80
Figure 5.10: Performance of the PIA using RF spectrum allocation scheme. ....	81
Figure 5.11: Blocking probabilities vs reward parameters MDP rule [RB16d]. ....	84
Figure 5.12: Blocking probabilities vs reward parameters MDP-PGMC rule [RB17a]. ....	84
Figure 6.1: Time-sequence diagram centralized implementation of the VDO. ....	88
Figure 6.2: Time-sequence diagram implementation of the PIR with the 3WHS protocol. ....	88
Figure 6.3: Example of the course of a Pass1 and a Pass2 signalling message probes. ....	89
Figure 6.4: Example of a task graph for a protocol made up of six tasks. ....	91
Figure 6.5: The four basic task graph components. ....	91
Figure 6.6: Task graph reduction for a protocol with six tasks. ....	93
Figure 6.7: Task graph for the 3WHS protocol and its intermediate reduction steps [RB17c]. ....	94
Figure 6.8: Optical network with distances in Km and definition of parameters [RB17c]. ....	97
Figure 6.9: Connection setup latencies for the case of deterministic task processing times. ....	99

Figure 6.10: Connection setup latencies for the case of exponential processing times. ....	100
Figure 6.11: Pass1 and Pass2 latencies for the case of deterministic task processing times. ....	101
Figure 6.12: Pass1 and Pass2 latencies for the case of exponential processing times. ....	102
Figure 6.13: Latency contributors for the case of deterministic task processing times. ....	103
Figure 6.14: Latency contributors for the case of exponential processing times. ....	104
Figure 7.1: Bottom-up framework for TCO calculation [RB17b]. ....	109
Figure 7.2: Design of ROADM architectures (a two-degree node is shown). ....	112
Figure 7.3: Network topologies with link lengths in Km and definition of component costs. ....	116
Figure 7.4: Dimensioning of the number of transponders to install at a network node. ....	116
Figure 7.5: Infrastructure costs vs traffic load for the three ROADM designs. ....	119
Figure 7.6: Transponders required by the colorless and the colorless & directionless solutions. ....	119
Figure 7.7: Percentage contribution of ROADMs, transponders and OLAs to the overall cost. ....	120

# List of Tables

Table 3.1: Notation of relevant variables and parameters defined in Chapter 3. ....	18
Table 3.2: Definition of possible states for the $i$ th spectrum slot on a network link. ....	19
Table 3.3: Solution to the linear system in Fig. 3.11 for different traffic loads. ....	37
Table 3.4: Network reward gains used to calculate a decision for class-1/class-2 arrivals in state $x_2$ . ...	38
Table 3.5: Decisions calculated by the PIR for class-1 and class-2 arrivals in state $x_2$ . ....	38
Table 4.1: Notation for relevant variables and parameters defined in Chapter 4. ....	42
Table 4.2: Comparison between the exact and the approximate state-dependent reward models. ....	47
Table 4.3: Admission decision rules used by the PIR within the $i$ th iteration cycle [RB17a]. ....	63
Table 4.4: Comparison between the exact and the approximate PIA algorithms. ....	63
Table 5.1: Parameters for the comparison of the exact and the approximate reward models. ....	69
Table 5.2: Comparison of the exact and the approximate link reward rates. ....	69
Table 5.3: Comparison of the exact and the approximate transient reward values. ....	70
Table 5.4: Relative approximate error long-term reward gain $g_1(n_1, \pi_1)$ . ....	71
Table 5.5: Relative approximate error long-term reward gain $g_1(n_2, \pi_1)$ . ....	72
Table 5.6: Relative approximate error long-term reward gain $g_1(n_3, \pi_1)$ . ....	72
Table 5.7: Relative approximate error long-term reward gain $g_1(n_5, \pi_1)$ . ....	72
Table 5.8: Relative approximate error long-term reward gain $g_2(n_1, \pi_1)$ . ....	72
Table 5.9: Relative approximate error long-term reward gain $g_2(n_2, \pi_1)$ . ....	72
Table 5.10: Parameter settings for performance evaluation of the PIA in the single-link network. ....	73
Table 5.11: Parameter settings for performance evaluation of the approximate PIA [RB16b]. ....	75
Table 5.12: Relative approximate error for the arrival rates estimated by the PIA. ....	76
Table 5.13: Percentage of connections admitted on each candidate path [RB17a]. ....	80
Table 5.14: Reward losses obtained by FF and RF algorithms in the partial-mesh network. ....	82
Table 5.15: Reward losses obtained by FF and RF algorithms in the ring network. ....	82
Table 5.16: Definition of parameters for GoS control simulations [RB16d, RB17a]. ....	83
Table 5.17: Grade of service equalization partial-mesh network MDP-PGMC rule [RB17a]. ....	85
Table 6.1: Variables describing the 3WHS protocol. ....	88
Table 7.1: Typical technical specifications of flex-grid optical transponders. ....	110
Table 7.2: Definition of relevant component costs for dynamic optical networks. ....	115
Table 7.3: Calculation of infrastructure costs for dynamic optical networks. ....	115
Table 7.4: Summary PIA performance with the MDP-PGMC rule (first-fist spectrum allocation). ....	118

# Bibliography

- [ADB<sup>+</sup>13] R.C. Almeida, R.A. Delgado, C.J. Bastos-Filho, D.A. Chaves, H.A. Pereira, and J.F. Martins-Filho. An evolutionary spectrum assignment algorithm for elastic optical networks. *Proceedings of 15th International Conference on Transparent Optical Networks (ICTON)*, June 2013.
- [ANE<sup>+</sup>11] S. Azodolmolky, R. Nejabati, E. Escalona, R. Jayakumar, N. Efstathiou, and D. Simeonidou. Integrated OpenFlow-GMPLS control plane: An overlay model for software defined packet over optical networks. *Proceedings of the 37th European Conference on Optical Communications (ECOC)*, September 2011.
- [ASMW11] J. Ahmed, F. Solano, P. Monti, and L. Wosinska. Traffic re-optimization strategies for dynamically provisioned WDM networks. *Proceedings of the 15th International Conference on Optical Network Design and Modeling (ONDM)*, February 2011.
- [AWY13] S. Azodolmolky, P. Wieder, and R. Yahyapour. Cloud computing networking: Challenges and opportunities for innovations. *IEEE Communications Magazine*, vol. 51, pp. 54–62, Jul. 2013.
- [BCC<sup>+</sup>11] G. Bosco, V. Curri, A. Carena, P. Poggiolini, and F. Forghieri. On the performance of Nyquist-WDM terabit superchannels based on PM-BPSK, PM-QPSK, PM-8QAM or PM-16QAM subcarriers. *Journal of Lightwave Technology*, vol. 29, pp. 53–61, January 2011.
- [Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [Ber03] L. Berger. Generalized multi-protocol label switching, GMPLS, signaling functional description. *RFC 3471*, January 2003.
- [Bou11] J. L. Boudec. *Performance Evaluation of Computer and Communication Systems*. EPFL Press, 2011.
- [BSBS08] J. Berthold, A. A. M. Saleh, L. Blair, and J. M. Simmons. Optical networking: past, present, and future. *Journal of lightwave technology*, vol. 20, pp. 1104-1118, May 2008.
- [CCC<sup>+</sup>12] A. L. Chiu, G. Choudhury, G. Clapp, R. Doverspike, M. Feuer, J. W. Gannett, J. Jackel, G. T. Kim, J. G. Klinecicz, T. J. Kwon, G. Li, P. Magill, J. M. Simmons, R. A. Skoog, J. Strand, A. V. Lehmen, B. J. Wilson, S. L. Woodward, and D. Xu. Architectures and protocols for capacity efficient, highly dynamic and highly resilient core networks. *Journal of Optical Communications and Networking*, vol. 4, pp. 1-14, January 2012.
- [CDF<sup>+</sup>15] G. Clapp, P. Douyon, D. Freimuth, K. Gullapalli, B. Han, J. Hartley, A. Mahimkar, E. Mavrogiorgis, J. O'Connor, J. Pastor, K. Ramakrishnan, M. Rauch, M. Stadler, A. Von Lehmen, B. Wilson, and S. L. Woodward. Using SDN technology to enable cost-effective bandwidth-on-demand for cloud services. *Journal of Optical Communications and Networking*, vol.7, pp. A326-A334, February 2015.
- [CKE<sup>+</sup>12] M. Channegowda, P. Kostecki, N. Efstathiou, S. Azodolmolky, R. Nejabati, P. Kaczmarek, A. Autenrieth, J. Elbers, and D. Simeonidou. Experimental evaluation of extended openflow deployment for high-performance optical networks. *Proceedings*



- of the 38th European Conference on Optical Communications (ECOC)*, September 2012.
- [CNDR01] C. Chigan, R. Nagarajan, Z. Dziong, and T. G. Robertazzi. On the capacitated loss network with heterogeneous traffic and contiguous resource allocation constraints. *Proceedings of Applied Telecommunication Symposium*, vol. 33, pp.153-160, April 2001.
- [CNF<sup>+</sup>12] M. Channegowda, R. Nejabati, M. Fard, S. Peng, N. Amaya, G. Zervas, D. Simeonidou, R. Vilalta, R. Casellas, R. Martínez, R. Muñoz, L. Liu, T. Tsuritani, I. Morita, A. Autenrieth, J. Elbers, P. Kostecki, and P. Kaczmarek. First demonstration of an OpenFlow based software-defined optical network employing packet, fixed, and flexible DWDM grid technologies on an international multi-domain testbed. *Proceedings of the 38th European Conference on Optical Communications (ECOC)*, September 2012.
- [Col13] B. Collings. New devices enabling software-defined optical networks. *In IEEE Communications Magazine*, vol. 51, pp. 66–71, March 2013.
- [CR93] S. P. Chung and K. W. Ross. Reduced load approximations for multirate loss networks. *IEEE transactions on communications*, vol. 41, pp. 1222-1231, August 1993.
- [CTV11a] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos. Elastic bandwidth allocation in flexible OFDM-based optical networks. *Journal of Lightwave Technology*, vol. 29, pp. 1354–1366, May 2011.
- [CTV11b] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos. Dynamic bandwidth allocation in flexible OFDM-based networks. *Proceedings of the OFC/NFOEC*, March 2011.
- [CTV13] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos. Time varying spectrum allocation policies and blocking analysis in flexible optical networks. *IEEE Journal on Selected areas in communications*, vol. 31, pp. 13–25, January 2013.
- [CVM<sup>+</sup>06] K. Casier, S. Verbrugge, R. Meersman, J. V. Ooteghem, D. Colle, M. Pickavet, and P. Demeester. A fair cost allocation scheme for CapEx and OpEx for a network service provider. *Proceedings of the 5th Conference of Telecommunication, Media and Internet Techno-Economics*, June 2006.
- [CVR<sup>+</sup>12] A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio. Dynamic routing and spectrum (re)allocation in future flexgrid optical networks. *Computer Networks*, vol. 56, pp 2869–2883, August 2012.
- [CW03] C. Courcoubetis and R. Weber. *Pricing Communication Networks*. John Wiley & Sons, 2003.
- [Das12] S. Das. *Unified control architecture for packet and circuit network convergence*. PhD. dissertation, Stanford University, Stanford, June 2012.
- [DBÇ15] D. M. Diez, C. D. Barr, and M. Çetinkaya-Rundel. *OpenIntro Statistics*. Duke University Press, 2015.
- [DM89] Z. Dziong and L. Mazon. Control of multi-service loss networks. *Proceedings of the 28th Conference on Decision and Control*, pp. 1099-1104, Tampa, December 1989.
- [DM92] Z. Dziong and L. Mazon. An analysis of near optimal call admission and routing model for multi-service loss networks. *Proceedings of the 11th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, pp. 141-152, Florence, May 1992.

- [DM94] Z. Dziong and L. Mazon. Call admission and routing in multi-service loss networks. *IEEE Transactions on Communications*, vol. 42, pp. 2011-2022, April 1994.
- [DPKW88] Z. Dziong, M. Pióro, U. Körner, and T. Wickberg. On adaptive call routing strategies in circuit switched networks - maximum revenue approach. *Proceedings of the 12th International Teletraffic Congress ITC*, Torino, June 1988.
- [DPM12] S. Das, G. Parulkar, and N. McKeown. Why OpenFlow/SDN can succeed where GMPLS failed. *Proceedings of the 38th European Conference on Optical Communications (ECOC)*, September 2012.
- [Dzi97] Z. Dziong. *ATM Network Resource Management*. McGraw-Hill, 1997.
- [FVA06] A. Farrel, J. P. Vasseur, and J. Ash. A path computation element (PCE)-based architecture. *IETF RFC 4655*, August 2006.
- [GBS<sup>+</sup>10] S. Gringeri, B. Basch, V. Shukla, R. Egorov, and T. J. Xia. Flexible architectures for optical transport nodes and networks. *IEEE Communications Magazine*, July 2010.
- [GCZ<sup>+</sup>13] O. Gonzalez de Dios, R. Casellas, F. Zhang, D. Ceccarelli, and I. Hussain. Framework and requirements for GMPLS based control of flexi-grid DWDM networks. *IETF Internet-Draft*, Oct. 2013.
- [GJLY12] O. Gerstel, M. Jinno, A. Lord, and S. J. Yoo. Elastic optical networking: A new dawn for the optical layer?. *IEEE Communications Magazine*, vol. 50, pp. 12–20, February 2012.
- [GREB17] A. Ghallaj, R. Romero-Reyes, M. Ermel, and T. Bauschert. Optimizing spectrum allocation in flex-grid optical networks. *Proceedings of the 18th ITG-Symposium in Photonic Networks (Photonische Netze)*, Leipzig, May 2017.
- [HKT00] R. H. Hwang, J. F. Kurose, and D. Towsley. MDP routing for multi-rate loss networks. *Computer Networks Journal*, vol. 34, pp. 241-261, August 2000.
- [How60] R. A. Howard. *Dynamic programming and Markov processes*. Technology Press of Massachusetts Institute of Technology, Cambridge, Massachusetts, 1960.
- [Hun89] P. J. Hunt. Implied costs in loss networks. *Advances in Applied Probability*, vol. 21, pp. 661-680, September 1989.
- [Hwa93] R. H. Hwang. *Routing in high-speed networks*. Ph.D. dissertation, University of Massachusetts, 1993.
- [ITU12] ITU-T Recommendation G.694.1 (ed. 2.0). Spectral grids for WDM applications: DWDM frequency grid. February 2012.
- [JTK<sup>+</sup>08] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, T. Yoshimatsu, T. Kobayashi, Y. Miyamoto, K. Yonenaga, A. Takada, O. Ishida, and S. Matsuoka. Demonstration of novel spectrum-efficient elastic optical path network with per-channel variable capacity of 40 Gb/s to over 400 Gb/s. *Proceedings of the 34th European Conference on Optical Communications (ECOC)*, September 2008.
- [JTK<sup>+</sup>09] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka. Spectrum efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies. *IEEE Communications Magazine*, vol. 47, pp. 66–73, November 2009.
- [Kau81] J. Kaufman. Blocking in a shared resource environment. *IEEE transactions on communications*, vol. 29, pp. 1474-1481, October 1981.

- [Kel88] F. P. Kelly. Routing in circuit-switched networks: Optimization, shadow prices and decentralization. *Advances in Applied Probability*, vol. 20, pp. 112-144, March 1988.
- [Kel91] F. P. Kelly. Loss Networks. *The annals of Applied Probability*, vol. 1, pp. 319-378, August 1991.
- [Kel11] F. P. Kelly. *Reversibility and Stochastic Networks*. Cambridge university press, 2011.
- [Kri91] K. R. Krishnan. Adaptive state-dependent traffic routing using on-line trunk-group measurements. *Proceedings of the 13th International Teletraffic Congress ITC*, pp. 407-411, June 1991.
- [KRM<sup>+</sup>10] A. Klekamp, O. Rival, A. Morea, R. Dischler, and F. Buchali. Transparent WDM network with bitrate tunable optical OFDM transponders. *Proceedings of the OFC/NFOEC, San Diego*, pp. NTuB5, March 2010.
- [KRV<sup>+</sup>13] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez and J. Comellas. Elastic spectrum allocation for time-varying traffic in flexgrid optical networks. *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 26-38, January 2013.
- [Küh14] P. J. Kühn. Performance and energy efficiency of parallel processing in data center environments. *Proceedings of the 3<sup>rd</sup> International Workshop on Energy Efficient Data Centers E2DC*, Cambridge, June 2014.
- [Küh16] P. J. Kühn. Real time control in 5G: Embedded communication networks. A system-theoretic modeling approach. *VDE/ITG Section 5.2.4 Meeting on IP and Mobility*, Dresden, June 2016.
- [KW11] M. Klinkowski and K. Walkowiak. Routing and spectrum assignment in spectrum sliced elastic optical path network. *IEEE Communication Letters*, vol. 15, pp. 884–886, August 2011.
- [KY14] F. P. Kelly and E. Yudovina. *Stochastic Networks*. Cambridge university press, 2014.
- [LCT<sup>+</sup>12] L. Liu, R. Casellas, T. Tsuritani, I. Morita, R. Martínez, and Raul Muñoz. Interworking between OpenFlow and PCE for dynamic wavelength path control in multi-domain WSON. *Proceedings of the OFC/NFOEC*, Los Angeles, March 2012.
- [LDL13] P. Lafaye de Micheaux, R. Drouilhet, and B. Lique. *The R Software, Fundamentals of Programming and Statistical Analysis*. Springer, 2013.
- [LYL<sup>+</sup>12] J. López, Y. Ye, V. López, F. Jiménez, R. Duque, and P. M. Krummrich. Cost evaluation for flexible-grid optical networks. *IEEE Globecom Workshops*, pp. 358-363, December 2012.
- [Man04] E. Mannie. Generalized multi-protocol label switching, GMPLS, architecture. *RFC 3945*, October 2004.
- [MCD<sup>+</sup>11] A. Mahimkar, A. Chiu, R. Doverspike, M. D. Feuer, P. Magill, E. Mavrogiorgis, J. Pastor, S. L. Woodward, and J. Yates. Bandwidth on demand for inter-data center communication. *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, November 2011.
- [Mei13] W. Meiqian. *Performance analyses of circuit switched networks*. Ph.D. dissertation, City University of Hong Kong, September 2013.
- [Nor97] J. R. Norris. *Markov Chains*. Cambridge university press, 1997.

- [Nor02] E. Nordström. Near-optimal CAC and routing for multi-service networks by Markov decision theory. *Proceedings of the fourth annual conference for promoting research and advanced education in topics of IT*, Skövde, April 2002.
- [ONF13] Open Networking Foundation (ONF). *Optical transport working group*. <http://www.opennetworking.org/working-groups/optical-transport>, 2013.
- [ONF14] Open Networking Foundation (ONF). *Technical report on SDN Architecture*. <https://www.opennetworking.org>, 2014.
- [PFRC11] S. Poole, S. Frisken, M. Roelens, and C. Cameron. Bandwidth-flexible ROADMs as network elements. *Proceedings of the OFC/NFOEC*, March 2011.
- [PJJW11a] A. Patel, P.N. Ji, J.P. Jue, and T. Wang. Dynamic routing, wavelength assignment, and allocation in transparent flexible optical WDM networks. *Proceedings of Optical Metro Networks and Short-Haul Systems III*, vol. 7959, January 2011.
- [PJJW11b] A. Patel, P. N. Ji, J. Jue, and T. Wang. Defragmentation of transparent flexible optical WDM (FWDM) networks. *Proceedings of the OFC/NFOEC*, March 2011.
- [PP12] J. Pedro and S. Pato. Quantifying the Impact of DWDM Nodes with Flexible Add/Drop Port Utilization for Dynamic Connection Setup. *Proceeding of the OFC*, March 2012.
- [RAB18] R. Romero-Reyes, O. Ali, and T. Bauschert. Infrastructure Costs in Dynamic Optical Networks. *Proceedings of the 10th Latin-American Conference on Communications LATINCOM*, Guadalajara, November 2018.
- [RB15] R. Romero-Reyes and T. Bauschert. Online routing and spectrum assignment in flexgrid optical networks. *Proceedings of the 17th International Conference on Transparent Optical Networks ICTON*, Budapest, July 2015.
- [RB16a] R. Romero-Reyes and T. Bauschert. Adaptive routing and spectrum assignment in flex-grid optical networks. *Proceedings of the 17th ITG-Symposium in Photonic Networks (Photonische Netze)*, Leipzig, May 2016.
- [RB16b] R. Romero-Reyes and T. Bauschert. Reward-based connection admission control in flex-grid optical networks. *Proceedings of the 20th International Conference on Optical Network Design and Modeling (ONDM)*, Cartagena, May 2016.
- [RB16c] R. Romero-Reyes and T. Bauschert. State-dependent connection admission control and routing and spectrum assignment in multirate flex-grid optical networks. *Proceedings of the 18th International Conference on Transparent Optical Networks ICTON*, Trento, July 2016.
- [RB16d] R. Romero-Reyes and T. Bauschert. Reward-based online routing and spectrum assignment in flex-grid optical networks. *Proceedings of the 17th International Network Strategy and Planning Symposium NETWORKS*, Montreal, September 2016.
- [RB17a] R. Romero-Reyes and T. Bauschert. Adaptive and state-dependent online resource allocation in dynamic optical networks. *Journal of Optical Communications and Networking*, vol. 9, pp. B64-B77, March 2017.
- [RB17b] R. Romero-Reyes and T. Bauschert. Bottom-up framework for cost allocation to services in telecommunication networks. *NETNOMICS: Economic Research and Electronic Networking*, vol. 18, pp. 81-105, May 2017.
- [RB17c] R. Romero-Reyes and T. Bauschert. Analytical performance evaluation of connection setup latency in dynamic optical networks. *Proceedings of the 19th International Conference on Transparent Optical Networks ICTON*, Girona, July 2017.

- [RB18a] R. Romero-Reyes and T. Bauschert. Task-Graph reductions based performance analysis of a 3-way handshake connection setup protocol for dynamic optical networks. *Proceedings of the 19th ITG-Symposium in Photonic Networks (Photonische Netze)*, Leipzig, June 2018.
- [RB18b] R. Romero-Reyes and T. Bauschert. Online Resource Allocation in Dynamic Optical Networks. *Proceedings of the 30th International Teletraffic Congress ITC*, Vienna, September 2018.
- [REB15] R. Romero-Reyes, M. M. Ejaz, and T. Bauschert. Techno-economic analysis of multiservice networks: Cost allocation to services, a study for flex-grid optical networks. *Proceedings of the 12th Conference of Telecommunication, Media and Internet Techno-Economics (CTTE)*, Munich, November 2015.
- [RKD<sup>+</sup>13] F. Rambach, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotti, L. Serra, and V. López. A Multilayer cost model for metro/core networks. *Journal of Optical Communications and Networking*, vol. 5, pp. 210-225, March 2013.
- [Rob05] L. Roberts. Enabling Data-Intensive iGrid Applications with Advanced Network Technology. In *iGrid 2005*, San Diego, CA, September 2005.
- [Ros95] K. W. Ross. *Multiservice Loss Models for Broadband Telecommunication Networks*. London: Springer, 1995.
- [Rpr17] The R Project for Statistical Computing. <https://www.r-project.org/>, 2017.
- [Sal06] A. A. M. Saleh. Dynamic multi-terabit core optical networks: architecture, protocols, control and management (CORONET). *DARPA BAA 06-29, Proposer Information Pamphlet*, August 2006.
- [Sal07] A. A. M. Saleh. Technologies, architecture and services for the next-generation core optical networks. *Proceedings of the OFC/NFOEC, Workshop on the Future Optical Networking*, Anaheim, March 2007.
- [SCG<sup>+</sup>12] R. Skoog, G. Clapp, J. Gannett, A. Neidhardt, A. V. Lehman, and B. Wilson. Architectures, protocols and design for highly dynamic optical networks. *Optical Switching and Networking Journal*, vol. 9, pp. 240-251, July 2012.
- [SDM10] D. A. Schupke, S. Duhovnikov, and C. Meusburger. Guidelines for Connection-Level Performance Simulation of Optical Networks. *Technical Report, LKN-TR-5, Institute of Communication Networks, TU München*, January 2010.
- [SGK<sup>+</sup>14] R. Skoog, J. Gannett, K. Kim, H. Kobrinski, M. Rauch, A. V. Lehmen, and B. Wilson. Analysis and implementation of a 3-Way handshake signaling protocol for highly dynamic transport networks. *Proceedings of the OFC*, pp. WK.1, March 2014.
- [SHS<sup>+</sup>14] J. Shin, R. Hwang, Y. Shim, S. Cho, J. Park, and A. Choi. Transport SDN: Trends, standardization and architecture. *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC)*, October 2014.
- [Sim14] J. M. Simmons. *Optical Network Design and Planning*. New York: Springer, 2014.
- [SN09] R. A. Skoog and A. L. Neidhardt. A fast, robust signaling protocol for enabling highly dynamic optical networks. *Proceedings of the OFC/NFOEC*, San Diego, pp. NTuB5, March 2009.
- [SP10] F. Solano and M. Pióro. Lightpath Reconfiguration in WDM Networks. *Journal of Optical Communications and Networking*, vol. 2, pp. 1010-1021, November 2010.

- [SS11] A. A. M. Saleh and J. M. Simmons. Technology and architecture to enable the explosive growth of the internet. *IEEE Communications Magazine*, vol. 49, pp. 126-132, January 2011.
- [SS12] A. A. M. Saleh and J. M. Simmons. All-optical networking – evolution, benefits, challenges, and future vision. *Proceedings of the IEEE*, vol. 100, pp. 1105-1117, May 2012.
- [SW10] R. A. Skoog and B. J. Wilson. Transponder pool sizing in highly dynamic translucent WDM optical networks. *Proceedings of the OFC/NFOEC*, San Diego, pp. NTuA3, March 2010.
- [TAK<sup>+</sup>14] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Salama, and G.N. Rouskas. Spectrum management techniques for elastic optical networks: A survey. *Optical Switching and Networking Journal*, vol. 13, pp. 34-48, February 2014.
- [TAS<sup>+</sup>14] I. Tomkos, S. Azodolmolky, J. Solé-Pareta, D. Careglio, and E. Palkopoulou. A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges. *Proceedings of the IEEE*, vol. 102, pp. 1317-1337, June 2014.
- [THS<sup>+</sup>11] T. Takagi, H. Hasegawa, K. I. Sato, Y. Sone, A. Hirano, and M. Jinno. Disruption minimized spectrum defragmentation in elastic optical path networks that adopt distance adaptive modulation. *Proceedings of the 37th European Conference on Optical Communications (ECOC)*, September 2011.
- [THT<sup>+</sup>14] Y. Takita, T. Hashiguchi, K. Tajima, T. Katagiri, T. Naito, Q. Zhang, X. Wang, I. Kim, P. Palacharla, and M. Sekiya. Network reconfiguration targeting minimum connection disruption. *Proceedings of the International Conference on Optical Network Design and Modeling (ONDM)*, May 2014.
- [Tij86] H.C. Tijms. *Stochastic modelling and analysis: a computational approach*. John Wiley & Sons, Inc., New York, 1986.
- [TTHK15] Y. Takita, K. Tajima, T. Hashiguchi, and T. Katagiri. Maximizing spectrum utilization of optical networks through intelligent wavelength defragmentation. *Proceedings of the International Conference on Photonics in Switching (PS)*, September 2015.
- [TTHK16] Y. Takita, K. Tajima, T. Hashiguchi, and T. Katagiri. Wavelength defragmentation with minimum optical path disruptions for seamless service migration. *Proceedings of the OFC/NFOEC*, March 2016.
- [TVW<sup>+</sup>13] M. Tahon, S. Verbrugge, P. J. Willis, P. Botham, D. Colle, M. Pickavet, and P. Demeester. Real options in telecom infrastructure projects - A Tutorial. *IEEE Communications Surveys and Tutorials*, vol. 16, pp. 1157-1173, July 2013.
- [VCOL08] S. Verbrugge, K. Casier, J. V. Ooteghem, and B. Lannoo. Practical steps in techno-economic evaluation of network deployment planning part 1: methodology overview. *The 13th International Telecommunications Network Strategy and Planning Symposium*, pp. 1-101, September 2008.
- [VPW<sup>+</sup>05] S. Verbrugge, S. Pasqualini, F. J. Westphal, M. Jäger, A. Iselt, A. Kirstädter, R. Chahine, D. Colle, M. Pickavet, and P. Demeester. Modeling operational expenditures for telecom operators. *Proceedings of the 9th Conference on Optical Network Design and Modeling*, pp. 455-466, February 2005.
- [WFJA10] S. L. Woodward, M. D. Feuer, J. Jackel, and A. Agarwal. Massively-scaleable highly-dynamic optical node design. *Proceeding of the OFC/NFOEC*, San Diego, pp. JThA18, March 2010.

- [WGK<sup>+</sup>15] B. Wilson, J. W. Gannett, H. Kobrinski, R. Skoog, and A. V. Lehmen. Fast, secure call setup in CORONET multiple carrier domain networks. *Journal of Optical Communications and Networking*, vol. 7, pp. 505-515, June 2015.
- [WWH<sup>+</sup>11] X. Wan, L. Wang, N. Hua, H. Zhang, and X. Zheng. Dynamic routing and spectrum assignment in flexible optical path networks. *Proceedings of the OFC/NFOEC*, pp. JWA55, March 2011.
- [WWL10] W. Wei, C. Wang, and X. Liu. Adaptive IP/optical OFDM networking design. *Proceedings of the OFC/NFOEC*, March 2010.
- [WZK<sup>+</sup>11] X. Wang, Q. Zhang, I. Kim, P. Palacharla, and M. Sekiya, Blocking performance in dynamic flexible grid optical networks—what is the ideal spectrum granularity?. *Proceedings of 37th European Conference on Optical Communication (ECOC)*, September 2011.
- [WZZ<sup>+</sup>12] Y. Wang, J. Zhang, Y. Zhao, J. Wang, and W. Gu. Routing and spectrum assignment by means of ant colony optimization in flexible bandwidth networks. *Proceedings of the OFC/NFOEC*, March 2012.
- [You85] H.P. Young. *Cost allocation: methods, principles, applications*. Amsterdam: North Holland Publishing Co, 1985.
- [YZZ<sup>+</sup>12] H. Yang, Y. Zhao, J. Zhang, S. Wang, W. Gu, Y. Lin, and, Y. Lee, Cross stratum optimization of application and network resource based on global load balancing strategy in dynamic optical networks. *Proceedings of the OFC/NFOEC*, March 2012.
- [YZZ<sup>+</sup>14] Y. Yu, J. Zhang, Y. Zhao, H. Li, Y. Ji, and W. Gu. Exact performance model for spectrum allocation in flexible grid optical networks. *Optical Fiber Technology Journal*, vol. 20, pp. 75-83, January 2014.
- [ZDMM13] G. Zhang, M. De Leenheer, A. Morea, and B. Mukherjee. A survey on OFDM-based elastic core optical networking. *IEEE Communications Magazine*, vol. 15, pp. 65–87, First Quarter 2013.
- [ZRP00] Y. Zhu, G. N. Rouskas, and H. G. Perros. A path decomposition approach for computing blocking probabilities in wavelength-routing networks. *IEEE transactions on networking*, vol. 8, pp. 747-762, December 2000.
- [ZZY<sup>+</sup>13] J. Zhang, Y. Zhao, H. Yang, Y. Ji, H. Li, Y. Lin, G. Li, J. Han, Y. Lee, and T. Ma. First demonstration of enhanced software defined networking (eSDN) over elastic grid (eGrid) optical networks for data center service migration. *Proceedings of the OFC/NFOEC*, March 2013.
- [ZZZ<sup>+</sup>13] J. Zhang, J. Zhang, Y. Zhao, H. Yang, X. Yu, L. Wang, and X. Fu. Experimental demonstration of OpenFlow-based control plane for elastic lightpath provisioning in flexi-grid optical networks. *Journal Optics Express*, vol. 21, pp. 1364–1373, January 2013.





# **Annex**

This annex contains additional information as required by the Technische Universität Chemnitz.



## Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts habe ich Unterstützungsleistungen von folgenden Person erhalten:

Prof. Dr.-Ing. Thomas Bauschert

Weitere Personen waren an der Abfassung der vorliegenden Arbeit nicht beteiligt. Die Hilfe eines Promotionsberaters habe ich nicht in Anspruch genommen. Weitere Personen haben von mir keine geldwerten Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Chemnitz, den 31.05.2018

Ronald Romero Reyes



## Thesen

1. Nowadays optical transport networks are circuit-switched systems that carry customer demands on optical connections. The setup times of these connections are in the order of weeks as in most cases manual configuration is required. Once established, connections remain active for months or years. The emergence of grid computing and cloud-based services is imposing new connectivity requirements on the optical layer. This has raised awareness on the need for a changeover to dynamic optical networks. Instead of providing long-term installed constant bit-rate connectivity, these networks are envisioned to offer BoD in the form of connections with different bit-rate requirements, with random interarrival and holding times, and with low connection setup latencies.
2. To provide BoD the network control plane has to enable automatic configuration of optical connections without any manual intervention. Flex-grid WDM, directionless & colorless ROADMs, and bandwidth variable & tunable transponders are the technological prerequisites.
3. To design and operate dynamic optical networks, network planning and online resource allocation are essential. Network planning ensures that sufficient capacity is installed to cope with an expected traffic demand growth. Online resource allocation, on the other hand, ensures that the installed capacity is efficiently allocated to connections so as to attain a desired performance objective. This is accomplished by implementing network control functions for RSA, CAC and GoS control. Upon arrival of a connection request, RSA is responsible for calculating a set of candidate lightpaths. Then CAC applies decision rules to select the lightpath on which the connection is carried. Although parts of the same problem, RSA and CAC can be solved separately so as to reduce modelling and implementation complexity.
4. Extrinsic and intrinsic factors determine the performance of dynamic optical networks. Among the extrinsic factors are the statistical properties of the bandwidth requirements, the interarrival and the holding times of connections. These properties, in principle, are independent of - and unknown to - the network. On the other hand, intrinsic factors include online resource allocation algorithms whereby lightpaths are provisioned on demand. The overall network performance (e.g. the overall blocking probability) emerges from the interaction between extrinsic and intrinsic factors. Understanding this causal relationship is essential for the design of an efficient network control policy. With this knowledge, it is possible to cope with the randomness of connection requests. Fast, adaptive and state-dependent resource allocation algorithms can be tailored to achieve a desired performance objective (or large-scale effect). Given the random nature of connection requests, dynamic optical networks behave as large-scale stochastic systems whose state (i.e. the network resource occupancy) changes by following the decisions of the resource allocation algorithm.
5. Resource allocation can be tackled as a reward-based online optimization problem. Connections are classified into classes which are assigned a reward whose meaning and value is defined by the network operator. It quantifies the immediate benefit that a connection brings to the network if admitted. The advantage of this approach is that the reward can be set either to optimize any desired performance objective (e.g. blocking minimization, maximization of carried traffic or economic revenue) or to equalize or differentiate the GoS of the connection classes. The objective of the optimization problem is to maximize the rate at which the network earns reward.
6. Online resource allocation is the mechanism whereby the network assigns, on demand, spectrum resources to connections (in addition to link spectrum slots, this includes transponders and ROADM resources as well). Such a mechanism is a decision making process where for every connection request, it is decided whether admission is granted or not. In case of admission, a suitable lightpath has to be defined for the connection. A decision depends on two factors: the class of the connection request and the network state at the time of the connection arrival. The collection of decisions that instruct the network how to process any incoming connection request defines the resource allocation policy. MDP theory can be used to formulate the reward-based optimization problem as a Markov decision process that is solved by an exact PIA. For that, the network resource allocation is modelled as a continuous-time stochastic process where the state is defined as the configuration of the optical spectrum on all network links. All feasible network states fulfil the spectrum contiguity and continuity constraints. With the exact PIA, it is possible to determine the state-dependent resource allocation policy that maximizes the rate at which the network earns reward.

7. The exact PIA is based on an iteration procedure that takes an arbitrarily selected resource allocation policy to calculate (via a sequence of online iterations) the policy that maximizes the network reward rate. The calculation of the policy decisions is based on the concept of long-term reward gain. It quantifies the long-term impact that a connection would have on the network reward if it is admitted on a specific lightpath. The reward gain is state and policy dependent, and can be either positive or negative. If positive, the connection leads to a long-term increment in the reward earned by the network. Otherwise, (i.e. if negative), the connection would prevent the network from admitting more valuable traffic, thereby causing reward losses. The exact PIA calculates the optimum policy so that a connection is admitted on a lightpath that brings the maximum positive reward gain. Given the huge cardinality of the network state-space for networks of realistic size, the straightforward calculation of the reward gains, and therefore of the optimum policy, is computationally intractable.
8. The computational complexity of the exact PIA can be overcome by an approximate PIA that resorts to the link independence assumption, and to a macrostate-dependent estimation of the reward gains. The link independence assumption is a decomposition approach, whereby the rate of reward from the network is approximated as the sum of the link reward rates. Thus, the problem is decomposed into separate link problems by assuming statistical independence of the link state distributions. This approach substantially reduces mathematical complexity and the computational effort required to calculate the policy decisions. The link macrostate represents the number of connections of each traffic class carried by the link at a certain time. It comprises different link states that fulfil the spectrum contiguity constraint. In the approximate PIA, it is assumed that a connection only changes the reward rates of the links that it uses. The implication of this is that upon arrival of a connection request, it is not necessary to know the overall network state to calculate a policy decision. Instead, it is sufficient to know the states of selected  $k$ -shortest paths that connect the source/destination nodes. The connection is admitted on the path that yields the maximum long-term reward gain. This gain is calculated as the sum of the macrostate-dependent reward gains of the links comprising the path. By this approach it is assumed that all lightpaths routed on the same path yield the same reward gain.
9. The approximate PIA overcomes computational complexity, and hence, it makes the reward-based optimization problem solvable. However, the resource allocation policies that it calculates are sub-optimal. In most networks, a performance comparison between the exact and the approximate PIA is not possible given that the exact policy cannot be calculated. From the performance evaluation results, it can be argued that although the link independence assumption is not always accurate, the performance of the approximate PIA is improved by defining simple MDP-based admission decision strategies, such as the MDP-SP, the MDP-PG and the MDP-PGMC rules. The performance of these rules depends on the spectrum allocation scheme used in the calculation of feasible lightpaths. In particular, compared to random-fit, the rules perform better when first-fit is used as spectrum allocation method. In fact, with first-fit, the MDP rules admit connections on lightpaths that tend to reduce spectrum fragmentation, thereby leading to less blocking in the network. The best performance is attained by the PIA variant that uses the MDP-PGMC rule with first-fit as spectrum allocation method. The differences in the PIA performance for first-fit and random-fit prove that the reward gains need not be alike for optical connections routed on the same path.
10. In dynamic optical networks connections can be rapidly set up and torn down. In some cases setup times in the range of milliseconds to seconds are required. To meet these stringent requirements, signaling protocols are needed that enable low setup latencies. Given that resource allocation is part of the connection setup process, the design and implementation of resource allocation algorithms has to consider the connection setup procedures applied in the network.
11. The implementation of the approximate PIA within the 3WHS protocol which provides a fast connection setup is feasible. Due to the link independence assumption the signalling load in the network remains quite low. The reason is that upon arrival of a connection request, it is only required to know the state of the  $k$ -shortest paths that connect the source/destination nodes. The collection of the actual states of those paths is easily accomplished by the probing mechanism of the 3WHS protocol. (This probing mechanism, which simultaneously collects information from  $k$ -shortest paths, is not enabled by the current GMPLS connection setup procedure.) For the exact PIA, neither GMPLS nor the 3WHS protocol are suitable to provide fast connection setup. The reason is that to establish a connection, the exact PIA must know the total actual network state. The collection of the

total state information via the signaling protocol precludes a fast connection setup. This statement is true for any resource allocation algorithm that needs global state information for resource allocation.

12. The latency of the 3WHS protocol can be evaluated by modelling the protocol as a task graph that represents the delays incurred in the connection setup process. By using reduction techniques, the graph is simplified so as to obtain a performance model that estimates the mean connection setup latency. The proposed analytical model provides accurate estimates of the protocol performance. For the proposed PIA implementation, the results show that, regardless of the network size, the cross-connection times of the ROADMs are the relevant contributors to the setup latency. For large-size networks, on the other hand, the connection setup latencies are mainly driven by the propagation delays of the signaling messages on the  $k$ -shortest paths probed by the protocol. The advantage of the proposed analytical method is that it can be extended to study the latencies incurred by any connection setup protocol and resource allocation algorithm.
13. The cost study performed on selected network scenarios indicates that, although not the least costly, colorless & directionless network infrastructures are the best solutions in terms of flexibility for dynamic optical networks. By adding the directionless property to colorless ROADMs, the number of transponders required in the network is substantially reduced (compared to purely colorless-based solutions). Furthermore, this property allows the transponders installed at a given network ROADM to add/drop connections over any ROADM link. With this feature, the network is able to automatically re-route traffic while providing quick recovery from link failures. These capabilities justify the additional costs incurred by the directionless property w.r.t. only colorless-based network designs.





## **Curriculum Vitae**

Ronald Romero Reyes is research associate at the chair for communication networks at the Technische Universität Chemnitz (TUC), Germany. He studied physics at the National University of Colombia and received, in 2006, a B.Sc. degree in Telecommunications Engineering from the District University of Bogotá, Colombia. In 2012, he got a M.Sc. degree in Communications Engineering from the Technische Universität München (TUM), Germany. His research interests are on the design and planning of advanced network architectures and services, techno-economic analysis of communications networks, cost modelling and regulation of ICT markets.

