N^o d'ordre : 2009telb0111

Thèse

présentée à

l'École Nationale Supérieure des Télécommunications de Bretagne

en habilitation conjointe avec l'Université de Rennes 1

en cotutelle avec la Universidad de la República en Uruguay

pour obtenir le grade de

Docteur de Télécom Bretagne

Mention : *Computer Sciences*

par

Pedro Casas Hernández

Statistical Analysis of Network Traffic for Anomaly Detection and Quality of Service Provisioning

---

Soutenue le 6 Juillet 2010 devant la commission d'Examen :

Composition du Jury :

Rapporteurs :	Fabrice Guillemin, France Télécom R&D Steve Uhlig, Technische Universität Berlin
Examinateurs :	Pablo Belzarena, Universidad de la República Stefano Giordano, University of Pisa Gerardo Rubino, INRIA-IRISA Sandrine Vaton, Télécom Bretagne
Invité :	Hervé Kerivin, Clemson University

A la memoria viviente de mi padre,
Carlos Alberto Casas García.

Gracias viejo.

All you need is love – *John Lennon*

Acknowledgements

This Ph.D. thesis is the result of my research works during the last three years, but it would not have been possible without the contributions and support of many people to whom I will always be grateful. To start with, I would like to deeply thank my thesis supervisor Sandrine, since the very beginning she gave me her complete support and guidance to develop my studies, as well as the necessary freedom to go in the directions I chose. Thank you Sandrine for giving me the opportunity of taking this PhD at Brest, I will always keep with me the prettiest memories from these years.

I am very grateful to my thesis director Gerardo Rubino, who was always there when I needed him. I would like to specially thank all the researchers I have worked with during the thesis: Lionel Fillatre, Igor Nikiforov, Pablo Belzarena, Thierry Chonavel, Hervé Kerivin, Walid Ben-Ameur, Federico Larroca, and Jean-Louis Rougier; working with you was not only an enriching experience, but also a real pleasure. Also thanks to the reviewers of the thesis and to the members of the jury for their willingness to evaluate my works.

Thanks to all the people from the Computer Science department at Télécom Bretagne, and from the Electrical Engineering Institute at the Engineering Faculty, Universidad de la República, I always felt at home in both places. I am specially grateful to María Le Goff, Laura Landin, María Misa, Armelle Lannuzel, Anne-Marie L'Hostis, Geneviève Larue, Martine Besnard, and Karine Langlet for their constant help in all the administrative aspects of the Ph.D., your efficiency and timeliness are only exceeded by your quality as human beings.

I will always be grateful to all the friends I met since my arrival to Brest; without you, one of the greatest adventures of my life would have been just a sad shadow of what it really was. Your friendship is the most important output of this thesis, thank you so much. Also thanks to my uruguayan friends at Paris, Montevideo, and elsewhere; even at the distance, I always received from you the positive energy and the sincere advice, so necessaries to reach destination. Special thanks to Edu, Pao, Fede, Rafa, Chupete, y a la Neumo.

I owe my deepest gratitude to my beloved mother Olga Hernández, my beloved second mother Alba Vicente, my dear sister Cecilia Casas, and to the rest of my so lovely family, your love and unconditional support give a meaning to my achievements.

Last but not least, and this is one of those cases where the "not least" part is a topmost, thanks to my wife Jimena Saporiti, the source of light in my life.

This Ph.D. thesis has been partially funded by the French program CARNOT. I thank the Scientific Direction (DS) and the Computer Science department of Télécom Bretagne for funding the presentation of my works at international conferences, as well as the Universidad de la República for funding my research activities in Montevideo.

Abstract

Network-wide traffic analysis and monitoring in large-scale networks is a challenging and expensive task. In this thesis work we have proposed to analyze the traffic of a large-scale IP network from aggregated traffic measurements, reducing measurement overheads and simplifying implementation issues. We have provided contributions in three different networking fields related to network-wide traffic analysis and monitoring in large-scale IP networks.

The first contribution regards Traffic Matrix (TM) modeling and estimation, where we have proposed new statistical models and new estimation methods to analyze the Origin-Destination (OD) flows of a large-scale TM from easily available link traffic measurements.

The second contribution regards the detection and localization of volume anomalies in the TM, where we have introduced novel methods with solid optimality properties that outperform current well-known techniques for network-wide anomaly detection proposed so far in the literature.

The last contribution regards the optimization of the routing configuration in large-scale IP networks, particularly when the traffic is highly variable and difficult to predict. Using the notions of Robust Routing Optimization we have proposed new approaches for Quality of Service provisioning under highly variable and uncertain traffic scenarios.

In order to provide strong evidence on the relevance of our contributions, all the methods proposed in this thesis work were validated using real traffic data from different operational networks. Additionally, their performance was compared against well-known works in each field, showing outperforming results in most cases. Taking together the ensemble of developed TM models, the optimal network-wide anomaly detection and localization methods, and the routing optimization algorithms, this thesis work offers a complete solution for network operators to efficiently monitor large-scale IP networks from aggregated traffic measurements and to provide accurate QoS-based performance, even in the event of volume traffic anomalies.

Keywords: Traffic Matrix, Network-Wide Traffic Monitoring, Traffic Modeling and Estimation, Optimal Volume Anomaly Detection and Localization, Proactive Traffic Management, Robust Routing, Dynamic Load Balancing, Reactive Robust Load Balancing, Quality of Service.

Contents

Acknowledgments **3**

Abstract **5**

1 Introduction **19**

1.1 Contributions of the Thesis 27

1.2 Outline of the Thesis 30

2 Traffic Matrix Modeling and Estimation **33**

2.1 State of the Art 36

2.2 Traditional TME: Gravity and Tomo-Gravity Methods 39

2.2.1 Background Concepts and Terminology 39

2.2.2 Gravity and Tomo-Gravity TME methods 40

2.3 Parsimonious TM Modeling and TME 43

2.3.1 Validation of the SB Model and SMLE TM Estimation 46

2.4 Recursive Traffic Matrix Estimation 51

2.4.1 A Simple State-Space Model for the Traffic Matrix 51

2.4.2 Drawbacks of the Former State-Space Model 52

2.4.3 State-Space model for centered TM variations: static mean . . . 54

2.4.4 Extending the model: dynamic mean 55

2.4.5 Evaluation of the RKFE TME method 56

2.5 The Random Neural Network for TME 59

2.5.1 The Random Neural Network Model 59

2.5.2 Learning in the Random Neural Network 61

2.5.3 Using the RNN Model for TM Estimation 68

2.5.4 Stability of RNNs vs ANNs for TME 70

2.5.5 Evaluation of the RNN-TME method 71

2.6 Principal Components Analysis for TME . . . 74
2.7 Comparative Analysis . . . 78
2.7.1 TME for Normal-Operation Traffic . . . 78
2.7.2 TME in the Presence of Volume Anomalies . . . 80
2.7.3 Numerical Complexity . . . 82
2.8 Conclusions . . . 84

3 Optimal Anomaly Detection and Localization 87
3.1 State of the Art . . . 90
3.2 Network Volume Anomalies Taxonomy . . . 92
3.3 Optimal Volume Anomaly Detection . . . 94
3.4 Optimal Sequential Volume Anomaly Detection and Localization . . . 96
3.5 Benchmarking Methods for Anomaly Detection and Localization . . . 100
3.5.1 Anomaly Detection and Localization with PCA and the Sub-Space Method . . . 100
3.5.2 Anomaly Detection with Kalman Filters . . . 102
3.6 Performance Evaluation . . . 104
3.6.1 Numerical Evaluation of the OSBD Method . . . 104
3.6.2 Limitations of PCA for Anomaly Detection . . . 106
3.6.3 Evaluation of the SSB Detection/Localization Method . . . 108
3.7 Discussion . . . 113
3.7.1 Complexity Analysis . . . 113
3.7.2 Implementation Issues . . . 115
3.7.3 Multiple Anomaly Localization . . . 118
3.8 Conclusions . . . 119

4 Routing Optimization Under Traffic Uncertainty 121
4.1 State of the Art . . . 126
4.2 Prediction-Based Routing and Robust Routing Optimization . . . 129
4.2.1 Routing Optimization for Instantaneous Traffic . . . 134
4.2.2 Robust Routing and Prediction-Based Routing with Time-Varying TMs . . . 136
4.3 Multi-Hour Robust Routing . . . 140
4.3.1 Multi-Hour Robust Routing Evaluation . . . 142
4.4 Reactive Robust Routing . . . 145
4.4.1 The Reactive Robust Routing Algorithm . . . 145
4.4.2 Back to Normal Operation . . . 148

4.4.3 The Complete Reactive Robust Routing Algorithm 148
4.4.4 Partial Robust Routing Reconfiguration and Reactive Robust Load Balancing 150
4.4.5 Reactive Robust Routing Evaluation 151
4.5 QoS in Robust Routing: Improving Network-Wide Performance 155
4.5.1 Improving Network-Wide Performance and E2E QoS 158
4.5.2 Comparison between RRMP and RRAP 159
4.6 Reactive Robust Load Balancing vs Dynamic Load Balancing 162
4.6.1 Dynamic Load Balancing 162
4.6.2 A Preliminary Comparison 164
4.6.3 Improving Dynamic Load Balancing 166
4.6.4 Evaluation and Discussion 169
4.7 Conclusions 172

Conclusions and Perspectives **173**

List of Publications **183**

Distinctions of the Thesis **185**

Bibliography **187**

Thesis Summary: French **197**

Thesis Summary: Spanish **213**

List of Figures

1.1 Internet traffic volume and FTTH deployment in the near-future. . . . 20

1.2 IPv4 Internet topology map in january 2009. 21

1.3 Different levels of traffic aggregation. 23

1.4 Intradomain Traffic Matrix monitoring. 27

1.5 Outline diagram of the thesis. 30

2.1 Components of an IP Network. 39

2.2 Approximation of real OD-flows traffic (dashed lines) by the Spline-Based model (full lines) in 3 operational networks. $\hat{x}_t^{SMLE}(k)$ stands for the estimated OD-flow k using the Spline-Based model, defined in equation (2.11). $\hat{x}_t^{TGE}(k)$ is the estimated OD-flow k using the Tomo-Gravity Estimation method. 44

2.3 (a) RRMSE(t) and (b) Cumulative RRMSE(t) for 672 measurements in the Abilene and the GEANT networks. 49

2.4 RRMSD(t) for 1500 OD-flows in a Tier-2 ISP network 49

2.5 QQ-plots for 2 residual processes from (a,c) Abilene and (b,d) GEANT. 50

2.6 (a) Single estimated OD-flow and (b) RRMSE(t) using RKFE for (1) model (2.14) and (2) model (2.25). 56

2.7 RRMSE(t) and Cumulative RRMSE(t) for 1 week of traffic in GEANT and Abilene, using (1) model (2.25) and (2) model (2.27) 57

2.8 Three-layers feed-forward RNN topology. 63

2.9 Block diagram of the TME method based on three-layers Feed-Forward (FF) Random Neural Networks. Each OD-flow volume $x_t(k)$ is estimated from link measurements using transfer-block $f_k(\cdot)$ in 2.9(b). Each of these blocks is built from a three-layers FF RNN like the one depicted in 2.9(a). The m blocks are applied in parallel to estimate a complete TM X_t in 2.9(c), using the connection blocks Δ and Π. 69

2.10 Cumulative distribution of the relative error as function of the mean number of hidden neurons $\bar{H}$ in the Abilene dataset, for (a) the RNN model and (b) the ANN model. 71

2.11 1 week of OD-flow traffic volume estimation using the RNN-TME approach for (a) an OD-flow in GEANT and (b) an OD-flow in Abilene. . 72

2.12 Cumulative distribution of the RRMSE(t) for 1 week of TMs estimated with the RNN-TME approach in GEANT and Abilene. 72

2.13 (a) Cumulative RRMSE(t) and (b) Cumulative SRRMSE(k) for 672 measurements in Abilene, for the SMLE, the RKFE, the TGE, the SGE, the RNN-TME and the PCAE methods. 79

2.14 Normalized OD-flow volume estimation under a large volume variation due to a BGP egress-point shift. 81

3.1 Network volume anomalies in large-scale IP networks. 88

3.2 Correct detection rate vs false alarm rate for the Optimal Spline-Based Detection method (OSBD - solid line) and the PCA approach, considering a different number of k first PCs $\mathbf{v}_k$ to model the normal sub-space.105

3.3 Temporal evolution of $||Y_{\text{residual}}||^2$, using a different number of first PCs to model $\mathcal{S}$. The squares indicate when an anomaly truly occurs. The dotted line depicts the detection threshold. Large anomalies pollute the normal sub-space and are not detected with the PCA approach. (a) Both large volume anomalies at samples 200 and 540 are correctly detected using 1 PC to describe $\mathcal{S}$. (b) Large volume anomalies are not detected using a 2 PCs representation of $\mathcal{S}$. 106

3.4 Temporal evolution of the total variance captured by each PC $\mathbf{v}_i$, $||\mathbf{Y}\mathbf{v}_i||^2$. Each time window $t_{j=1..10}$ consists of 6hs of SNMP data. Large volume anomalies may inadvertently pollute the normal subspace at t_3 and t_8. 107

3.5 Typical realizations of anomaly detection/localization functions in a Tier-2 ISP network. 109

3.6 Typical realizations of anomaly detection/localization functions in the Abilene network. 109

3.7 On-line volume anomaly detection and localization in Abilene, using the Sequential Spline-Based method. 110

4.1 The uncertainty set $\mathbb{X}$ as a polytope. 131

4.2 A combined columns and constraints generation method to solve the Robust Routing Optimization Problem. 133

4.3 Robustness of routing optimization facing Traffic Matrix estimation. . . 136

4.4 Robust Routing with Time Varying Traffic in Normal Operation. . . . 137

4.5 (a) Daily traffic link load, (b) Routing evaluation. 138

4.6 Daily variation of the polytope $\mathbb{X}_t$, (a) discrete-time $\mathbb{X}_t$, and (b) continuous time $\mathbb{X}_t$. (c) Time partitioning of $\mathbb{X}_t$. 141

4.7 Multi-busy-hour behavior in the traffic of two links in Abilene. 142

4.8 Routing performance, Stable vs. Multi-Hour Robust Routing. 143
4.9 Robust Routing Optimization Problems with Pre-established Paths P_k. 144
4.10 High-level description of the Reactive Robust Routing 146
4.11 Different anomaly polytopes for preemptive robust routing computation. 147
4.12 The Complete Reactive Robust Routing Algorithm. 149
4.13 Flow Diagram of the Complete Reactive Robust Routing Algorithm. . . 150
4.14 Reactive Robust Routing performance under a simulated single-flow volume anomaly. 152
4.15 Reactive Robust Load Balancing - load balancing after detection and isolation of a large volume anomaly in OD flow 13. 153
4.16 Mean queue size, measurements and approximations 156
4.17 (a) Maximum link utilization and (b) mean end-to-end queuing delay. Traffic demand volume abruptly increases after the 100th minute. (c) and (d) depict the corresponding boxplot performance summaries, relative to the optimal values. 157
4.18 (a,c) Maximum link utilization and (b,d) mean end-to-end queuing delay for RRMP and RRAP. The boxplot performance summaries are relative to the optimal values. 160
4.19 Maximum link utilization and mean end-to-end queuing delay. Traffic demand volume abruptly increases after the 100th minute. 165
4.20 Evolution of r_p^k for the anomalous OD pair (MinDG) 166
4.21 (a,c) Maximum link utilization and (b,d) mean end-to-end queuing delay for normal traffic variation. The boxplot performance summaries are relative to the optimal values. 170
4.22 (a,c) Maximum link utilization and (b,d) mean end-to-end queuing delay facing a volume anomaly. The boxplot performance summaries are relative to the optimal values. 171
4.23 Perspectives de croissance du trafic Internet et du déploiement de la technologie FTTH pour les prochaines années. 198
4.24 Topologie IPv4 de l'Internet en janvier 2009. 199
4.25 Différents niveaux d'agrégation de trafic réseau. 201
4.26 Surveillance de la Matrice de Trafic Intra-domaine. 206
4.27 Schéma structurel de la thèse. 211
4.28 Perspectivas del incremento de tráfico en Internet y del despliegue de la tecnología FTTH en los próximos años. 214
4.29 Mapa de la topología IPv4 de Internet en enero del 2009. 215
4.30 Diferentes niveles de agregación de tráfico. 218
4.31 Monitoreo de la Matriz de Tráfico Intra-dominio. 222
4.32 Esquema estructural de la tesis. 226

List of Tables

2.1 Network Topologies and Traffic Datasets. 46

2.2 Mean RRMSE values (%) for 672 TMs in Abilene. 78

2.3 Computational complexity of the different TME algorithms. The number of operations corresponds to the estimation of a complete TM with m OD-flows and r links, and it does not include the operations involved in the learning/calibration of the methods. 83

3.1 Different Network and Traffic Anomalies in IP Networks. 93

3.2 Results of the detection and localization for 864 SNMP measurements in Abilene, composed of 65 OD-flow volume anomalies. 111

3.3 Numerical complexity and memory usage for different on-line anomaly detection algorithms. On-line operations are divided into detection operations and localization operations. 115

3.4 Implementation issues in on-line anomaly detection/localization. 116

4.1 Routing performance under traffic uncertainty, relative to $u^*_{\max}$. 135

CHAPTER 1

Introduction

As the main enabler of our information era, Internet has become one of the main actors in our society. Internet is today the fundamental component of the world-wide communication infrastructure, playing a crucial role in education, entertainment, business, and social life. Its extraordinary and relentless growth around the world during the last decade has led to a burgeoning of companies generating, carrying, and sinking content in it. This proliferation of content has been followed by a sustained growth of starving Internet consumers, and nowadays Internet traffic is rapidly increasing, not only in volume but also in heterogeneity and complexity of composition.

After a brief mid-decade slowdown, major Internet actors forecast that Internet traffic will nearly double every two years in the very-near future, driven by high-definition video and high-speed access technology penetration. The overall IP traffic is expected to grow from 6.6 exabytes per month in 2007 to nearly 29 exabytes per month by 2011 (1 exabyte = 10^{18} bytes), more than quadrupling in less than a half decade [3, 4].

Simultaneously, the evolution of access technologies and the development of optical access networks, notably the Fiber To The Home technology (FTTH) will dramatically increase the bandwidth for end-users, imposing serious and unforeseen problems at the core network, so far assumed infinitely provisioned. The FTTH industry forecasts a demand of bandwidth per user as high as 30 Gbps in 2030 [1, 2]. Figure 4.28 depicts the prospective evolution of Internet traffic and ultra-high-speed access-bandwidth for the next couple of years.

This near-future scenario brings to light many challenging issues for network operators who are, after all, responsible for the networking support of the Internet growth. Internet users want a faster, higher-quality, more reliable, and more secure Internet, and traffic monitoring and analysis is probably the most efficient solution within reach for network operators. Knowing and understanding the traffic that flows through their networks is crucial for the health, the efficient design, and the engineering of network-services.

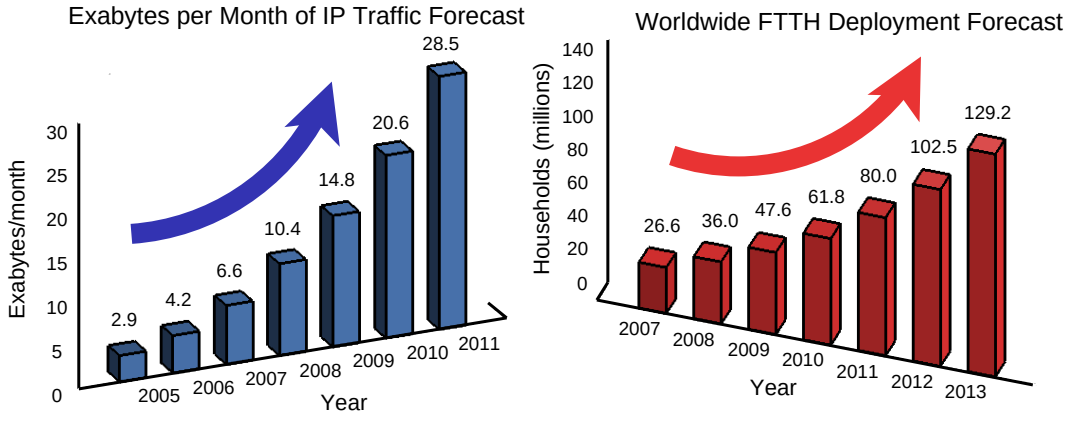


Figure 1.1 — Internet traffic volume and FTTH deployment in the near-future.

Traffic monitoring is certainly one of the paramount tasks for network operators that will be affected by the strong development of network traffic, simply because capturing and analyzing large volumes of heterogeneous traffic network-wide can be extremely costly. In the very first years of networking, network monitoring was more an art than a science, relying mainly on the expertise and know-how of network operators to analyze traffic "by hand". However, the increasing complexity and size of Internet Service Provider (ISP) networks as well as the heterogeneity and volume of network traffic has motivated the development of automatic and large-scale traffic monitoring systems in recent years.

Traffic monitoring can enhance different network management activities, such as network planing and design, traffic characterization and classification, failures or performance degradation identification, and even the detection of malicious traffic events. The last few years have seen an increasing emphasis on developing Intrusion Detection Systems (IDS), improving the ability of the network to handle malicious traffic.

The traffic monitoring process consists of three different consecutive tasks: the data collection, the data analysis, and the decision. Each of these tasks becomes increasingly challenging in current traffic scenario. Data collection is very costly, because there is too much data to gather in different parts of the network. Data analysis is more complex, because traffic is more heterogeneous and many different impairments can arise. The decision becomes more critical, because services provided in current Internet are more vital than in the past.

Another challenging issue related to network and traffic monitoring in current and future Internet scenarios is related to cost-effectiveness. Internet business models in the last five years have given rise to many Network Virtualization solutions [14], allowing small ISPs to capture part of the Internet market with very small infrastructure investments. This has encouraged network operators to reduce their investments in

networking infrastructure, looking for solutions that can get the most out of their current networks. Therefore, future monitoring systems should aim to build network-wide traffic analysis from limited measurements, relying on inference procedures and more intelligent algorithms.

Where to Monitor Traffic?

Despite the massive growth of the Internet, its global structure is still heavily hierarchical, with a core made of a reduced number of large Autonomous Systems (ASes) [5], known as Tier-1 networks. An AS is basically a collection of IP routing prefixes under the control of a single network operator, which share a common routing policy to the Internet [20]. A non-extensive list of current Tier-1 networks include AT&T, Global Crossing, Level 3 Communications, NTT Communications, Sprint, Tata Communications, Verizon Business (UUNET), Savvis, TeliaNet, Bell Canada, and XO Communications (XOXO). Figure 4.29 depicts a map of current Internet topology provided by CAIDA [127].

Tier-1 ISP networks provide global connectivity inside the Internet and represent the first level in the Internet hierarchy. The following levels of hierarchy are represented by smaller and less interconnected ASes known as Tier-2 and Tier-3 ISP networks, like Deutsche Telecom, British Telecom, and France Telecom among others. Finally, the edge of the topology is composed by terminal ASes, known as stub ASes.

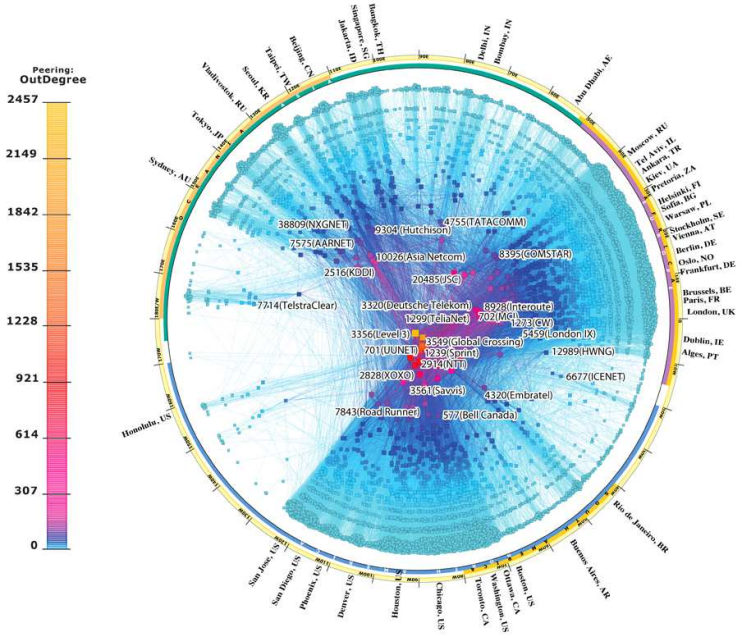


Figure 1.2 — IPv4 Internet topology map in january 2009.

This division in ASes provides two different structural views of the Internet: the intradomain Internet and interdomain Internet [125]. The intradomain Internet is composed of the interconnected routers within each single AS that exchange traffic among them, according to an intradomain routing protocol. Each AS has a different intradomain topology, perfectly known by its network operator. On the other hand, the interdomain Internet consists of the different ASes and their interconnections. The internal characteristics of each AS are transparent from an interdomain view, and traffic is exchanged between different ASes according to an interdomain routing protocol.

Traffic and network monitoring is usually performed at the intradomain level in every large-scale AS, generally Tier-1 and Tier-2 networks, mainly because the network topology is completely known and the AS is under the control of a single network operator, who can therefore manipulate his network and traffic without restrictions. Traffic monitoring at the interdomain level is more challenging, because the available information is much more restrictive, the different network operators do not necessarily cooperate with each other, there are privacy and economical issues that limits the exchange of information among network operators, and many other facets of interdomain that difficult the monitoring task.

In this thesis work we focused the attention on traffic monitoring and analysis at the intradomain level, for two main reasons. Firstly, the extensive control that we may have as regards intradomain permits to propose more complete and rich solutions, not only considering traffic monitoring and analysis, but also regarding the healing process of the network in the event of impairments. Secondly, the structure of the Internet is still concentrated in a small group of large-scale ASes, and thus the performance of the Internet as a whole highly depends on the individual performance of these networks.

What to Monitor?

Network operators are routinely confronted with a wide range of unusual events that threaten the proper operation of their networks. A significant problem when trying to detect these anomalous events is that their forms and causes can vary considerably. Network and traffic anomalies may arise from equipment failures, misconfigurations, and outages, unusual customers behavior (e.g., sudden changes in demand, flash crowds, high volume flows), external routing modifications, network attacks (e.g., DOS attacks, scans, worms), and even new previously unknown events.

An important challenge related to the detection of these events is that network and traffic anomalies are a moving target. It is difficult to precisely and permanently define the set of possible anomalies, especially in the case of malicious traffic. New network anomalies will continue to arise over time. Hence, anomaly detection systems should avoid being restricted to any predefined set of anomalies.

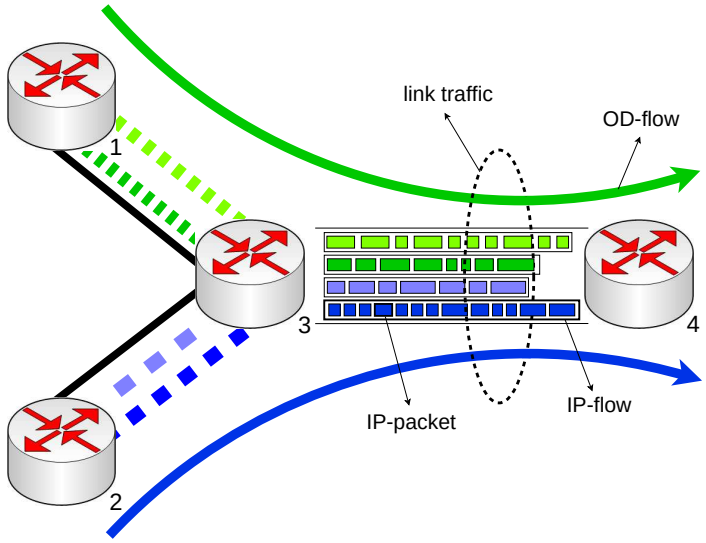


Figure 1.3 — Different levels of traffic aggregation.

Different kinds of network and traffic anomalies can be detected depending on the monitored traffic and its aggregation level. We shall consider four different levels of traffic aggregation: IP-packet, IP-flow, OD-flow, and link traffic. Figure 4.30 will help us to further explain this classification. Packet-level traffic analysis provides the most fine-grained and rich monitoring information. At this granularity it is possible to analyze the characteristics of each single IP-packet, accessing even its payload. Many IDS and traffic classification techniques and tools are developed at this traffic granularity [7, 8], using Deep Packet Inspection (DPI) techniques and packet sniffing tools [6]. Packet-level traffic analysis causes the highest measurement overhead and device exhaustion, and it is not a cost-effective or even an implementable solution for large-scale traffic monitoring.

IP-packets of similar characteristics can be grouped in a traffic flow. An IP-flow generally consists of a group of IP-packets that share the same 5-tuple, consisting in source and destination ports, source and destination IP addresses, and IP protocol. Figure 4.30 depicts four different individual IP-flows flowing between nodes 3 and 4. IP-flow analysis provides a better trade-off between traffic granularity and resources consumption than packet-level analysis, and many solutions for IP-flow-based monitoring have been developed in the last years, notably the well known NetFlow protocol [9]. Although initially implemented by Cisco, NetFlow is emerging as an IETF standard, the Internet Protocol Flow Information eXport (IPFIX) [10, 11], and many technology vendors currently add IPFIX support on their devices (e.g., Juniper, 3Com, Huawei, Alcatel-Lucent).

There are however different shortcomings when it comes to evaluate the performance of IP-flow-based traffic monitoring. To begin with, it requires additional dedicated technology for deployment in large-scale networks, including flow collectors, flow servers, and flow analyzers when using NetFlow. Maintaining IP-flow data can be

computationally expensive for routers and may burden a router's CPU or hardware to the point where it runs out of capacity. Additionally, exporting IP-flow records to a central server may result in a significant bandwidth reduction when monitoring large amounts of high-speed traffic [12]. These problems are further magnified when facing very heterogeneous traffic, simply because the number of IP-flow records may rapidly explode. To ease these problems, Cisco provides a variant known as "sampled NetFlow", where rather than looking at every packet to maintain IP-flow records, the router looks at every n-th packet.

However, sampled NetFlow has shortcomings that hinder the collection and analysis of traffic data [12, 13]. Firstly, selecting the right sampling rate is an inherently difficult task for the network manager, because no single rate gives the right trade-off between resources usage and measurement accuracy for all kinds of traffic. Traffic volumes measured with sampled IP-flows are an estimate rather than the actual measured flow volumes, which impacts the quality of the monitoring process. In addition, IP-flows reconstruction becomes a challenge when using sampling, because routers use simple time-bins and time-out based heuristics to record IP-flows.

IP-flows can be further aggregated into Origin-Destination (OD) flows. An OD-flow consists of all the IP-flows that share the same origin node and the same destination node. In figure 4.30, the four IP-flows previously described can be aggregated into two OD-flows, the former with origin in node 1 and destination in node 4, and the latter with origin in node 2 and destination in node 4. In order to construct OD-flows, the ingress and egress node of each IP-flow must be identified. This is generally achieved by routing data inspection [34, 124]. OD-flow aggregation yields a much smaller monitoring problem than using an IP-flow representation, but OD-flow-based monitoring still presents many of the shortcomings previously discussed, basically because the same measurement technology (i.e., NetFlow) is used to build OD-flow representations.

A network-wide view of OD-flows within a network is typically described by a Traffic Matrix (TM). A TM represents the total volume of traffic transmitted between every pair of ingress and egress nodes in a network. In practice, the term "volume of traffic" refers to the cumulative number of bytes observed between two consecutive measurements. In order to construct a complete TM, IP-flow measurement technology must be deployed at least in every ingress and egress node, with all the aforementioned shortcomings.

Given that the TM is a volume representation of traffic, the type of anomalies that can be detected from its analysis are volume anomalies. Volume anomalies represent large and sudden variations in OD-flows traffic. These variations are responsible for large changes in traffic characteristics, which may in turn significantly reduce the global QoS perceived by all the users of the network.

Finally, the most coarse-grained traffic granularity is represented by link-level aggregation. In figure 4.30, the two OD-flows share the same link between nodes 3 and 4. Link traffic refers to the total volume of traffic that flows between two nodes, physically connected by a network link. Link traffic volume can be easily collected using the widely spread Simple Network Management Protocol (SNMP). The SNMP protocol permits to collect management device readings from network devices, known as Management Information Base (MIB) variables. Every network device has a set of MIB variables that are specific to its functionality, like memory usage, CPU load, and interface bandwidth usage among others. In order to measure the total number of bytes through a network link, two MIB variables are generally used: the `ifInOctets` variable and the `ifOutOctets` variable. Both variables are counter-variables that simply cumulate the total bytes that have passed through a particular network interface. The volume of traffic provided by SNMP consists in the cumulative number of bytes observed between two consecutive polling intervals, which is simply the difference of the counter values between both consecutive measurements.

SNMP is unique in that it is supported by basically every device in an IP network, and it is readily available for traffic monitoring, without the need of additional measurement technology. In addition, SNMP-based monitoring is the technique that causes the least measurement overhead, and thus represents an appealing alternative for large-scale traffic monitoring. However, it also has practical limitations, like missing data due to the use of the unreliable UDP transport protocol to export readings, or lack of readings synchronization in large-scale networks.

In this thesis work we have decided to conduct traffic monitoring and analysis at the OD-flow level of traffic aggregation. We justify our decision for three main reasons: in the first place, OD-flow aggregation is fine enough so as to detect many of the impairments that threaten the health of large-scale networks [71], which are after all the support of the Internet itself. Secondly, it permits to conduct traffic monitoring in a network-wide scale, represented in practice by a TM. Finally, it is possible to design countermeasures with a global impact on the performance of the services supported by these large-scale networks.

In order to avoid the measurement problems associated with OD-flow analysis, we shall monitor the behavior of the TM from a higher level of traffic aggregation, using link-traffic SNMP measurements as the input data. The use of coarse-grained SNMP measurements permits to conceive light and easy-to-deploy large-scale monitoring algorithms, getting the most out of this simple and widely available technology. However, every traffic granularity has an associated cost to manage: the number of links in a network is generally much smaller than the number of OD-flows, and thus the TM is not directly observable from link-traffic SNMP measurements. The good thing is that this cost can be partially "refunded" by developing better statistical algorithms for network traffic modeling and analysis instead of using more expensive and complex technology.

Which Decision?

The first step in fixing a problem is knowing it existence. But what comes next? Network operators need not only to detect traffic anomalies but also locate their origins in order to take appropriate countermeasures. Countermeasures must rapidly reduce the negative impacts of traffic anomalies over the global performance of the network, as well as maintaining the integrity of the compromised services and data in case of network attacks. A complete network monitoring system should then help the network manager in detecting the presence of anomalous behaviors, locating their origins, and propose accurate countermeasures.

The application of countermeasures in large-scale networks is a difficult-to-automate decision process, basically because different kinds of anomalies require different countermeasures. In our context of TM monitoring, we are particularly interested in OD-flows traffic volume anomalies. The most important impacts of these kind of anomalies are the large and unexpected congestion problems that they may rise, which directly affects the overall performance of the network.

One possible countermeasure to fightback massive congestion problems is routing adaptation. The performance of every network depends in large part on the operation of the underlying routing protocols. Large IP networks usually combine protection and restoration mechanisms to minimize performance degradation in the event of network anomalies [57, 58], designing over-provisioned and redundant network topologies. However, the ever-increasing costs associated with robust-network designs have played an important role in determining the mechanisms that are currently used by network operators for recovery [56]. As an alternative, many large-scale network operators have opted for network restoration based on routing re-configuration and path re-computation [56].

In this thesis work we have explored a novel routing optimization paradigm, the Robust Routing (RR) optimization approach. RR permits to compute robust and efficient routing configurations to alleviate the impacts of volume anomalies on the global performance of the network. Different variants of RR have been proposed and analyzed in the thesis, including not only routing reconfiguration mechanisms but also load balancing techniques. These proposals not only help in reducing the congestion problems induced by volume anomalies, but also provide better resources utilization from a Quality of Service (QoS) perspective. This represents a paramount feature to maintain network services properly running, even in the event of traffic anomalies.

Figure 4.31 depicts the adopted context for the traffic analysis and monitoring problem that we tackle in the thesis. To sum-up, we propose to analyze network traffic in large-scale networks, detecting network-wide volume anomalies in the Traffic Matrix from coarse-grained SNMP measurements. Additionally, we propose to identify the origins of the detected volume anomalies, deploying accurate countermeasures based on robust routing reconfiguration and load-balancing mechanisms.

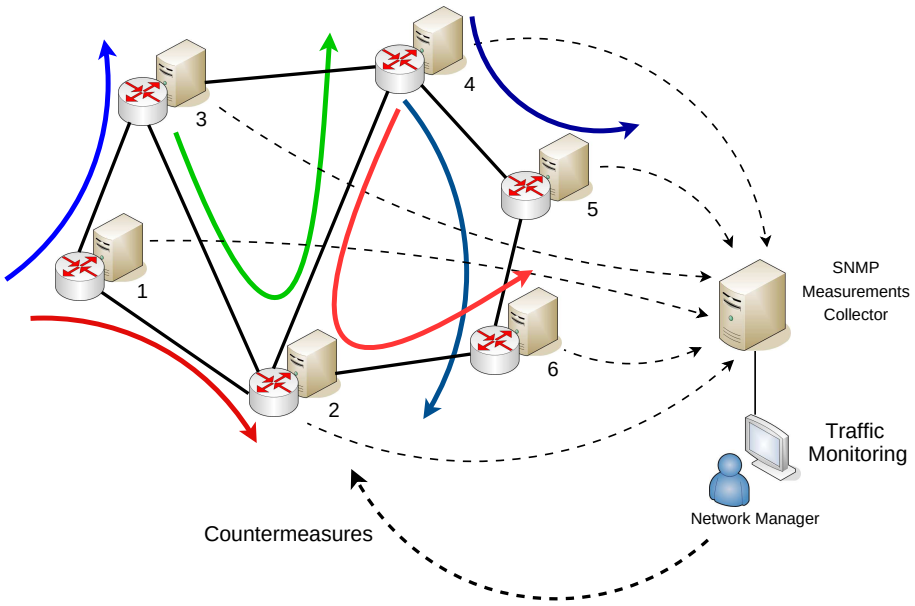


Figure 1.4 — Intradomain Traffic Matrix monitoring.

1.1 Contributions of the Thesis

For the multiple reasons previously presented, we believe that large-scale monitoring systems must aim to build network-wide views of traffic by collecting limited measurements and combining them with intelligent and efficient statistical analysis algorithms. In addition, these systems must be capable of rapidly and accurately detecting and locating traffic anomalies, responding with proper countermeasures that enable the network to maintain its functions with a reasonable performance level. A reliable implementation of such approach would be highly beneficial for network operators, providing a light and easy to deploy first-line traffic monitoring mechanism.

This thesis work provides strong contributions in three different networking fields related to network-wide traffic analysis and monitoring in large-scale IP networks: (i) Traffic Matrix modeling and estimation, (ii) Volume anomaly detection and localization from coarse-grained measurements, and (iii) Routing optimization under highly variable and uncertain network traffic. Despite the large literature available in these three fields, which is analyzed in chapters 2, 3, and 4, we shall evidence that to date there is no single approach to optimally detect and locate volume anomalies in the TM from link-traffic SNMP measurements, subsequently deploying routing countermeasures based on QoS provisioning.

The first major contribution regards Traffic Matrix modeling. We have developed a novel parametric, linear, and parsimonious traffic model to describe the anomaly-free behavior of OD-flows traffic in a large-scale IP network. This traffic model has several applications and advantages with respect to previously proposed TM models in the field. (i) Being parsimonious by conception, it permits to solve the TM observation

problems when using link-traffic SNMP measurements, allowing in particular to solve the well known TM Estimation (TME) problem, which is introduced in chapter 2. (ii) Contrary to many data-driven traffic models, ours is parametric and as we shall evidence it is remarkably stable in time, making it possible to design reliable anomaly detection methods on top of it. (iii) The model relies exclusively upon SNMP measurements to construct an accurate picture of the TM, simplifying practical issues. Finally and most importantly, (iv) this parsimonious linear model permits to remove the anomaly-free traffic from the anomaly detection problem, producing residuals sensitive to traffic anomalies. This a-priori simple feature has allowed us to construct optimal algorithms for volume anomaly detection and localization.

Our study in the TM modeling field has also produced interesting results in the TME field, where we have proposed several enhancements to previously introduced techniques, improving different conception drawbacks that were identified. In particular, we have proposed two TME techniques, the former based on recursive Kalman filters and the latter based on statistical learning techniques.

The second major contribution regards network-wide volume anomaly detection and localization in the TM, using link-traffic SNMP measurements. Based on the linear parsimonious traffic model previously described, we have proposed two different algorithms for volume anomaly detection and localization, with a paramount advantage with respect to previous proposals, that of presenting solid optimality properties. Optimality support is a feature generally absent in previous works, but it is fundamental in the conception of general algorithms, not tied to any particular network and more importantly, independent of individual evaluations in particular network and traffic scenarios. In-house methods may work rather well in certain scenarios, but without a principled and generalizable support they can be easily rebutted.

The first algorithm is designed for optimal volume anomaly detection, maximizing the correct detection rate for a bounded false alarm rate. The second algorithm permits to simultaneously detect and locate a particular anomalous OD-flow within the TM, minimizing the maximum mean detection/localization delay for given bounds in the false localization and false alarm rates.

The third major contribution regards intradomain routing optimization, reconfiguration, and load-balancing under highly variable traffic. Driven by the impressive ability to handle uncertain and variable traffic provided by the recently introduced Stable Robust Routing (SRR) paradigm, we have deeply explored its possible application to manage volume anomalies. Our studies revealed different shortcomings of SRR to efficiently handle large and abrupt traffic modifications, and different solutions were proposed. Firstly, we have proposed two routing reconfiguration variants for the former SRR approach, the former based on a multi-hour extension and the latter based on a reactive response to volume anomalies. Secondly, we have analyzed new optimization criteria to provide RR configurations with QoS properties. Finally, we

have explored the Dynamic Load-Balancing (DLB) paradigm, providing an in-depth comparative analysis between RR and different DLB mechanisms under highly variable traffic. To the best of our knowledge this was the first study that conducted such a comparative evaluation, necessary indeed for network operators who seek cost-effective and robust solutions to face abrupt traffic variations.

To provide strong evidence of the applicability of our contributions, all the proposed algorithms of the thesis were validated using real traffic data from different operational networks. Additionally, we have compared their performance against well-known works in each field, showing outperforming results in most cases.

To conclude, I would like to state that the different contributions of this thesis work are a result of different joint works carried out between 2006 and 2009 with many professors and researchers from various institutions. In particular, contributions relative to TM modeling and estimation, and contributions relative to volume anomaly detection and localization, are a result of joint works with Associate Professor Lionel Fillatre and Professor Igor Nikiforov (Université de Technologie de Troyes), and Professor Thierry Chonavel (Télécom Bretagne). Contributions relative to routing optimization and load-balancing are a result of joint works with Professor Walid Ben-Ameur (Télécom & Management SudParis), Assistant Professor Hervé Kerivin (Clemson University), Postdoctoral Research Associate Federico Larroca and Associate Professor Jean-Louis Rougier (Télécom ParisTech).

1.2 Outline of the Thesis

The presentation of this thesis work is organized in three different chapters. Figure 4.32 depicts the distribution of the thesis contributions and the interaction among the three chapters.

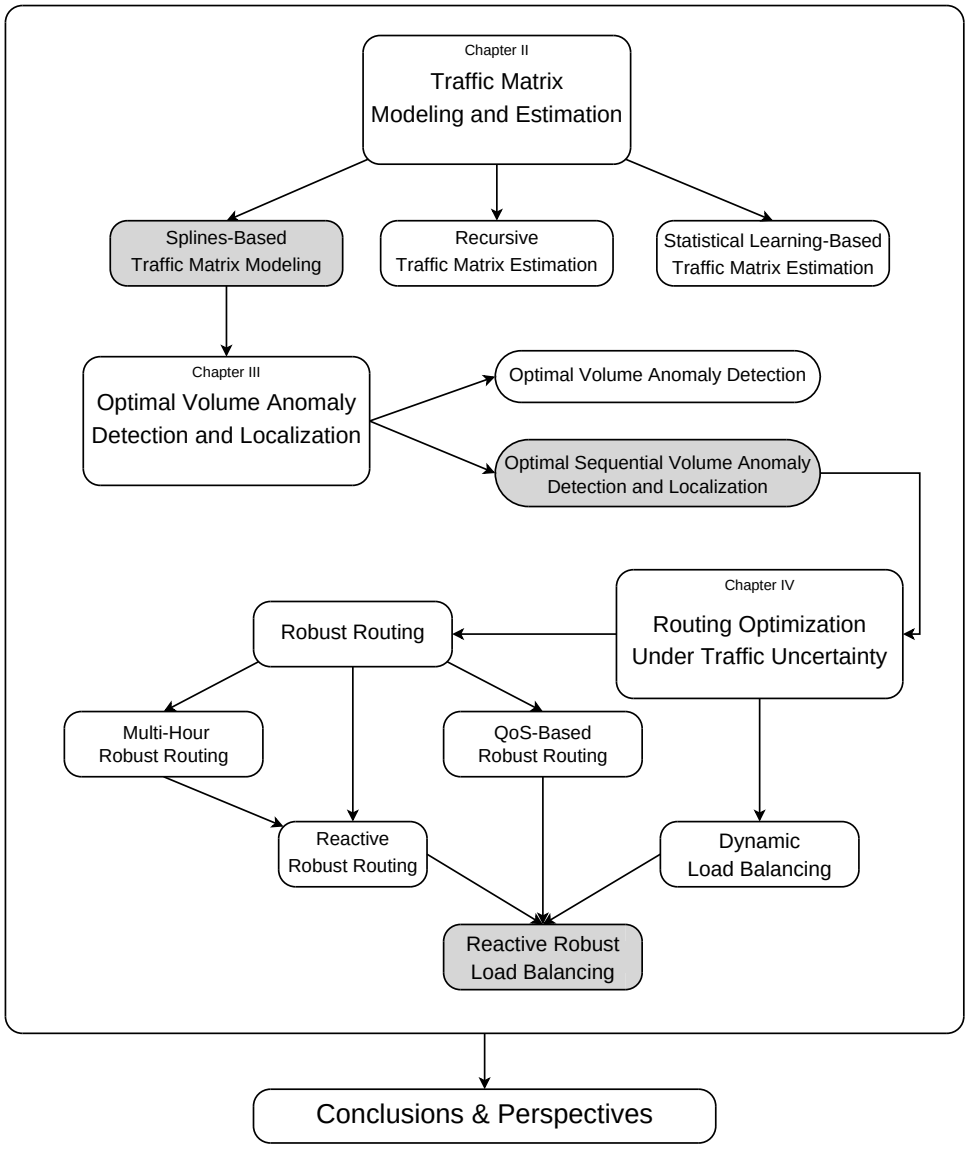


Figure 1.5 — Outline diagram of the thesis.

Chapter 2 presents our studies in Traffic Matrix modeling and estimation. Three models to analyze a complete large-scale TM from SNMP measurements are described and evaluated. The first of them consists in parsimonious polynomial-based modeling techniques, the second consists in state-space modeling and recursive filtering methods, and the third consists in statistical learning theory. In addition, we describe previously developed TM modeling and estimation techniques, which are further used as benchmark for our proposals.

Chapter 3 presents the design and evaluation of two optimal algorithms for volume anomaly detection and localization, using the principles of decision theory. Both algorithms rely on the parsimonious traffic model introduced and validated in chapter 2. Benchmarking algorithms are also described and analyzed in this chapter. Finally, a deep comparative evaluation between benchmark and our algorithms is conducted, considering not only detection and localization performance, but also complexity and implementation issues.

Chapter 4 presents the study of Robust Routing and Dynamic Load-Balancing paradigms. In this chapter, different variants and improvements to these paradigms are proposed and evaluated. A complete approach for QoS-based robust load-balancing is developed in this chapter, using the volume anomaly detection and localization algorithm introduced and validated in chapter 3.

Finally, the conclusions of the thesis work are presented, including various perspectives and clues for future work in the developed fields.

CHAPTER 2 Traffic Matrix Modeling and Estimation

The first step to conceive an anomaly detection algorithm is to develop accurate and stable traffic models to describe what constitutes an anomaly-free traffic behavior. This is indeed a critical step in the detection of anomalies, basically because a rough or unstable traffic model may completely spoil the correct detection performance and cause many false alarms.

In this thesis we have focused the attention on the detection of volume anomalies in the OD-flows of a Traffic Matrix (TM), and therefore, the traffic models that we have developed consider the anomaly-free traffic behavior at the TM scale. As we have previously explained, an additional constraint adds to our modeling problem, that of analyzing the behavior of OD-flows traffic from aggregated SNMP link-load measurements. In this sense, our modeling problem results in the traditional and well known Traffic Matrix Estimation (TME) problem [25].

Let us briefly introduce the TME problem. Assuming a network with m OD flows and r links, $X_t = [x_t(1), .., x_t(m)]^T$ represents the TM organized as a vector, where $x_t(k)$ stands for the volume of traffic of each OD flow $k = 1 \dots m$ at time t. The routing matrix $R \in \mathbb{R}^{r \times m}$ indicates which links are traversed by each OD-flow, being element R_{ij} equal to 1 if OD flow j takes link i and 0 otherwise. Finally, the vector $Y_t = [y_t(1), .., y_t(l)]^T$ represents the SNMP measurements vector, where $y_t(i)$ represents the total aggregated volume of traffic from those OD-flows that traverse link $i = 1 \dots r$ at time t. The relation between X_t, R, and Y_t can be described by a system of linear equations in the form of:

$$Y_t = R\, X_t \tag{2.1}$$

The TME problem basically consists in the invertion of (2.1), estimating the value of X_t from R and Y_t. We use the term "estimating" instead of "computing" because the number of unknown OD-flows is much larger than the number of links in the network, i.e., $r << m$ in (2.1), thus resulting in a massively under-constrained problem.

Different methods have been proposed in the last 10 years to tackle the TME problem. In general, methods developed before 2004 rely exclusively upon SNMP

measurements and routing information to estimate a TM, whereas newer methods additionally consider the availability of partial flow measurements used for calibration purposes. Therefore, we shall classify the former group of methods as *pure SNMP methods*, and the latter as *mixed methods*. Pure SNMP methods rely on modeling assumptions to infer a TM. Mixed methods additionally exploit temporal and spatial correlations among the multiple OD-flows of the TM to improve estimation results. These methods assume that, despite being expensive and resource consuming, direct flow measurements can be conducted during short periods of time. Mixed methods use then this rich but scarce data to extract OD-flows characteristics and to calibrate the underlying models during certain calibration or *learning* phase, previous to estimation.

In this chapter we present, evaluate and compare new TM models and new pure SNMP and mixed methods to estimate a TM from SNMP measurements. We begin by introducing a linear parametric model for the spatial distribution of OD-flows traffic within a TM, using very few parameters. Using this TM model, we build a simple yet accurate pure SNMP TME method. The principal virtue of this method is that it does not require direct flow measurements, neither to perform the estimation nor to calibrate the underlying model.

We introduce a novel mixed TME method based on statistical learning with neural networks. The method learns the relation between OD-flows traffic and SNMP measurements without assuming any particular model, using Random Neural Networks (RNNs). A RNN is a new kind of neural network, introduced in recent years by E. Gelenbe in [46]. As it has been shown in many previous works [40, 41, 42, 43, 44], RNNs are a very powerful tool to capture the intrinsic model behind the learning data.

We also present a recursive mixed TME method, based on the use of Kalman Filters. This method consists in an improved version of the recursive estimation approach presented in [22]. By introducing a new simple dynamic model for OD-flows we show that the performance of the approach can be improved, regarding both accuracy and stability.

Using the real network topologies and real traffic measurements from different operational networks, we compare these new methods to some of the most well known and accepted pure SNMP and mixed TME estimation techniques in the field, generally used as benchmark. The performance of all methods is evaluated in different traffic scenarios, including both anomaly-free and anomalous traffic variations. Our results show that it is still possible to improve the field of Traffic Matrix Estimation, encouraging the development and implementation of new techniques to lighten routers tasks in the future.

The remainder of this chapter is organized as follows. Section 2.1 presents the State of the Art in the field of Traffic Matrix Estimation. In section 2.2 we describe two traditional pure SNMP methods, the Gravity and the Tomo-Gravity TME methods.

The Tomo-Gravity TME method is a widely accepted method to estimate OD-flows volume from SNMP measurements and routing/topology information with confident results, and thus it is generally used as benchmark. In section 2.3 we introduce a novel parametric, linear, and parsimonious model for anomaly-free OD-flows traffic, and build a pure SNMP TME method based on this traffic model. Both the model and the TME method are validated and evaluated in this section. In section 2.4 we present and analyze different OD-flow traffic models for recursive estimation of the TM, using a Kalman filter approach. We analyze in depth the drawbacks and omissions in the original proposal of this recursive mixed TME method, introducing some modifications to improve the technique. These improvements are further verified with real traffic measurements. Section 2.5 introduces a new mixed TM estimation technique based on Random Neural Networks. Since RNNs are quite novel, even in the statistical learning domain, we present a detailed description of the general algorithm. Using real traffic data, we evaluate the method and evidence the virtues of RNNs w.r.t. classical Neural Network models when applied to the TME problem. In section 2.6 we describe another TME method used as benchmark, this time a mixed method based on Principal Components Analysis [37]. A comparative analysis of our TME methods w.r.t. the benchmark techniques is presented in section 2.7, considering both their accuracy and numerical complexity. Finally, section 2.8 concludes this chapter.

2.1 State of the Art

The problem of inferring a complete Traffic Matrix from links aggregated traffic data has been extensively studied during the past 10 years. The first approach to tackle the problem was to search for direct solutions to the ill-posed problem (2.1), introducing additional information to create additional constraints. This was achieved by simple TM modeling assumptions in [25, 27], deriving higher order statistics of the OD-flows traffic as the additional constraints. For instance, Vardi assumes a Poisson model for OD-flows in [25], using the covariances of the links traffic as the additional constraints. OD-flows volumes are then estimated by Maximum Likelihood (ML) estimation. The Poisson model is also used by Tebaldi et al. in [26], but rather than using a ML estimation they use a Bayesian approach. Since posterior distributions are hard to calculate, authors use a Markov Chain Monte Carlo (MCMC) approach to simulate the posterior distribution. Cao et al. [27] generalize the ML approach by assuming a Gaussian traffic distribution, considering that the variance is related to the mean through a power-law. Bermolen et al. derive in [29] the Cramér-Rao lower bound for the variance of the ML estimator. Additionaly, authors propose in [30] fast methods to estimate the TM under the same assumptions. Medina et al. [33] showed that the basic assumptions underlying these statistical models were not always justified in real TMs from operational networks, and that some of these methods performed badly when the underlying assumptions were violated.

The Bayesian approach was refined by Vaton et al. in [28], where authors proposed an iterative method to improve the prior distribution of OD-flows. This algorithm is made up of two different blocks that exchange probabilistic information iteratively between them. The first block uses MCMC methods, more precisely a Metropolis within Gibbs algorithm, as introduced by Tebaldi et al. in [26]. At each time slot, the inputs are the SNMP measurements and a prior distribution for each OD-flow, in the form of a weighted mixture of Gaussian distributions. The output is a multivariate time series that converges, in distribution, to the posterior distribution of the OD-flow. The common principle to any MCMC method (Hastings Metropolis, Gibbs, etc...) is to produce an ergodic Markov chain, which steady state distribution is the so called "target" distribution. For each OD-flow, the second block takes as input a time series of the successive OD-flow estimated values. These are averaged values that have been produced by running the first block algorithm for each measurement time slot. This chronological time series is considered to be a Markov modulated Gaussian process. Then, using standard inference methods in the framework of Markov modulated process, namely the EM algorithm, the "underlying" Markov state is estimated, namely the probability of each state is estimated for each OD-flow and each time slot. Parameters of the Markov modulated Gaussian process such as means and variances of the Gaussian components, as well as transition probabilities of the Markov chain are also re-estimated. These parameters as well as the estimate of the Markov modulating state for each OD-flow and each time slot are then transmitted to the first block in a feedback loop. In order to force this iterative process to converge, a smoothing parameter of the transfered information is also introduced. The purpose

of this smoothing parameter is basically to reduce/increase the effect of the exchanged information depending on its "accuracy"; the prior distributions exchanged at the beginning are rather smooth (large entropy value), whereas in the last iterations the priors are strong (lower entropy value). It is worth noting that this algorithm performs very well in the case of bursty trafic data (for example, trafic on a Local Area Network) since the Markov modulated assumption is observed into practice for this kind of data. Authors validate this algorithm on simulated traffic, but also using real traffic measured on a single router network in [31]. An interesting State of the Art about Bayesian methos for TME (MCMC methods, EM algorithm, etc.) is provided in [32].

Additional spatial information about the TM was included into the problem, taking into account the network topology and the routing process. This encouraged the application of Gravity models [35] to the TM estimation issue [36]. Zhang et al [21] made a breakthrough in the TME problem, by combining network tomography methods [25] with Gravity models to highly improve accuracy and reduce computational complexity. This method is the well-known Tomo-Gravity Estimation (TGE) approach. The TGE method was the first pure SNMP method to provide a successful estimation of the TM in real operation circumstances, being used by a major ISP as AT&T. However, the error rates produced by the method were not sufficiently low for critical real-time tasks such as on-line traffic monitoring, which motivated the development of more accurate techniques.

A new step was achieved by considering the strong diurnal patterns found in the TM [37] into the TME problem, together with a new strong assumption not considered before: the TM can be directly measured during short periods of time. By 2004, the advances in flow monitoring techniques permitted to include more rich data into the TME problem. Different mixed methods were proposed in 2004 and 2005 that exploited these assumptions [22, 23, 38]. In [38] authors proposed a pure data-driven method to estimate the TM, based on the stability of the node *fanouts*. The fanout for a source node is simply the fraction of its total traffic that is sent to a given destination. Authors in [37] proposed another data-driven approach to analyze OD-flows, using a Principal Component Analysis (PCA) method to capture spatial correlations between flows. The last contribution was proposed in [22, 78], where a dynamic model was adopted to capture the temporal correlation of the TM, using a Kalman filter approach to recursively estimate the TM. These methods make use of direct OD-flow measurements for calibration purposes. Although they seem quite accurate and they improve previous proposals, results presented in [23, 22, 76] showed that they can be unstable and several recalibration steps should be conducted in order to provide reliable results.

New mixed methods have emerged during the last couple of years to accurately estimate the TM. From those methods, we highlight the one that inspired our statistical learning-based algorithm. In [39], an Artificial Neural Network (ANN) model was used to learn the relation between links and OD-flows traffic. This method is interesting but

presents a major conception drawback: statistical learning with ANNs provides results which are very sensitive to the particular definition of the neural network topology [44, 45] and cannot therefore be easily generalized. This turns current implementation of the method highly unstable and difficult to calibrate, and thus difficult to apply in a real scenario.

2.2 Traditional TME: Gravity and Tomo-Gravity Methods

From the several pure SNMP TME methods developed in the past, the most celebrated method is by far the Tomo-Gravity Estimation (TGE) method, developed by the research team of AT&T Labs and introduced in 2003 by Zhang et al [21]. The TGE method is based on a simple model for traffic demands, known as the gravity model [35, 36]. The gravity approach was successfully applied in telecommunications to model the telephone traffic exchanged between area codes in [35], and was later used for backbone traffic demands in [21, 36].

Gravity models, taking their name from Newton's gravitation law, are commonly used to model the movement of people, goods or information between geographic areas. In a geographic gravity model for cities, for example, the relative strength of the interaction between two cities is proportional to the product of the populations divided by the squared distance between both.

Before going into the particular details of the gravity model applied to backbone networks, we shall briefly introduce some basic network concepts and terminology that will help us in the specification of the Gravity and Tomo-Gravity estimation methods.

2.2.1 Background Concepts and Terminology

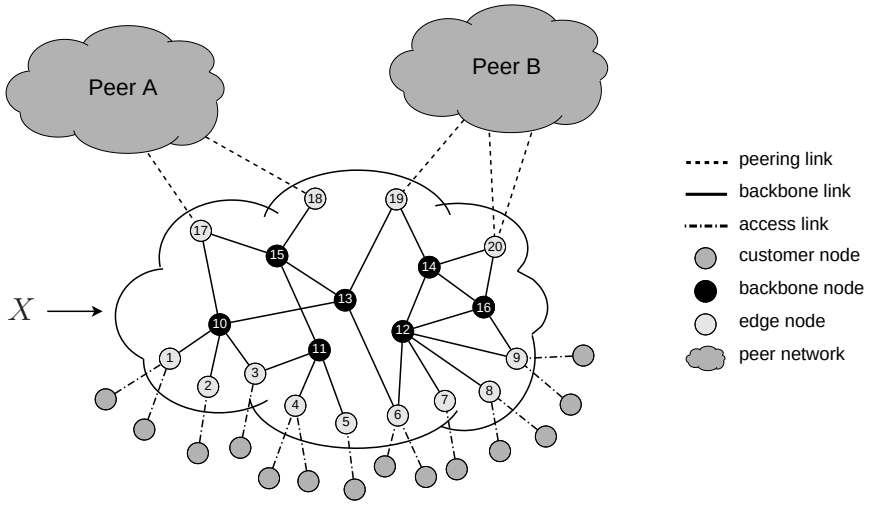


Figure 2.1 — Components of an IP Network.

An IP network is made up of an interconnection of IP routers within a single Autonomous System (AS) or administrative domain. As shown in figure 2.1, the network can be seen as a set of nodes and links, associated with routers and their interconnections. We shall refer to those nodes and links that are wholly internal to the network as *backbone* nodes and links.

Every IP network is generally connected to other ASes and customers via *edge* links. Edge links can be classified as *access* links, connecting customers, and *peering* links, which connect other ASes or *peers*. The ingress and egress backbone nodes of an IP network are further classified as *edge* nodes.

A significant fraction of the traffic in an IP network is interdomain traffic, which is exchanged between customers and peer networks. Traffic to peer networks is today largely focused on dedicated peering links. Under the typical routing policies implemented by large ISPs, very little traffic will transit the backbone from one peer network to another under normal operating circumstances. Transit traffic between peers may reflect a temporary step in network consolidation following an ISP merger or acquisition, but should not occur in general.

The Traffic Matrix X that we consider throughout the thesis represents an intradomain view of traffic, and it express the total volume of traffic exchanged between pairs of backbone nodes, both edge and internal. In figure 2.1, the TM reflects the volume of traffic exchanged among the 20 numbered nodes that lie inside the cloud, representing the AS under analysis.

2.2.2 Gravity and Tomo-Gravity TME methods

In the simplest form of the Gravity model for backbone traffic proposed in [36], authors compute the total traffic volume entering at edge node i, namely $x(i,*)$, and the total traffic volume leaving from each edge node j, namely $x(*,j)$. The value $x(i,*)$ corresponds to the incoming traffic on all the access and peering links. Likewise, $x(*,j)$ includes all the traffic leaving the AS on either access or peering links. Note that for ease of exposition, we have omitted the time index t in the notation. Using these quantities, the gravity model simply states that the traffic exchanged between an origin node i and a destination node j is proportional to the traffic volumes entering at node i and exiting at node j:

$$x(i,j) = x(i,*)\frac{x(*,j)}{\sum_j x(*,j)} \tag{2.2}$$

This Simple Gravity Estimation (SGE) method provides a quite rough view of the TM, but as shown in [21], its accuracy can be improved with some additional information regarding link classification and routing policies. The simple gravity model essentially assumes complete independence between sources and destinations. However, routing policies applied by ISPs treat customer and peering traffic differently, leading to deviations from pure independence. The first of these policies is known as the hot-potato routing, which basically states that traffic from a customer node traveling towards a peer will be sent to the nearest exit node. The second policy regards the traffic transiting the network from one peer to another, which should be zero as we have previously explained.

To capture these policies, authors in [21] propose to separately treat the traffic exchanged among customers and peers, identifying link types. Broadly speaking, they define two sets of edge links: the set $\mathcal{A}$ of access links, and the set $\mathcal{P}$ of peering links. In the reference IP network of figure 2.1, the set $\mathcal{A}$ includes all edge links connected to nodes 1 to 9, and the set $\mathcal{P}$ all edge links connected to nodes 17 to 20. This analysis leads to a generalization of the Simple Gravity Estimation method, known as the Generalized Gravity Estimation (GGE) method, which can be defined as follows:

$$x(i,j) = \begin{cases} 0 & \forall i \in \mathcal{P}, j \in \mathcal{P} \\ \frac{x(i,*)}{\sum_{i \in \mathcal{A}} x(i,*)} x(*,j) & \forall i \in \mathcal{A}, j \in \mathcal{P} \\ x(i,*) \frac{x(*,j)}{\sum_{j \in \mathcal{A}} x(*,j)} & \forall i \in \mathcal{P}, j \in \mathcal{A} \\ \rho \frac{x(i,*)}{\sum_{i \in \mathcal{A}} x(i,*)} x(*,j) & \forall i \in \mathcal{A}, j \in \mathcal{A} \end{cases} \tag{2.3}$$

where ρ is a normalization constant. Note that in this case, the obtained TM has a finer level of resolution than (2.2), as $x(i,j)$ represents now link-to-link traffic rather than node-to-node traffic. However, it is very easy to obtain a backbone node-to-node TM from this link-to-link TM, using routing information [19]. There are some additional details about the Generalized Gravity (GG) model that we prefer to omit in favor of brevity and ease of comprehension, but we refer the interested reader to [21] for them.

The Generalized Gravity (GG) model is simple and improves estimation results, but it has a significant drawback regarding the estimation of a complete TM: the model solution is guaranteed to be consistent with measured link loads at the network edge, but not in the interior links. Thus, the relation between links load traffic and OD-flows traffic defined in (2.1) is not necessarily verified.

To remedy this problem, authors in [21] propose to combine the GG model with a least mean squares approach, *refining* the estimated TM subject to the constraints imposed by internal link measurements (2.1). The idea of this approach, known as the Tomo-Gravity Estimation (TGE) method, is to pick the closest TM to an initial GG estimation $\hat{X}^{GGE}$, among all the TMs that satisfy (2.1). This optimization problem can be formulated as a quadratic program:

$$\begin{aligned} \min_{X} \quad & ||X - \hat{X}^{GGE}|| \\ \text{s.t.} \quad & Y = RX \end{aligned}$$

where $||.||$ is the L_2 norm of a vector (i.e., the Euclidean distance to the origin). Briefly speaking, the TGE solution is nothing but the euclidean projection of $\hat{X}^{GGE}$ on the space defined by $Y = RX$. Given that constraints are ill-posed, authors propose

to use a Singular Value Decomposition (SVD) of the routing matrix R to compute its pseudo-inverse R_{inv}, using this pseudo-inverse matrix to compute an *additive correction factor* X' to refine the initial GG estimate $\hat{X}^{GGE}$. The final algorithm produces a Tomo-Gravity estimate $\hat{X}^{TGE}$ from a routing matrix R, a SNMP measurements vector Y, and an initial GG estimate $\hat{X}^{GGE}$:

$$\begin{cases} Y' &= Y - R\hat{X}^{GGE} \\ R_{\text{inv}} &= \text{pinv}(R) \\ X' &= R_{\text{inv}}Y' \end{cases}$$

$$\boxed{\hat{X}^{TGE} = \hat{X}^{GGE} + R_{\text{inv}}\left(Y - R\hat{X}^{GGE}\right)} \tag{2.4}$$

As the algorithm uses a pseudo-inverse matrix to compute the correction factor X', the application of the least mean squares algorithm in the TGE method may result in negative values for certain OD-flows of the TM. This problem can be avoided by treating the problem as a positively constrained optimization problem, adding the constraint $\hat{X}^{TGE} \geqslant 0$. However, and as suggested in [27], authors in [21] use a simple iterative procedure known as the Iterative Proportional Fitting Procedure (IPFP) to ensure non-negativity of the final estimate $\hat{X}_t^{TGE}$ at a low computational cost.

2.3 Parsimonious TM Modeling and TME

In this section we develop a parametric linear model for the TM, which can correctly reflect the total volume of traffic exchanged among all the backbone nodes of the network (both edge and interior nodes) when OD-flows traffic behavior is free of anomalies. The model has a paramount property, that of being parsimonious in its structure, which basically means that the TM can be described with a small number of parameters. This property permits to easily solve the TME problem, and thus we design an estimation method based on this model.

The basic idea of the model is that OD-flows traffic X_t, sorted by volume, can be decomposed at each time t over a known family of q basis functions $S = \{\mathbf{s}(1), \mathbf{s}(2), \ldots, \mathbf{s}(q)\}$, with the great virtue that $q << m$, even several orders of magnitude smaller (in the evaluation of the model, we show that $q < 10$ even for a network with more than $m > 1000$ OD-flows). Therefore, we assume that X_t can be expressed as:

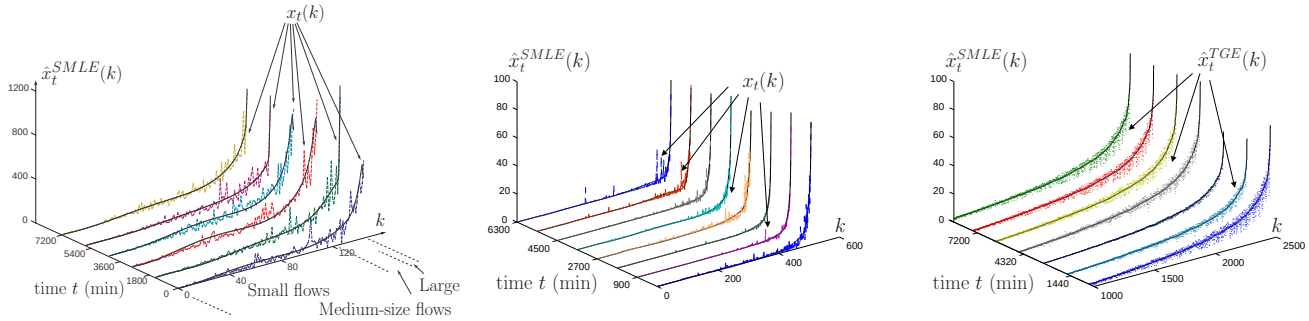
$$X_t = S\boldsymbol{\mu}_t + \boldsymbol{\xi}_t \tag{2.5}$$

where $\boldsymbol{\xi}_t$ is a white Gaussian noise with covariance matrix $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$ that models the natural variability of the TM together with the modeling errors. The vector $\boldsymbol{\mu}_t = \{\mu_t(1) \ldots \mu_t(q)\}^T$ is the unknown time varying parameters vector which describes the OD-flows traffic volume distribution w.r.t. the set of vectors $\mathbf{s}(i)$.

In our investigations, we found that the order of increasing OD-flows traffic w.r.t. their volumes remains stable in time for several days. We attribute this stationarity on the spatial distribution of traffic to two different but related phenomena. The former concerns geographic locality, the latter is the well-known mice and elephants phenomenon.

While geographic locality is not the determinant routing factor in today's Internet as compared to ISP routing policies, it is clear that the volume of traffic that flows between major Points of Presence (PoPs) nodes has an underlying origin that remains stable in time: people. The traffic exchanged between two major cities or countries tends to be stable w.r.t. the traffic exchanged between two other cities or countries, simply because the people that generates this traffic do not move from one to another. As regards the mice and elephants phenomenon, it is well-known that a small percentage of OD-flows contribute to a large proportion of the total traffic in every large-scale IP network [103, 33]. The existence of such dominant OD-flows together with the geographic locality issue makes reasonable to assume that, in the absence of anomalies, the OD-flows with the largest volume in a network remain the largest, and the smallest OD-flows remain the smallest during long periods of time.

It should be clear to the reader that this model can not be generalized to all network topologies and scenarios, but that it holds for networks with a high level



(a) Abilene Network (b) GEANT Network (c) Tier-2 ISP Network

Figure 2.2 — Approximation of real OD-flows traffic (dashed lines) by the Spline-Based model (full lines) in 3 operational networks. $\hat{x}_t^{SMLE}(k)$ stands for the estimated OD-flow k using the Spline-Based model, defined in equation (2.11). $\hat{x}_t^{TGE}(k)$ is the estimated OD-flow k using the Tomo-Gravity Estimation method.

of traffic aggregation, like backbone networks, and networks that do not present a noticeable multi-busy-hour behavior.

Figure 2.2 shows the OD-flows traffic for three different IP networks, sorted from smallest to largest volumes and for different times t. The three networks are (a) the Abilene network, an Internet2 backbone network at the US, (b) the GEANT network, a European large-scale research network, and (c) a private commercial Tier-2 ISP network.

The sorted volumes of OD-flows traffic can be approximated by a non-decreasing function with a certain smoothness. The curve obtained by interpolating this function is parameterized by using a polynomial splines approximation. Given the shape of this curve, a cubic splines approximation is used. A discrete spline basis is finally designed to approximate the sorted volume of OD-flows traffic, discretizing the continuous splines according to m points uniformly chosen in the interval $[1; m]$. The vectors $\mathbf{s}(i)$ in S form the set of discrete spline basis vectors, which describe the spatial distribution of traffic. From now on, we shall refer to this traffic model as the Spline-Based (SB) model.

To illustrate the structure of matrix S, let us consider a polynomial splines of degree $p = 3$, with $p - 1$ continuous derivatives and two integer knots k_1 and k_2 such that $1 < k_1 < k_2 < m$. A natural cubic spline $c(x)$ with two knots k_1 and k_2 has the form:

$$c(x) \;=\; \mu(1) + \mu(2)\, x + \mu(3)\, x^2 + \mu(4)\, x^3 + \mu(5)\, (x - k_1)_+^3 + \mu(6)\, (x - k_2)_+^3 \qquad (2.6)$$

where x belongs to a real interval $[a; b]$ containing $[1; m]$, i.e. $[1; m] \subseteq [a; b]$, the reals $\mu(i)$ are the spline coefficients and $(x)_+ = \max\{0, x\}$. The interested reader can find additional information on splines representations in [81]. Then, the sampled vector $\mathbf{c} = (c(k))_{1 \le k \le m}$ verifies $\mathbf{c} = V\boldsymbol{\mu}$, where the matrix V is given by:

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 2^2 & 2^3 & 0 & 0 \\ 1 & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \vdots & \vdots & \vdots & 0 & 0 \\ 1 & \vdots & \vdots & \vdots & 1 & 0 \\ 1 & \vdots & \vdots & \vdots & 2^3 & 0 \\ 1 & \vdots & \vdots & \vdots & \vdots & 1 \\ 1 & \vdots & \vdots & \vdots & \vdots & 2^3 \\ 1 & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & m & m^2 & m^3 & (m-k_1)^3 & (m-k_2)^3 \end{pmatrix} \tag{2.7}$$

The matrix S is obtained from the matrix V by permuting the rows according to the OD-flows sorting order: the i-th row of S is the j-th row of V, provided that the OD-flow i becomes the j-th OD-flow after sorting, from smallest to largest OD-flows traffic volumes.

Back to figure 2.2, the dashed lines depict the real value of each sorted OD-flow $x_t(k), k = 1 \dots m$, the full lines represent the polynomial approximation of the sorted flows provided by the SB model. In order to appreciate the time stability of this approximation, the curves are plotted for various consecutive days, at different times. Using (2.1) and (2.5), we can express the links traffic Y_t as a function of $\boldsymbol{\mu}_t$:

$$Y_t = G\boldsymbol{\mu}_t + \boldsymbol{\upsilon}_t, \tag{2.8}$$

where $G = RS$ and $\boldsymbol{\upsilon}_t \sim \mathcal{N}(0, \Phi)$, with $\Phi = R\Sigma R^T$. The computation of the rank of G is not simple since it depends on the routing matrix R. In practice, since the number of columns of G is very small, the product RS and its rank can be computed very fast. Therefore, it will be assumed that G is a full column rank matrix. To simplify notation and computations, we use the standardized measurements vector Z_t:

$$Z_t = \Phi^{-\frac{1}{2}} Y_t = H\boldsymbol{\mu}_t + \boldsymbol{\gamma}_t, \tag{2.9}$$

where $H = \Phi^{-\frac{1}{2}} G$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(0, I)$, and I is the identity matrix of correct dimensions. The purpose of this transformation is simply to reduce a given noise covariance matrix to the identity one. The covariance matrix Σ is generally unknown, thus an empirical covariance matrix $\widehat{\Sigma}$ is computed from a few measurements; in section 2.3.1 we show that using just 1 hour of SNMP measurements is enough to provide proper results. Basic results on the estimation of $\widehat{\Sigma}$ can be found in [82].

In this chapter we use this parsimonious SB model to estimate a complete TM X_t from SNMP measurements Y_t at certain time t. Later, in chapter 3, we will

use the SB model to filter-out the contribution of the anomaly-free traffic into the SNMP measurements, producing residuals sensitive to volume anomalies. This allows to infer anomalies in X_t directly from aggregated data Y_t, without the preliminary TM estimation step. This approach clearly improves accuracy and reduces detection delays, because it does not drag possible errors from previous steps. We shall come back to this application of the model in chapter 3.

The TM can be easily computed from SNMP measurements using equation (2.9). Indeed, given the parsimonious structure of the SB model, equation (2.9) represents a well-posed estimation problem, as the number of columns in H is much smaller than the number of rows. We particularly use a Maximum Likelihood (ML) estimation approach to compute an estimated TM. The ML estimate presents well established statistical properties [82]: it is asymptotically optimal, which means that it is asymptotically unbiased and efficient (i.e., it achieves the Cramér-Rao lower bound [29]). Since the SB traffic model (2.9) is a Gaussian linear model, the ML estimate of $\boldsymbol{\mu}_t$, namely $\hat{\boldsymbol{\mu}}_t^{ML}$ corresponds to the least mean squares estimate:

$$\hat{\boldsymbol{\mu}}_t^{ML} = (H^T H)^{-1} H^T Z_t \tag{2.10}$$

This finally leads to the Maximum Likelihood estimate of the TM under the SB model, which we will refer to as the Spline-based Maximum Likelihood Estimate (SMLE) $\hat{X}_t^{SMLE}$, defined as:

$$\boxed{\hat{X}_t^{SMLE} = S\,\hat{\boldsymbol{\mu}}_t^{ML} = \left(S(H^T H)^{-1} H^T \Phi^{-\frac{1}{2}}\right) Y_t} \tag{2.11}$$

2.3.1 Validation of the SB Model and SMLE TM Estimation

Let us present the validation of the Spline-Based model and the evaluation of the SMLE estimation method, using real traffic measurements from different operational backbone networks. We shall begin by describing the datasets that were used. The evaluation is conducted using the real network topologies and real data from the three previously described IP networks: the Abilene network, the GEANT network, and a private commercial Tier-2 ISP network.

Network	n° nodes - links	n° OD-Flows	Data	Sampling
Abilene	12 - 54	132	OD-flows traffic	5’
GEANT	23 - 74	506	OD-flows traffic	15’
Tier-2 ISP	50 - 168	2450	links traffic	10’

Table 2.1 — Network Topologies and Traffic Datasets.

Table 2.1 presents the topology of each of these networks and the corresponding dataset characteristics. Abilene consists of 12 PoPs connected by 30 very-high-speed optical links, and the number of OD-flows is $m = 132$. Abilene traffic data consists

of 5' sampled TMs collected via Netflow from the Abilene Observatory [128] and available at [129]. The GEANT network consists of 23 routers interconnected through 74 high-speed links, representing a total of $m = 506$ OD-flows. GEANT traffic data consists of 15' sampled TMs, built from IGP and BGP routing information and Netflow data in [124], and available on the TOTEM website [130]. The Tier-2 ISP network is a private network composed of 50 nodes interconnected through 168 links, and data is not public. Direct OD-flow measurements are not available for this network. Instead, link traffic volumes are gathered each 10' via SNMP. Using this data and a description of the topology, we perform a Tomo-Gravity estimation of the real OD-flows traffic volume. As regards routing configurations, both the Abilene and the Tier-2 ISP datasets provide them. In the case of GEANT, we use the provided link-weights to compute a shortest-paths routing configuration.

In the following evaluations we assume that traffic flows X_t are unknown, and consider the SNMP measurements Y_t as the input known data. The real values of X_t are only used for validation of the obtained results. In order to verify the stability properties of the proposed traffic model, two sets of measurements are used for each network topology: the "learning" dataset, used to construct the splines basis matrix S and to estimate the covariance matrix $\widehat{\Sigma}$, and the "validation" dataset, used to evaluate the performance of the estimation. Let T_{learn} and T_{val} be the sets of time indexes associated with measurements from the learning and validation datasets respectively.

The SMLE method is a pure SNMP TME method, which uses exclusively SNMP measurements both for calibration and estimation purposes. Thus, both the learning and the validation datasets consist of SNMP measurements. The learning dataset used to construct the SB model is remarkably short: we take just 1 hour of SNMP measurements in Abilene to construct S. Given that the sampling rates in the Tier-2 network and in GEANT are lower than the one used in Abilene, we interpolate intermediate measurements in both learning datasets to keep the duration of the learning dataset in 1 hour. In all cases, the validation dataset is composed of 672 SNMP measurements. The learning dataset is measured one hour before the validation dataset.

The SB model is calibrated for each network using the corresponding learning dataset, following these steps: (i) the Tomo-Gravity Estimate $\hat{x}_t^{TGE}(k)$ is computed for all OD-flows k and all $t \in T_{\text{learn}}$; (ii) the mean OD-flow volume values $\bar{x}^{TGE}(k) = \frac{1}{|T_{\text{learn}}|} \sum_{t \in T_{\text{learn}}} \hat{x}_t^{TGE}(k)$ are computed, where $|T_{\text{learn}}|$ is the number of time indexes in the learning dataset; (iii) finally, the obtained mean values $\bar{x}^{TGE}(k)$ are sorted in ascending order to obtain a rough estimate of OD-flows traffic volume. The SB model is designed with cubic splines ($p = 3$) and 2 knots, representing small, medium-size, and large OD-flows, see figure 2.2. The use of cubic splines comes directly from the shape of the curve to approximate. We use the Matlab Splines Toolbox to design q discrete splines $\mathbf{s}(i)$, $1 \leq i \leq q$. The choice of cubic splines and number of knots results in a total of $q = (p + 1) + 2 = 6$ splines [81], similar to the

example provided in (2.6) and (2.7). This clearly reflects the low-dimensionality of our traffic model, as q is effectively much smaller than the number of OD-flows m in the three network topologies. Finally, the mean traffic volume of each OD-flow $\bar{x}^{TGE}(k)$ is used to compute an estimate $\hat{\sigma}_k^2$ of σ_k^2, which leads to an estimate $\hat{\Phi}$ of Φ, quite efficient and sufficient in practice.

The obtained calibrated SB model is used to infer OD-flows volume from the SNMP measurements of the validation dataset, using the SMLE method defined in (2.11). To assess the accuracy of the SMLE method and to test the performance of the short learning step, we compute the Relative Root Mean Squared Error (RRMSE) for every time t in the validation dataset:

$$\text{RRMSE}(t) = \frac{\sqrt{\sum_{k=1}^{N} \left(x_t(k) - \hat{x}_t^{SMLE}(k)\right)^2}}{\sqrt{\sum_{k=1}^{N} x_t(k)^2}}, \quad \forall t \in T_{\text{val}} \tag{2.12}$$

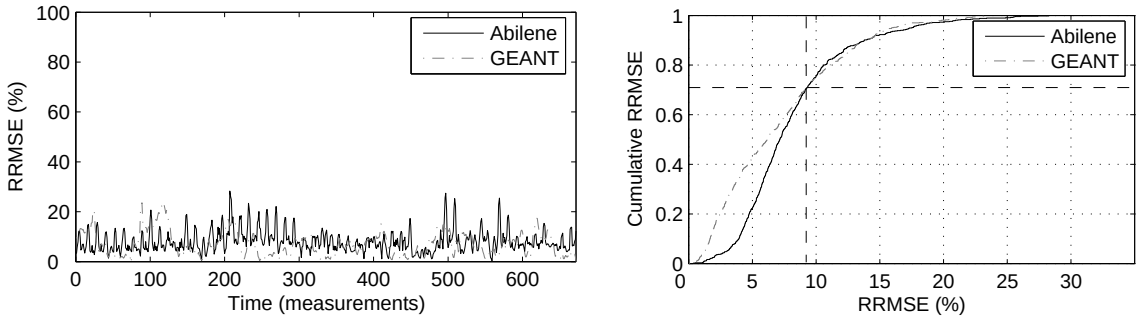
where $x_t(k)$ is the true traffic volume of OD-flow k at time t and $\hat{x}_t^{SMLE}(k)$ denotes the corresponding SMLE estimation. The RRMSE has been used in previous works [22, 23] as a summary of the relative estimation error for a complete TM produced at every time t. In this sense, we shall refer to the RRMSE(t) as the *temporal* estimation error. The value N corresponds to the number of OD-flows that are compared in the RRMSE(t) index. Small-volume OD-flows are well known to be hard to estimate [21, 23], and are generally not considered in (2.12), simply because they have little impact on Traffic Engineering tasks, and so are generally less important to estimate. Following previous works [21, 23], we shall generally exclude from the RRMSE(t) index about 5% of the total traffic, corresponding to these small-volume OD-flows.

Small OD-flows become more important when the objective is Intrusion Detection, mainly because many kinds of network attacks are associated with small OD-flows rather than large OD-flows (e.g., worms propagation, network scans, distributed denials of service). These kinds of attacks are difficult to detect using an OD-flow resolution, and more fine-grained data must be used to detect them.

As regards the computation of RRMSE(t) for the validation dataset of the Tier-2 ISP network, we compare the value of the SMLE estimation $\hat{x}_t^{SMLE}(k)$ against the Tomo-Gravity estimation $\hat{x}_t^{TGE}(k)$, using the Relative Root Mean Squared Difference (RRMSD) between both estimates:

$$\text{RRMSD}(t) = \frac{\sqrt{\sum_{k \in topTG\text{-}T_h} \left(\hat{x}_t^{TGE}(k) - \hat{x}_t^{SMLE}(k)\right)^2}}{\sqrt{\sum_{k \in topTG\text{-}T_h} \left(\hat{x}_t^{TGE}(k)\right)^2}}, \quad \forall t \in T_{\text{val}} \tag{2.13}$$

Comparing all OD-flows in (2.13) is not a reasonable approach. The TGE method provides quite accurate results for relatively large-volume OD-flows, but poor for small OD-flows [21], which are hard to estimate as we have already explained.



(a) RRMSE(t) (b) CDF of RRMSE(t)

Figure 2.3 — (a) RRMSE(t) and (b) Cumulative RRMSE(t) for 672 measurements in the Abilene and the GEANT networks.

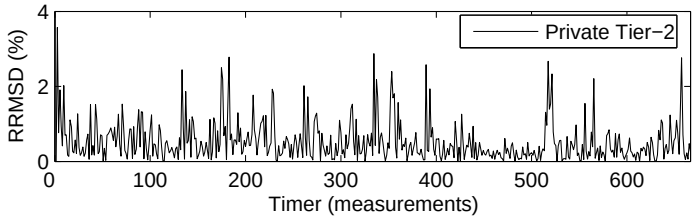


Figure 2.4 — RRMSD(t) for 1500 OD-flows in a Tier-2 ISP network

Therefore, we define the *topTG-*T_h OD-flows as those estimated OD-flows by the TGE method that are reasonably stable in time and that have a mean volume value that exceeds a threshold T_h. In this sense we only keep the most accurately estimated OD-flows, removing the noisy or erratic estimates which seems to be wrongly estimated.

Figure 2.3(a) presents the temporal evolution of the RRMSE(t) for the 672 measurements in the validation datasets for Abilene and GEANT. In both cases, the relative error remains stable in time, reinforcing the observations about time-stability of the SB model we drew from figure 2.2. Figure 2.3(b) shows that more than 70% of the time, estimation relative errors are below 10%. A deeper study of the estimation error shows that in most cases, large relative errors occur in the lowest-volume OD-flows that are analyzed. We further analyze this issue in section 2.7.

The mean values of the RRMSE(t) for the validation dataset are 8.14% for Abilene and 7.04% for GEANT. Many OD-flows in the Abilene and GEANT datasets are negligible, which can be appreciated from figure 2.2. These further explains the accuracy of the results. Methods proposed in the literature as accurate estimates present relative errors that may vary between 5% and 15% [22, 23], so obtained results are satisfactory w.r.t. those works. However, as we will see in section 2.7, these results are somewhat biased by the particular characteristics of the Abilene and the GEANT datasets, and all that we can state for sure is that the SMLE method provides estimation results similar to those obtained by the TGE method in the general case.

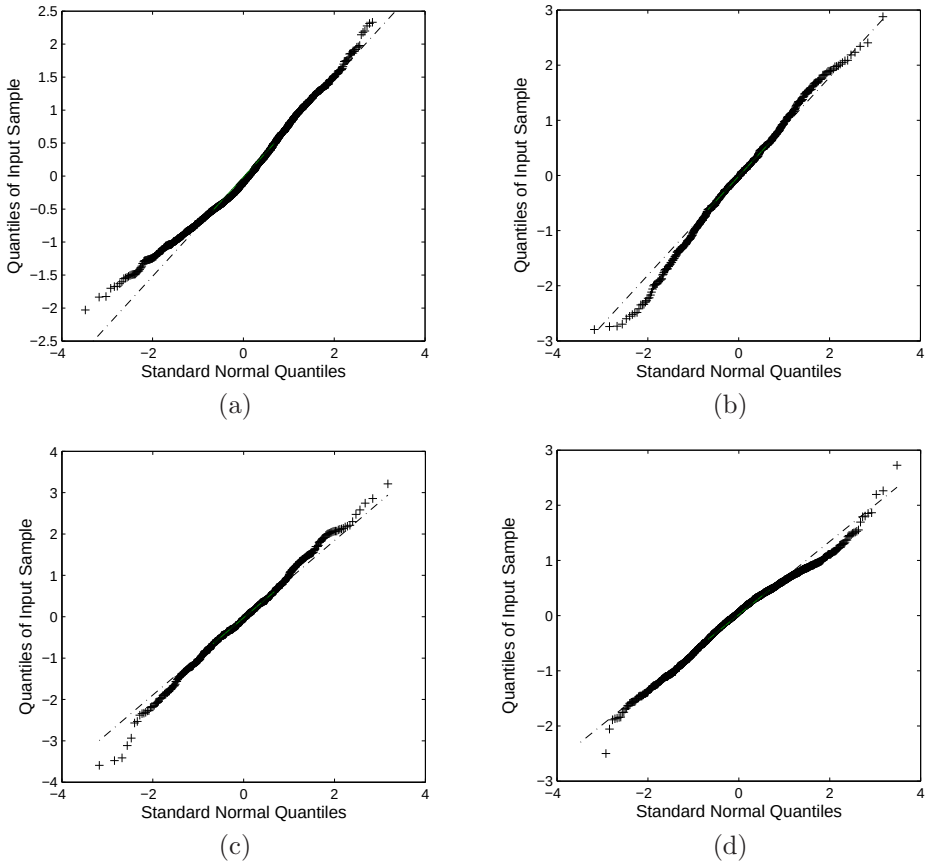


Figure 2.5 — QQ-plots for 2 residual processes from (a,c) Abilene and (b,d) GEANT.

Figure 2.4 depicts the temporal evolution of the RRMSD(t) between the TGE and SMLE estimations, for the commercial Tier-2 ISP network. In this evaluation, we tune T_h such that 95% of the total traffic is evaluated, which represents approximately 60% of the total OD-flows in the network. The relative difference between both estimates is stable in time and has a mean value of 0.57%. As we claimed before, the TGE method is a widely accepted method to estimate OD-flow volumes from link traffic measurements and topology information with confident results, at least for for relatively large-volume OD-flows. Thus, we conclude that the SB model is also accurate for this Tier-2 ISP network.

As a final validation of the SB model, we verify the Gaussian assumption for Abilene and GEANT. For doing so, we analyze the "residuals" of the standardized measurements vector Z_t, i.e., the obtained process after filtering the mean traffic $H\boldsymbol{\mu}_t$ in (2.9). The residuals are obtained by projection of Z_t onto the left null space of H. Quantile-Quantile plots for two of these residual processes are plotted in figure 2.5, both for Abilene and GEANT. These residual processes closely follow a Gaussian distribution, with a variance close to 1 in all cases. Additionally, we verify the Gaussian assumption by applying a Kolmogorov-Smirnov goodness-of-fit hypothesis test to the residual processes. The acceptance rate of this test at the level 5% is 98.5% for Abilene and 97.7% for GEANT, which also confirms the Gaussian assumption of the SB model.

2.4 Recursive Traffic Matrix Estimation

The pure SNMP TME methods presented in previous sections 2.2 and 2.3 are spacial TME methods, which means that they compute an estimated TM X_t at a certain time t given a single value of SNMP measurements Y_t. In this section we present a mixed TME method that not only uses Y_t to estimate X_t, but also takes advantage of the TM temporal correlation, using a set of past SNMP measurements $\{Y_{t-1}, Y_{t-2}, \ldots, Y_1\}$ to compute an estimate $\hat{X}_{t|t} = \mathbb{E}(X_t|Y_t, Y_{t-1}, \ldots, Y_1)$. In [22, 78], authors use the standard Kalman filtering method [51] to recursively compute $\hat{X}_{t|t}$. We draw on the ideas of [22] as a point of departure, then analyze the weaknesses of the proposed approach, and finally extend the method to achieve more accurate and stable results.

2.4.1 A Simple State-Space Model for the Traffic Matrix

Let us consider the model that is assumed in [22, 78]. In this model, authors consider the OD-flows of the TM as the hidden states of a dynamic system. A linear state space model is adopted to capture the temporal evolution of the TM, and the relation between the TM and the SNMP measurements given by (2.1) is used as the observation process:

$$\begin{cases} X_{t+1} &= A\, X_t + W_{t+1} \\ Y_t &= R\, X_t + V_t \end{cases} \tag{2.14}$$

The first equation in (2.14) characterizes the evolution of the TM X_t. Matrix A is the transition matrix that captures the dynamic behavior of the system, and W_t is an uncorrelated zero-mean Gaussian white noise that accounts both for modeling errors and randomness in the traffic flows. The second equation in (2.14) relates the observed links traffic Y_t to the unobserved state X_t through the routing matrix R. The measurement noise V_t is also an uncorrelated zero-mean Gaussian white noise process that models possible inconsistencies in (2.1). Authors in [22, 78] also assume a stationary situation, where A, R, and the noise covariance matrices Q_w and Q_v are constant in time.

Given this model it is possible to recursively derive the least mean squares linear estimate of X_t given $\{Y_t, Y_{t-1}, \ldots, Y_1\}$, $\hat{X}_{t|t} = \mathbb{E}(X_t|Y_t, Y_{t-1}, \ldots, Y_1)$ by using the standard Kalman Filter (KF) method. The KF is an efficient recursive filter that estimates the state X_t of a linear dynamic system from a series of noisy measurements $\{Y_t, Y_{t-1}, \ldots, Y_1\}$. It consists of two distinct phases, iteratively applied: the *Prediction Phase* uses the state estimate from the previous time-step $\hat{X}_{t|t}$ to produce an estimate of the state at the current time-step $t+1$, usually known as the "predicted" state $\hat{X}_{t+1|t} = \mathbb{E}(X_{t+1}|Y_t, Y_{t-1}, \ldots, Y_1)$,

$$\begin{cases} \hat{X}_{t+1|t} &= A\, \hat{X}_{t|t} \\ P_{t+1|t} &= A\, P_{t|t}\, A^T + Q_w \end{cases} \tag{2.15}$$

where $P_{t|t}$ and $P_{t+1|t}$ are the covariance matrices of the estimation error $\mathrm{e}_{t|t} = X_t - \hat{X}_{t|t}$, and the prediction error $\mathrm{e}_{t+1|t} = X_{t+1} - \hat{X}_{t+1|t}$ respectively.

In the *Update Phase*, the measurements vector at current time-step Y_{t+1} is used to refine the prediction $\hat{X}_{t+1|t}$, computing a more accurate state estimate for current time-step $t+1$,

$$\begin{cases} \hat{X}_{t+1|t+1} &= \hat{X}_{t+1|t} + K_{t+1}\,(Y_{t+1} - R\,\hat{X}_{t+1|t}) \\ P_{t+1|t+1} &= (I - K_{t+1}\,R)\,P_{t+1|t}\,(I - K_{t+1}\,R)^T \\ & \quad +K_{t+1}\,Q_v\,K_{t+1}^T = (I - K_{t+1}\,R)\,P_{t+1|t} \end{cases} \tag{2.16}$$

where K_{t+1} is the optimal Kalman gain which minimizes the mean-square error $\mathbb{E}(||\mathrm{e}_{t+1|t+1}||^2)$:

$$K_{t+1} = P_{t+1|t}\,R^T(R\,P_{t+1|t}\,R^T + Q_v)^{-1} \tag{2.17}$$

In order to begin the Kalman filter recursion, initial conditions $\hat{X}_{0|0}$ and $P_{0|0}$ are defined. Since the value of the initial state is unknown, the initial estimate is generally chosen to be $\hat{X}_{0|0} = \mathbb{E}(X_0)$ and its corresponding estimation error covariance matrix $P_{0|0} = \mathbb{E}(||\mathrm{e}_{0|0}||^2)$. The calibration of matrices A, Q_w, and Q_v requires direct OD-flow measurements; in [22], authors use a 24hs period of anomaly-free OD-flow measurements for this purpose.

Combining equations (2.15) and (2.16), the Recursive Kalman Filter-based Estimation (RKFE) method recursively computes $\hat{X}_t^{RKFE} = \hat{X}_{t|t}$ from:

$$\boxed{\hat{X}_t^{RKFE} = (I - K_t R)\,A\hat{X}_{t-1}^{RKFE} + K_t Y_t} \tag{2.18}$$

where the Kalman gain K_t can be recursively computed using equations (2.15), (2.16), (2.17), and the initial conditions $\hat{X}_0^{RKFE}$ and $P_{0|0}$.

2.4.2 Drawbacks of the Former State-Space Model

In [22], authors adopt a non-diagonal structure for the transition matrix A of the simple model previously described, while in [78] they consider a diagonal structure for A. We shall evidence that both choices have major impacts when using a model like (2.14).

Let us compute the expected values of the right and left hand side terms in the first equation of (2.14). From this computation we obtain that $m_X = A\,m_X$, where $m_X = \mathbb{E}(X)$ denotes the average TM value. This implies that $(I - A)m_X = 0$, that is to say that m_X should be in the kernel of $I - A$. Let us consider the case where

A is a diagonal matrix. In this case, the only solution to the system $(I - A)m_X = 0$ is $m_X = 0$, and obviously this condition is not satisfied by the average TM. So particularly, the first equation in (2.14) is false in [78], and in this context it is only valid for centered data, i.e., $m_X = 0$. Even more, our following analysis shows that using (2.14) with non-centered data has convergence implications.

On the contrary, if we consider that A is non-diagonal, it must be calibrated in such a way that $(I - A)m_X = 0$. This is essential in model (2.14) as presented in [22]. In that work, authors claim that the Kalman filter must be re-calibrated every few days, when the underlying model changes, using once again direct OD-flow measurements for a new 24hs period. This seems reasonable for such a particular calibration of A. As we will show in the evaluation results, this need of recalibration can be reduced with some simple corrections to the model.

Let us modify the first equation in (2.14) in order to have a correct state space model for the case of a diagonal state transition matrix A. If we consider the variations of the TM X_t around its average value m_X, i.e., $X_t^c = X_t - m_X$, the system (2.14) becomes:

$$\begin{cases} X_{t+1}^c &= A\, X_t^c + W_{t+1} \\ Y_t &= R\, X_t^c + V_t + R\, m_X \end{cases} \tag{2.19}$$

The first equation in (2.19) is now correct for A diagonal, which corresponds to the case of modeling the centered OD-flows as spatially independent AR(1) processes; even more, the equality of expected values of the left and right hand side terms holds, whatever the choice of A. In this setting, the model is not as sensitive to the definition of the state transition matrix A as in (2.14), where the only solution is to choose A non-diagonal and such that $(I - A)m_X = 0$.

However, the deterministic term that appears in the observation process violates the Kalman filter assumptions; particularly, the "measurement noise" $V_t + R\, m_X$ is not a zero-mean Gaussian process. The appropriate way of treating this problem would be to center the observation process before applying the Kalman filter, using the centered observation measurements vector $Y_t^c = Y_t - \mathbb{E}(Y_t) = Y_t - R\, m_X$. Nevertheless, we apply the Kalman filter equations to system (2.19) in order to appreciate the impact of using non-centered observation data when A is diagonal.

Let us define $\tilde{X}_{t|t}$ as the estimate that one would obtain if the Kalman equations (2.15) and (2.16) were applied with the non-centered SNMP measurements Y_t as input. Using the Kalman filter equations, we can express both the evolution of the estimate $\hat{X}_{t|t}^c = \mathbb{E}(X_t^c | Y_t^c, \ldots, Y_1^c)$ and the evolution of $\tilde{X}_{t|t}$ as:

$$\begin{aligned} (*) \quad \hat{X}_{t+1|t+1}^c &= A\, \hat{X}_{t|t}^c + K_{t+1}\, (Y_{t+1}^c - RA\, \hat{X}_{t|t}^c) \\ (**) \quad \tilde{X}_{t+1|t+1} &= A\, \tilde{X}_{t|t} + K_{t+1}\, (Y_{t+1} - RA\, \tilde{X}_{t|t}) \end{aligned} \tag{2.20}$$

where we have assumed the same Kalman gain in both equations as its value does not depend on the observations. If we define the error $\eta_t = \tilde{X}_{t|t} - \hat{X}^{\mathrm{c}}_{t|t}$, the difference between $(**)$ and $(*)$ can be written as:

$$\eta_{t+1} = (I - K_{t+1}\, R)A\, \eta_t + K_{t+1}\, R\, m_X \tag{2.21}$$

Let us assume that the Kalman filter converges; in that case, we can substitute the Kalman gain in (2.21) by its limit value $K = \lim_{t\to\infty} K_t$:

$$\eta_{t+1} = (I - KR)A\, \eta_t + KR\, m_X \tag{2.22}$$

Without loss of generality, let us suppose that $\eta_0 = 0$. We are going to prove that an error term is propagated and that the error either diverges to infinity or converges to a constant non-null value. As $\eta_0 = 0$, we can express η_t as:

$$\eta_t = \sum_{k=0}^{t-1}((I - KR)A)^k KR\, m_X, \quad \forall t > 0 \tag{2.23}$$

If the spectral radius of $(I-KR)A$ is greater than 1, then the error term η_t diverges to infinity. On the contrary, if the spectral radius of $(I - KR)A$ is lower than 1, then the error term η_t converges to a constant value:

$$\eta_\infty = \lim_{t\to\infty} \eta_t = (I - (I - KR)A)^{-1} KR\, m_X \tag{2.24}$$

This shows that, when considering a diagonal structure for the state transition matrix A in (2.14), not only the state space model is false but even after centering the data and explicitly introducing the mean value m_X, the Kalman filter does not converge to the real value of the traffic matrix if non-centered data Y_t is used in the filter. On the contrary, there is a gap between the real an the estimated value that is proportional to m_X. This is further verified in the evaluation results.

2.4.3 State-Space model for centered TM variations: static mean

This problem can be easily solved in different ways. As we said before, the most obvious solution would be to consider a centered observation process Y^{c}_t. However, we will consider a more standard approach: a deterministic term in the observation process can always be removed by adding a new deterministic state to the state model. Let us define a new state variable $U_t = [X^{\mathrm{c}}_t\ m_X]^T$. In this case, (2.19) becomes:

$$\begin{cases} U_{t+1} &= \begin{bmatrix} A & \mathrm{O} \\ \mathrm{O} & I \end{bmatrix} U_t + \begin{bmatrix} W_{t+1} \\ \mathrm{O} \end{bmatrix} = C\, U_t + \Psi_{t+1} \\ \\ Y_t &= \begin{bmatrix} R & R \end{bmatrix} U_t + V_t = B\, U_t + V_t \end{cases} \tag{2.25}$$

where O is the null matrix of correct dimensions. This new model has twice the number of states, augmenting the computation time and complexity of the Kalman filter. However, it presents several advantages:

- It is not necessary to center the observations Y_t.
- Matrix A can be chosen as a diagonal matrix, which corresponds to the case of modeling the centered OD-flows as AR(1) processes. Autoregressive models have been widely applied in the traffic matrix literature [80]; as we show in the validation, obtained results with a simple AR(1) model and the RKFE technique are accurate compared to the target error for standard TME methods, and this is clearly much easier and more stable than calibrating a non-diagonal matrix such that $(I - A)m_X = 0$.

  In fact, authors in [78] observe that re-calibrations are often not needed when using a diagonal transition matrix, and the results we obtain are stable during the whole evaluation period of one week, which is not the case in [22].
- The Kalman filter estimates the mean value of the OD-flows m_X, assumed constant in (2.25).
- This model allows to impose a dynamic behavior to m_X, improving the estimation properties of the filter.

This is exactly the step we take in the following section.

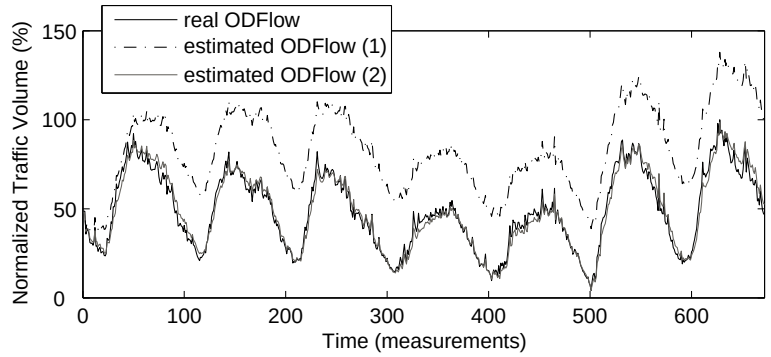
2.4.4 Extending the model: dynamic mean

Using model (2.25) with the Kalman filtering technique produces quite good estimation results as we show in section 2.4.5. However, this model presents a major drawback: it assumes that the mean value of the OD-flows m_X is constant in time. We improve (2.25) by adopting a simple dynamic model for m_X, in order to allow small variations of the OD-flows mean value:

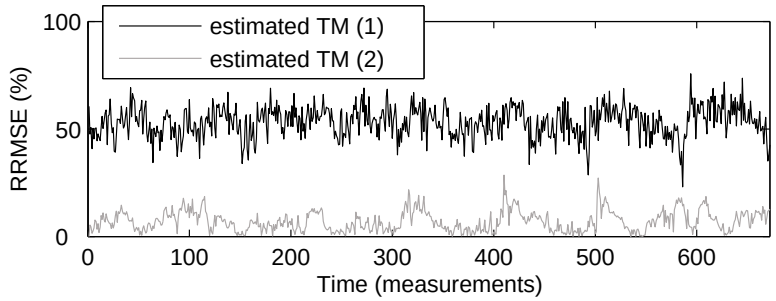
$$m_X(t+1) \;=\; m_X(t) + \boldsymbol{\zeta}_{t+1} \tag{2.26}$$

where $m_X(t)$ represents the dynamic mean value of X_t and $\boldsymbol{\zeta}_t$ is a zero-mean white Gaussian noise process with covariance matrix $Q_{\boldsymbol{\zeta}}$. This model corresponds to a random walk process, which is commonly applied to describe several dynamic models in economics, physics, etc. In this context, (2.25) becomes:

$$\begin{cases} U_{t+1} &= \begin{bmatrix} A & \mathrm{O} \\ \mathrm{O} & I \end{bmatrix} U_t + \begin{bmatrix} W_{t+1} \\ \boldsymbol{\zeta}_{t+1} \end{bmatrix} \;=\; C\,U_t + \Theta_{t+1} \\ \\ Y_t &= \begin{bmatrix} R & R \end{bmatrix} U_t + V_t \;=\; B\,U_t + V_t \end{cases} \tag{2.27}$$



(a) Single estimated OD-flow using the RKFE method.



(b) RRMSE(t) for the RKFE method.

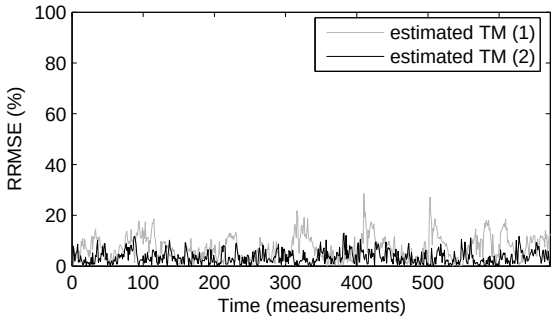
Figure 2.6 — (a) Single estimated OD-flow and (b) RRMSE(t) using RKFE for (1) model (2.14) and (2) model (2.25).

As we show in the results obtained in the following section, such a simple model provides more accurate and more stable results.

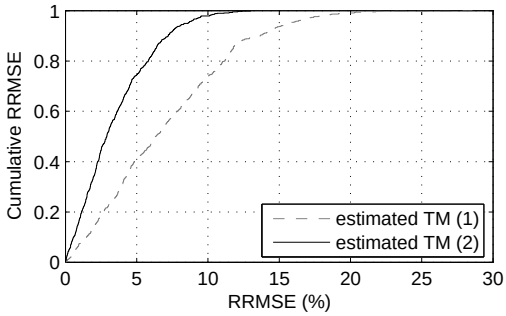
2.4.5 Evaluation of the RKFE TME method

The first evaluation consists in evidencing the convergence problem of the RKFE method when using a model like (2.14) with A diagonal, as it is done in [78]. In this sense, we compare the performance of the Kalman filter using models (2.14) and (2.25). In both cases we adopt a diagonal structure for the state transition matrix A, namely an AR(1) model for each OD-flow.

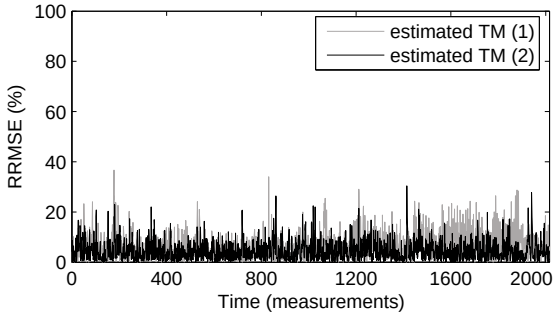
In this evaluation and through the rest of section 2.4.5, the learning dataset is composed of 24hs of direct OD-flow measurements X_t, as it is the case in [22]. The validation dataset consists of 1 week of SNMP measurements from the GEANT dataset, which represents 672 measurements. We also assume that the relation between X_t and Y_t is exact, that is to say $V_t = 0, \forall t$. The learning dataset is used to calibrate both models (2.14) and (2.25), namely estimate the corresponding transitions matrices and noise covariance matrices, i.e., the AR(1) parameters. We use the Yule-Walker method to compute these matrices. This method solves the Yule-Walker equations for the AR processes by means of the Levinson-Durbin recursion, see [52] for details.



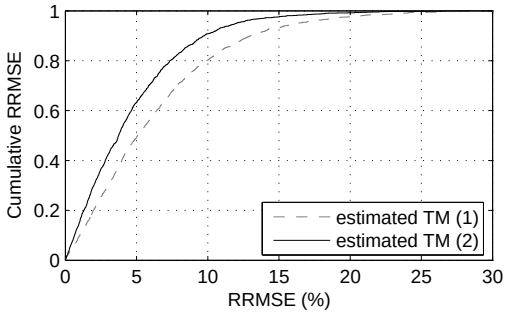
(a) RRMSE(t) in GEANT



(b) CDF RRMSE(t) in GEANT



(a) RRMSE(t) in Abilene



(b) CDF RRMSE(t) in Abilene

Figure 2.7 — RRMSE(t) and Cumulative RRMSE(t) for 1 week of traffic in GEANT and Abilene, using (1) model (2.25) and (2) model (2.27)

Figure 2.6(a) depicts the estimation of one sample OD-flow with both Kalman filters; the full black curve represents the real OD-flow; the dashed black curve depicts the estimated OD-flow using model (2.14); the full gray curve depicts the estimated OD-flow using model (2.25). In both cases, the Kalman filter properly tracks the real traffic behavior, as both curves shape are similar to the real one. However, there is a clear error-gap when using model (2.14), which comes from our previous analysis.

Figure 2.6(b) shows the evolution of the relative estimation error RRMSE(t). The mean relative error is 53.4% for model (2.14) and 6.2% for model (2.25). In both cases, the error evolution is quite stable around its mean value during the whole evaluation week, providing a first evidence of the stability advantages of a diagonal transition matrix.

We now compare the estimation performance of the RKFE method for models (2.25) and (2.27), namely assuming a constant mean value for the TM or a random walk process, using a diagonal transition matrix in both cases. For this purpose, we consider a week of traffic in Abilene and GEANT. We consider the same assumptions adopted in the previous evaluation and calibrate the different matrices in the same way. In order to estimate the covariance matrix $Q_{\boldsymbol{\zeta}}$ of the random walk noise process $\boldsymbol{\zeta}_t$, we take the following steps: using a sliding window averaging filter, we

first remove the fast temporal variations from the direct OD-flow measurements of the learning dataset. For each OD-flow time-series, we consider the approximate derivative time-series, i.e., the difference of consecutive measurements, and compute its variance. We finally use this variance as an estimate of each diagonal element in $Q_{\boldsymbol{\zeta}}$.

Figure 2.7 depicts the relative estimation error evolution for the TM using both models and one week of measurements in GEANT and Abilene. The cumulative RRMSE is also depicted in these figures. The obtained mean values of the relative errors are 6.20% and 4.23% in GEANT and 6.87% and 4.48% in Abilene, for models (2.25) and (2.27) respectively. These results are slightly better than those obtained for an ISP Tier-1 network (Sprint) in the former work of Kalman filters for TME [22], where estimation errors have a mean value around 7%.

We can draw two important conclusions from both evaluations: in both cases, considering a variable mean value $m_X(t)$ produces better results, both as regards accuracy and stability, as the curve of cumulative RRMSE shows a sharper growth. The second conclusion is about the advantage of correctly using a diagonal transition matrix. In all evaluations the stable evolution of the error shows that the underlying model remains valid during several days when considering such a transition matrix, a major advantage w.r.t. the results obtained in the former work [22]. This simple observation has a major impact on the applicability of the method in a real scenario: if the underlying model remains stable, it is not necessary to conduct periodical re-calibrations, dramatically reducing measurement overheads.

2.5 The Random Neural Network for TME

The mixed TME method introduced in this section has its origins in the statistical learning field. More precisely, the method uses multiple Random Neural Networks (RNNs) to reconstruct OD-flows volume X_t from aggregated SNMP link measurements Y_t. From (2.1), we know that traffic volume at link i and time t is a linear combination of OD-flows volume at time t, given by the i-th row of the routing matrix R, referred as $R(i,\cdot)$:

$$y_t(i) = R(i,\cdot)\, X_t \tag{2.28}$$

Note that we have intentionally omitted the subscript t in the routing matrix R; as in previous sections, we assume that R is constant in time. We shall discuss the impacts of such a choice in section 2.7. The main idea of our method is to find a certain non-linear transfer-block $f_k(\cdot) : \mathbb{R}^{n_k} \to \mathbb{R}$ for each OD-flow k, such that:

$$x_t(k) = f_k\left(Y_t(\boldsymbol{\delta}_k)\right) = f_k\left(y_t(\delta_k^1), y_t(\delta_k^2), .., y_t(\delta_k^{n_k})\right) \tag{2.29}$$

The vector $Y_t(\boldsymbol{\delta}_k)$ contains the traffic volume of the n_k links which are crossed by OD-flow k, where $\boldsymbol{\delta}_k = (\delta_k^1, \delta_k^2, .., \delta_k^{n_k})$ has the indexes of the n_k elements in the k-th column of R which are different from zero. The non-linear transfer-block $f_k(\cdot)$ extracts the value of OD-flow traffic volume k from the trace that this flow leaves, as a result of the routing process. We use the term transfer-block instead of function because $f_k(\cdot)$ can not be formally defined as it. It is easy to see that, in theory, the same values of links volume can result from different combinations of OD-flows traffic, and thus the inverse function may not even exist. However and as we will see in the results, this does not happen in practice and $f_k(\cdot)$ can be seen as a non-linear function, without a formal expression. Computing $f_k(\cdot)$ can be simply thought as computing a pseudo-inverse matrix from routing matrix R, for a particular element of the TM. Indeed, we will show in the evaluations that the structure of $f_k(\cdot)$ is strongly related to the characteristics of R.

The idea of the TME method is then to learn the transfer-block $f_k(\cdot)$ from measurements, using in particular a Random Neural Network model. RNNs, also known as G-Networks are a new family of neural networks developed by E. Gelenbe in the late 80's [46]. RNNs have been successfully applied in many different areas during the past years [40, 41, 42, 43, 44], but as a statistical learning tool they are still quite unknown. In this sense, we shall present below a detailed description of the RNN model, the learning algorithm of the RNN and its use as an estimation tool.

2.5.1 The Random Neural Network Model

The RNN model can be described as a merge between the classical Artificial Neural Network (ANN) model and queuing networks. RNNs are, as ANNs, composed of a set of interconnected neurons. Each neuron exchange impulse signals with other neurons and with the environment, and has a potential associated with it, which is an

integer random variable. The potential of neuron i at time t is denoted by $q_t(i)$. If the potential of neuron i is strictly positive, the neuron is *excited*; in this state, it randomly sends signals according to a Poisson process with rate r_i. In the RNN model, neurons exchange *positive* and *negative* signals. The probability that a signal sent by neuron i goes to neuron j as a positive one is denoted by $p^+_{i,j}$, and as a negative one by $p^-_{i,j}$. The signal leaves the network with probability d_i. So, if N is the number of neurons, we must have for all $i = 1, \ldots, N$ that:

$$d_i + \sum_{j=1}^{N} \left(p^+_{i,j} + p^-_{i,j}\right) = 1, \qquad \forall i = 1, \ldots, N \tag{2.30}$$

When a neuron receives a positive signal, its potential is increased by 1; if it receives a negative one, its potential decreases by 1 if it was strictly positive, and it does not change if its value was 0. In the same way, when a neuron sends a signal, positive or negative, its potential is decreased by one. The flow of positive and negative signals arriving from the environment to neuron i is also a Poisson process of rate λ^+_i and λ^-_i respectively. It is possible to have $\lambda^+_i = 0$ and/or $\lambda^-_i = 0$ for some neuron i, but in general we have that $\sum_{i=1}^{N} \lambda^+_i > 0$, that is to say that the neural network receives incoming signals from the environment. Finally, we make the usual independence assumptions between the arrival processes, the processes composed of the signals sent by each neuron, etc.

E. Gelenbe proved in [46] that this model has a product form stationary solution. This is similar to the classical Jackson's result for open queuing networks. If the process of neurons potential $\mathbf{q}_t = (q_t(1), q_t(2), .., q_t(N))$ is ergodic (we will say that the neural network is *stable*), Gelenbe demonstrated that the joint stationary probability distribution $\boldsymbol{\pi}(\beta_1, \beta_2, \ldots, \beta_N) = \lim_{t\to\infty} \Pr(\mathbf{q}_t = (\beta_1, \beta_2, \ldots, \beta_N))$ exists and that it is given by the product of the marginal neuron potential probabilities in equilibrium $\pi_i(\beta_i)$:

$$\boldsymbol{\pi}(\beta_1, \beta_2, \ldots, \beta_N) = \prod_{i=1}^{N} \pi_i(\beta_i) = \prod_{i=1}^{N} (1 - \rho_i)\, \rho_i^{\beta_i} \tag{2.31}$$

where ρ_i is the limit probability (i.e. in equilibrium) that neuron i is excited, which corresponds to a strictly positive potential:

$$\rho_i = \lim_{t\to\infty} \Pr(q_t(i) > 0) \tag{2.32}$$

Equation (2.31) does not imply that the interconnected neurons have a behavior that is independent of each other. Indeed, the probabilities that each neuron is excited are obtained from a coupled non-linear system of equations. Using some basic queuing theory concepts, it is easy to see that, for every neuron i, ρ_i is nothing but the ratio between the rate of incoming positive signals, namely λ_i, and the sum of the rate of outgoing signals plus the rate of incoming negative signals, namely μ_i:

$$\boxed{\begin{array}{lcll} \rho_i & = & \dfrac{\lambda_i}{\mu_i} & \forall i = 1, \ldots, N \\ \lambda_i & = & \lambda_i^+ + \displaystyle\sum_{j=1}^{N} \rho_j \, r_j \, p_{j,i}^+ & \forall i = 1, \ldots, N \\ \mu_i & = & r_i + \lambda_i^- + \displaystyle\sum_{j=1}^{N} \rho_j \, r_j \, p_{j,i}^- & \forall i = 1, \ldots, N \end{array}} \tag{2.33}$$

The non-linear system of equations defined in (2.33) has $3N$ equations and $3N$ unknowns, defined by ρ_i, λ_i, and μ_i, for all $i = 1, \ldots, N$.

Similar to Jackson's result, Gelenbe proved in [46, 47, 48] that this system of equations has a unique solution if, for every neuron i in the network, $\rho_i < 1$. In this case the neural network is said to be stable, where stability is understood in the sense that all moments (marginal or joint) of the neural network can be found, and are all finite.

From the simple neural network model defined in (2.33) and using equation (2.30) we can see that, given the external *stimulus* rates λ_i^+ and λ_i^-, and the probabilities of outgoing signals d_i, the RNN has $2N^2+N$ free parameters (the rates r_i and the branching probabilities $p_{i,j}^+$ and $p_{i,j}^-$) that can be calibrated to build a non-linear transfer-block $f(\cdot)$ like the one previously described. In the following section we present a simple statistical learning procedure to build such a block.

2.5.2 Learning in the Random Neural Network

Let us now describe the use of the RNN model as a statistical learning tool. The RNN can be seen as a black box composed of N interconnected neurons, where the incoming signal rates $\boldsymbol{\lambda}^+ = \left(\lambda_1^+, \lambda_2^+, \ldots, \lambda_N^+\right)$ and $\boldsymbol{\lambda}^- = \left(\lambda_1^-, \lambda_2^-, \ldots, \lambda_N^-\right)$ are the inputs, and the steady-state probabilities of neuron excitement $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_N)$ are the outputs.

As in most RNN applications, we shall consider that the rate of incoming negative signals from the environment λ_i^- is 0 for all the neurons of the network, which means that only excitatory signals arrive from outside the network. In this context, this black box has a certain transfer-block $f(\cdot)$ that relates the N inputs with the N outputs:

$$\boldsymbol{\rho} = f\left(\boldsymbol{\lambda}^+\right) \tag{2.34}$$

where the transfer-block $f(\cdot)$ depends on the number of neurons N, the connection topology of the RNN network, the branching probabilities $p_{i,j}^+$ and $p_{i,j}^-$, and the rates r_i (the probabilities of outgoing signals d_i are generally defined by the RNN topology).

Not every input λ_i^+ and output ρ_i in (2.34) is necessarily used in a RNN application. In general, some λ_i^+ are set to 0, and only a subset of ρ_is is used as output.

In our particular application of TM estimation, we have m transfer-blocks $f_k(\cdot)$ to build, one for each OD-flow $k = 1, .., m$ of the TM. For each of these blocks there are n_k inputs and one single output. The n_k inputs correspond to the traffic volume of the n_k links $Y_t(\boldsymbol{\delta}_k)$ that are traversed by OD-flow k, see (2.29). The output is the estimated OD-flow traffic volume $x_t(k)$.

Given that the output of the RNN is a probability, we must scale the value of the OD-flow volume $x_t(k)$ during the learning phase to be consistent with the RNN model. We do this by simply normalizing the OD-flow volume $x_t(k)$ by the smallest link capacity of the n_k links that it crosses. In this normalization we assume that the routing process is always stable, in the sense that every link in the network is never over-loaded, even in the occurrence of strong congestion situations. In other words, we suppose that $y_t(i) < c_i$, $\forall i = 1, \ldots, r$ and $\forall t$. We shall use $z_t(k)$ as the normalized volume of OD-flow k:

$$\begin{aligned} z_t(k) &= \frac{x_t(k)}{c_{\delta_k^{\min}}}, \quad 0 \le z_t(k) \le 1 \\ c_{\delta_k^{\min}} &= \min_{\delta_k^i} \{C(\boldsymbol{\delta}_k)\} = \min \left\{ c_{\delta_k^1}, c_{\delta_k^2}, \ldots, c_{\delta_k^{n_k}} \right\} \end{aligned}$$

In a similar way, and even if the inputs in the RNN model can take any arbitrary positive value, i.e. $\lambda_i^+ > 0$, we will use the link utilization $u_t(i) = y_t(i)/c_i$ as input instead of the traffic volume of the links $y_t(i)$:

$$U_t(\boldsymbol{\delta}_k) = \frac{Y_t(\boldsymbol{\delta}_k)}{C(\boldsymbol{\delta}_k)} = \left(\frac{y_t(\delta_k^1)}{c_{\delta_k^1}}, \frac{y_t(\delta_k^2)}{c_{\delta_k^2}}, \ldots, \frac{y_t(\delta_k^{n_k})}{c_{\delta_k^{n_k}}} \right) \tag{2.35}$$

In order to simplify notation and remove the index k from previous notation, we shall describe from now on how to build a particular transfer-block $f(\cdot)$ to estimate the traffic volume of some particular OD-flow in the TM.

Let us suppose that we have defined a certain RNN topology to construct $f(\cdot)$, and that we have a *learning dataset*, composed of T input-output pairs or *patterns* $\{U_t, z_t\}$, $t = 1, \ldots, T$. The *supervised* learning algorithm permits to obtain the values of the free parameters $p_{i,j}^+$, $p_{i,j}^-$, and r_i for all $i, j = 1, \ldots, N$ such that, if we set the n inputs of the RNN $\boldsymbol{\lambda}^+ = \left(\lambda_1^+, \lambda_2^+, \ldots, \lambda_n^+\right)$ to the n link utilization values $U_t = (u_t(1), u_t(2), .., u_t(n))$, then the excitement probability of the output neuron ρ_o is close to z_t. This must hold for every pattern of the learning dataset, i.e. $\forall\, t = 1, \ldots, T$.

In our TME method we use a particular neural network topology that simplifies the non-linear system of equations of the RNN model (2.33). As in most applications of neural networks for learning purposes, we use a three-layers feed-forward neural network topology like the one depicted in figure 2.8. In such a topology, the set of N neurons is divided into three subsets: a set of I input neurons that compose the *input layer*, a set of H hidden neurons that compose the *hidden layer*, and a set of O output neurons that compose the *output layer*, such that $N = I + H + O$. Input neurons receive positive signals from the environment and are only connected to hidden neurons. Hidden neurons do no interact with the environment and are only connected to output neurons. Output neurons only send signals to the environment, and there is no interaction between neurons of the same layer.

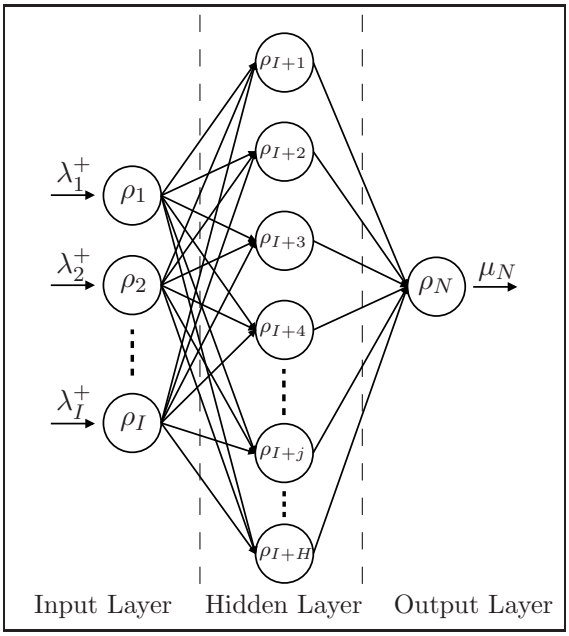


Figure 2.8 — Three-layers feed-forward RNN topology.

In our case, the number of input neurons I is equal to the number of links n that route the OD-flow that we want to estimate, and $O = 1$. The number of hidden neurons H is a variable generally defined by heuristics, but we will come back to this *critical* issue later. Given this topology, we can rewrite (2.33) as follows:

$$
\begin{array}{lll}
\text{(a)} \quad \rho_i = \dfrac{\lambda_i^+}{r_i} & & \forall \text{ input neuron } i \\
\text{(b)} \quad \rho_h = \dfrac{\sum_{\text{input neuron } i} \rho_i w_{i,h}^+}{r_h + \sum_{\text{input neuron } i} \rho_i w_{i,h}^-} & & \forall \text{ hidden neuron } h \\
\text{(c)} \quad \rho_o = \dfrac{\sum_{\text{hidden neuron } h} \rho_h w_{h,o}^+}{r_o + \sum_{\text{hidden neuron } h} \rho_h w_{h,o}^-} & & o \equiv N
\end{array}
\tag{2.36}
$$

where instead of working with branching probabilities $p^+_{i,j}$ and $p^-_{i,j}$, we use neural network *weights* $w^+_{i,j} = r_i\, p^+_{i,j}$ and $w^-_{i,j} = r_i\, p^-_{i,j}$. We use the term weights by analogy to standard ANNs.

In this topology, the only neuron that sends signals to the environment is the output neuron o, and thus $d_i = 0$, $\forall\, i \neq o$, and $d_o = 1$. The rate r_o is a design constant and it can be chosen arbitrarily. Using (2.30), we can express the values of r_i and r_h as a function of the neural network weights:

$$\begin{aligned} r_i &= \sum_{\text{hidden neuron } h} \left(w^+_{i,h} + w^-_{i,h}\right) && \forall \text{ input neuron } i \\ r_h &= w^+_{h,o} + w^-_{h,o} && \forall \text{ hidden neuron } h \end{aligned} \tag{2.37}$$

The system of equations (2.36) and (2.37) that define the RNN model for the three-layers feed-forward topology has a total of $2H(I+1)$ parameters to calibrate, the $2HI$ weights $w^{+/-}_{i,h}$ between the input and hidden layers, and the $2H$ weights $w^{+/-}_{h,o}$ between the hidden and output layers.

In [48], authors present a simple recursive algorithm to learn these parameters from a given learning dataset. Let us define some additional matrices to describe this learning algorithm: we shall consider the positive and negative weight matrices $\mathbf{w}^+ = \{w^+_{i,h}, w^+_{h,o}\}$ and $\mathbf{w}^- = \{w^-_{i,h}, w^-_{h,o}\}$, as well as the RNN weights matrix $\mathbf{w} = \{\mathbf{w}^+, \mathbf{w}^-\}$. The learning algorithm is a first order method that consists in a simple gradient descent method, corresponding to the minimization of a quadratic cost function, the Mean Square Error (MSE) at the output neuron, namely $J(\mathbf{w})$. Given a learning couple (U_t, z_t), the MSE cost function can be expressed as:

$$J_t(\mathbf{w}) = \frac{1}{2}\left(z_t - \rho_o(U_t, \mathbf{w})\right)^2 \tag{2.38}$$

where $\rho_o(U_t, \mathbf{w})$ can be derived from (2.36) and (2.37), using $\lambda^+_i = U_t(i)$, $\forall$ input neuron i.

The learning algorithm starts by initializing the RNN weights at some value $\mathbf{w}_0$; in the absence of additional information, this initialization is done at random, among non-negative matrices. At each iteration t, the weights $\mathbf{w}_t$ are updated in a direction that reduces the value of the cost function J_{t+1}:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{\Delta w}_t \tag{2.39}$$

$$\mathbf{\Delta w}_t = -\eta \frac{\partial J_t(\mathbf{w})}{\partial\, \mathbf{w}}\bigg|_{\mathbf{w}\,=\,\mathbf{w}_t} \tag{2.40}$$

where η is the learning rate, which merely indicates the relative size of the change in the weights. Using (2.38), the derivative in (2.40) can be written as:

$$\mathbf{\Delta w}_t \;=\; \eta\left(z_t - \rho_o\left(U_t, \mathbf{w}_t\right)\right) \frac{\partial \rho_o\left(U_t, \mathbf{w}\right)}{\partial\, \mathbf{w}}\bigg|_{\mathbf{w}\,=\,\mathbf{w}_t} \tag{2.41}$$

Thus, the rule of weight update for a generic weight term $w_{i,j}^{(t)}$ of the RNN at iteration t is given by the following expression:

$$w_{i,j}^{(t+1)} \;=\; w_{i,j}^{(t)} \;+\; \Delta w_{i,j}^{(t)} \tag{2.42}$$

$$\Delta w_{i,j}^{(t)} \;=\; \eta\left(z_t - \rho_o\left(U_t, \mathbf{w}_t\right)\right) \frac{\partial \rho_o\left(U_t, \mathbf{w}\right)}{\partial w_{i,j}}\bigg|_{\mathbf{w}\,=\,\mathbf{w}_t} \tag{2.43}$$

where $\rho_o\left(U_t, \mathbf{w}_t\right)$ and its partial derivatives w.r.t. to positive and negative weights, both at the input and hidden layers, can be computed from (2.36) and (2.37). The computation of these partial derivatives is probably the most challenging step in the update of network weights. Nevertheless, [48] provides an algebraic expression for the aforementioned quantities, easily and efficiently computable.

This learning rule makes intuitive sense. Indeed, note that the weight update is proportional to the difference between the desired output z_t and the *learned* output $\rho_o\left(U_t, \mathbf{w}_t\right)$, which means that no updates should be done if we get the desired output. At the same time, the rule is proportional to the sensitivity of the output w.r.t. to the weight to update, represented by the partial derivative $\partial\rho_o/\partial w_{i,j}$. If this value is small, then weight $w_{i,j}$ has little effect on the output and changing its value does not modify the error. The variable η permits to control the *learning speed* of the algorithm, and it can be updated during the learning process to avoid possible instabilities; in general, bigger values of η are used at the beginning of the recursive process, using smaller values when reaching the optimum to avoid oscillations around it.

The gradient descent algorithm does not ensure a positive value of weights $w_{i,j}^{(t)}$ during the iterative process. Since negative weights are not allowed in the RNN model, we simply set this weight to zero and use its null value in the following iteration in case we get a negative weight update. Another approach usually applied is to modify the learning rate so as to obtain a positive update, but to make it simpler and following previous studies we keep the former solution.

A major concern regarding the iterative learning algorithm is that of local minimum problems. Depending on the particular problem, the error function $J\left(\mathbf{w}\right)$ may not be convex w.r.t. network weights $\mathbf{w}$, and thus the gradient descent may rapidly lead to a local minimum solution. In [49], the author evidenced this drawback in a combinatorial optimization problem arising in emergency response, and proposed an improved weight initialization method to start the descent algorithm from a closer to the global optimum starting point. In our particular application we were not able to distinguish this kind of behavior, and thus we kept the simple random initialization, which is in additional the most used approach in the literature.

To conclude with the learning section, we shall discuss two important related issues: the learning *protocol* and the stopping criterion. The learning protocol specifies the way the patterns of the learning dataset $\{U_t, z_t\}$, $t = 1, \ldots, T$ are used in the supervised algorithm previously described. The three most well known learning protocols are *batch* learning, *stochastic* learning, and *on-line* learning.

In batch learning, the complete dataset is used to compute a single update of the RNN weights. At each iteration t of the recursive algorithm, the T patterns of the learning dataset are applied to the network and the corresponding $k = 1, \ldots, T$ individual weight updates are summed; only then the actual RNN weight is updated, according to (2.42). As a result, the rule of weight update in (2.43) is slightly modified for batch leaning:

$$\Delta w_{i,j}^{(t)} = \eta \sum_{k=1}^{T} \left(z_k - \rho_o\left(U_k, \mathbf{w}_t\right)\right) \frac{\partial \rho_o\left(U_k, \mathbf{w}\right)}{\partial w_{i,j}}\bigg|_{\mathbf{w} = \mathbf{w}_t} \tag{2.44}$$

Batch learning can be seen as a *robust* gradient descent method, because the gradient is estimated at each iteration t based on the whole learning dataset, which improves the accuracy of each step towards the optimum. The drawback of this approach is that training usually becomes slower.

In stochastic and on-line learning, the RNN weights are updated for each single pattern of the learning dataset, which means that the iteration index t in (2.42) and (2.43) corresponds to a single learning pattern. The only difference between stochastic and on-line learning is that the learning pattern used at each iteration is randomly chosen in the former, whereas patterns are used sequentially in the latter.

In contrast with batch learning, both stochastic and on-line learning methods approximate the gradient descent direction based on a single pattern, which makes the learning faster but more uncertain. Batch learning is usually applied when the learning dataset is small and/or highly *heterogeneous* (very different patterns with small representation in the learning dataset). In our case, the learning dataset is large compared to the number of RNN weights that we have to calibrate, thus we use an on-line learning approach.

The learning algorithms continue to iterate through the learning dataset until some convergence or stopping criterion is met. An intuitive and easy to apply stopping criterion is to conclude the learning procedure when the learning error becomes smaller than a predefined value, or when the change in the cost function is below certain threshold. There are many other stopping criteria which can be used instead and that may provide better learning results, mainly regarding *over-fitting* problems.

Over-fitting occurs when the neural network memorizes the learning patterns instead of learning the underlying model. Networks with too many free parameters

will generally present over-fitting problems if the stopping criterion is to minimize the learning error. The most well known method to avoid over-fitting is *early-stopping*. In early-stopping, part of the learning dataset is put aside to validate the quality of the training procedure after each iteration t. Instead of stopping the algorithm when the learning error becomes small enough, the method stops when the error in the validation dataset begins to increase instead of decreasing. As we said before, in our case the learning dataset is large enough compared to the number of RNN weights to learn, and thus the occurrence of over-fitting problems is unlikely.

To sum up, we use an on-line learning protocol, stopping the learning recursive algorithm defined in (2.42) and (2.43) when the estimation error becomes bellow a certain predefined threshold θ_{stop}, or alternatively, when a maximum number of iterations $k_{\max}$ is attained. Such an approach usually requires more iterations than the number of samples T in the learning dataset, and thus we may have to iterate more than once over the complete learning dataset, starting again from the first pattern once the T samples have been used. The following pseudo-code describes this learning algorithm:

Algorithm 1 RNN Learning for TM Estimation

begin (RNN Initialization)
  (a) set number of hidden neurons H
  (b) set η, θ_{stop}, and $k_{\max}$
  (c) $\mathbf{w} \leftarrow \mathbf{w}_0$
  (d) $k \leftarrow 0$, $t \leftarrow 0$
end (RNN Initialization)
do $(k \leftarrow k+1)$ & $(t \leftarrow t+1)$
  for (every pair (i,j) of interconnected neurons)
    $\Delta w_{i,j} \leftarrow \eta \left(z_t - \rho_o\left(U_t, \mathbf{w}\right)\right) \dfrac{\partial \rho_o\left(U_t, \mathbf{w}\right)}{\partial w_{i,j}}$
    $w_{i,j} \leftarrow w_{i,j} \; + \; \Delta w_{i,j}$
  end
  if $t = T$
    $t \leftarrow 0$
  end
until $(J(\mathbf{w}) < \theta_{\text{stop}}) \parallel (k > k_{\max})$
$\mathbf{w}_{\text{trained}} \leftarrow \mathbf{w}$
return $\mathbf{w}_{\text{trained}}$

2.5.3 Using the RNN Model for TM Estimation

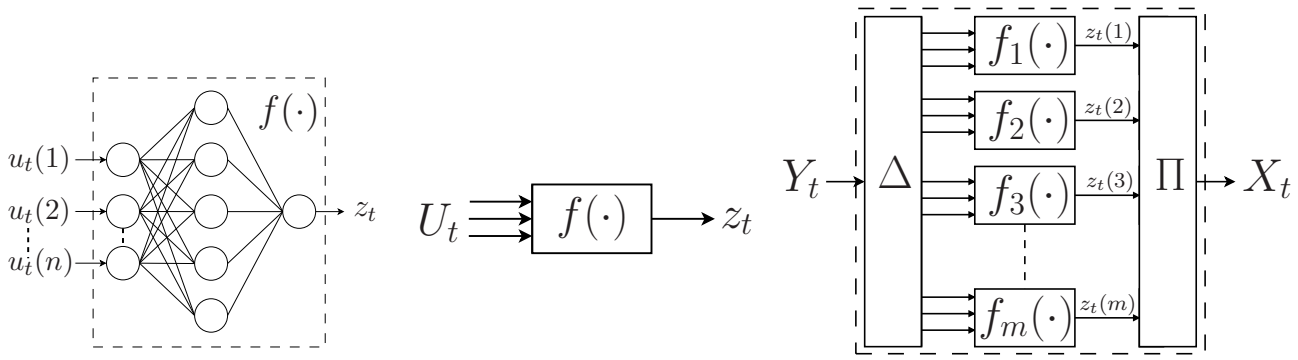
Once the RNN model has been correctly trained, the estimation of the traffic volume of each OD-flow $x_t(k)$ is performed extremely fast. This is a direct consequence of the three-layers feed-forward topology in the RNN model, defined in (2.36). Given the RNN weights $\mathbf{w}$, both r_i and r_h are constants for every input and hidden neuron i and h. Thus, to obtain the estimation result at the output, we have a computation cost of: I products in (2.36)(a), $2IH + H$ products and $2IH + H$ sums in (2.36)(b), and finally $2(H + 1)$ products and $2(H + 1)$ sums in (2.36)(c). This accounts for a total of $4IH+6H+I+4$ basic operations to estimate the volume of a single OD-flow.

The number of input neurons I is equal to the length of the path that carries OD-flow k (i.e. the number of traversed links). Path length represents a crucial Traffic Engineering metric and it is highly optimized in every large-scale network, so its value is generally very small. For example, in our datasets, the mean number of links traversed by every OD-flow is below 5.

Whereas the number of inputs and outputs of the RNN are determined by the estimation problem itself, we do not know a priori which is the optimal number of hidden neurons H, and there is no foolproof method for setting it. On the one hand, if we have too many degrees of freedom we will probably have over-fitting problems, because the RNN *memorizes* instead of learning. On the other hand, if we have too few, then the RNN does not have enough *expressive* power to fit the learning data and to capture the underlying model. This issue will clearly depend upon the number of learning patterns T and the complexity of the problem itself, but as a general rule of thumb, the number of hidden neurons should be kept as small as possible.

A convenient heuristic to set H in an ANN is to choose the number of hidden neurons such that the total number of weights is roughly $T/10$ [50]. Such an approach has worked well over a range of practical problems [50]. The number of weights in a RNN is twice the number of weights of an ANN, because of the negative weights. As a consequence, if we apply the same heuristic to a RNN model, we have to choose the number of hidden neurons such that the total number of weights is about $T/5$. A more principled method is to adjust the complexity of the network in response to the learning data, for instance starting with a "large" number of hidden neurons, pruning then the topology until some criterion is achieved [50].

The number of weights in our RNN model is $2H(I + 1)$. Thus, according to the simple rule of thumb previously described we should choose a number of hidden neurons $H = T/10(I+1)$, which results in about 6 or 7 hidden neurons in our datasets. For $I = 4$ and $H = 7$, the number of basic operations involved in the estimation of the volume of a single OD-flow is around 160, a quite negligible computational cost. The reader should note that this cost is almost independent of the size of the network. Indeed, even if the number of OD-flows m and links r increases, the length of the paths in every large-scale network remains small and almost independent of the size



(a) 3-layer FF RNN Block. (b) Basic RNN-TME Block. (c) RNN-TME Block Diagram.

Figure 2.9 — Block diagram of the TME method based on three-layers Feed-Forward (FF) Random Neural Networks. Each OD-flow volume $x_t(k)$ is estimated from link measurements using transfer-block $f_k(\cdot)$ in 2.9(b). Each of these blocks is built from a three-layers FF RNN like the one depicted in 2.9(a). The m blocks are applied in parallel to estimate a complete TM X_t in 2.9(c), using the connection blocks Δ and Π.

of the network. For the estimation of a complete TM, this cost must be scaled by the number of m OD-flows that are estimated in parallel.

To sum up, the computational cost of the RNN based estimation method is $\mathcal{O}(m)$, i.e. the cost is linear with the number of OD-flows to estimate. As we show in section 2.7, this is a paramount advantage w.r.t. current estimation methods, which may have a computational complexity as high as $\mathcal{O}(m^3)$.

Figure 2.9 depicts a block diagram of the TME method based on Random Neural Networks. From now on we shall use the acronym RNN-TME as a reference to this method. The basic building block of the RNN-TME method $f(\cdot)$ in 2.9(b) maps the normalized SNMP measurements U_t into the normalized OD-flow traffic volume z_t, using a 3-layers feed-forward RNN topology 2.9(a). When m of these basic blocks are used in parallel like in 2.9(c), a complete TM X_t can be estimated from SNMP measurements Y_t at every time t. The *connection* blocks $\Delta(R, C)$ and $\Pi(R, C)$ in figure 2.9(c) are simply used to normalize the inputs Y_t and scale the outputs $z_t(k)$ respectively. The block $\Delta(R, C)$ additionally selects the $\boldsymbol{\delta}_k$ links used as input at each block $f_k(\cdot)$, using routing matrix R.

Given that each OD-flow k presents particular characteristics, we do not expect to obtain the same estimation performance from each block $f_k(\cdot)$ in the practice. Indeed, the RNN-TME method presents similar problems to those identified in general TME techniques to correctly estimate OD-flows with small volumes. An additional issue that arises in the RNN-TME method is the inherent dependence on the structure of the routing matrix R. It is easy to see that the learning process of block $f_k(\cdot)$ will be more simple or more complex depending on the way different OD-flows share the different links involved in the estimation, which is determined by R. Consider for example the case of estimating the volume of an OD-flow that traverses one of the

input links without sharing it with any other OD-flow. In this case, the learning of $f_k(\cdot)$ is trivial and does not depend on the particular characteristics of the input data, as all that the RNN model has to learn is to copy at the output a scaled replica of the correct input. We shall exemplify this strong dependence in section 2.7.2.

2.5.4 Stability of RNNs vs ANNs for TME

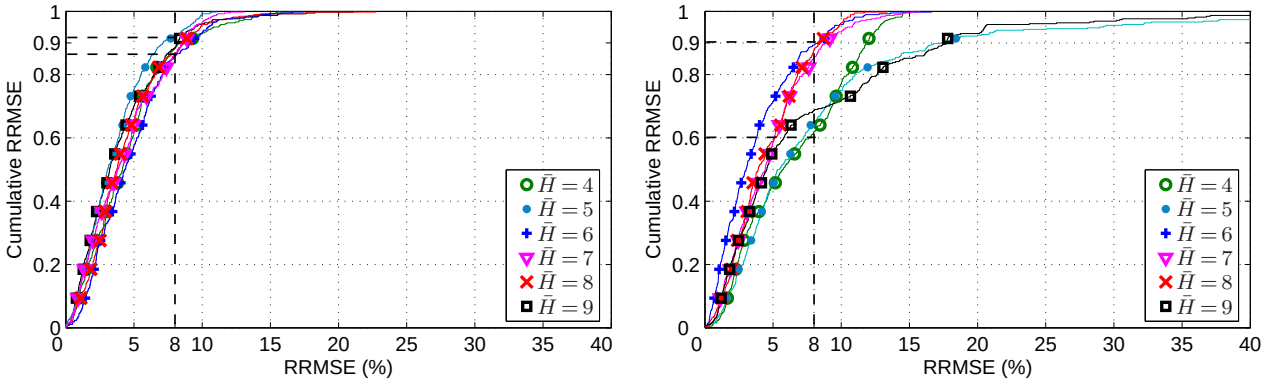
In this section we shall evidence with a real-data evaluation the advantages of using a RNN model for the TME problem instead of a classical ANN model as in [39]. In [44], authors evidence the strong sensitivity of the three-layers feed-forward ANN model to the number of hidden neurons H used in the topology. While this is only verified for their particular application, authors mention many other different applications where the same drawback has been found. As we show below, this sensitivity impacts the quality of results and limits the applicability of the ANN model for TME.

For the evaluation we shall use real traffic and the real network topology of Abilene. As regards the ANN model, we consider a three-layers feed forward topology, using a multilayer perceptron model as in [39]. To be as fair as possible, we use exactly the same learning and estimation schemes previously described for both the ANN and the RNN models.

We take 8 consecutive days of traffic from Abilene and divide it into two disjoint datasets. The learning dataset is used in the calibration of the Neural Network models and it is composed of 24 hours of consecutive measurements, representing a total of 288 patterns. The validation dataset is used to verify the properties of the estimation methods and it is composed of 1 week of traffic, which accounts for 2016 measurements. In order to compare the estimation performance of both models and their dependence on H, we shall use the relative root mean squared error RRMSE(t) previously defined in (2.12).

Figure 2.10 depicts the cumulative distribution of the relative estimation error RRMSE(t) for all the 2016 TMs in the validation dataset, varying the mean number of hidden neurons $\bar{H}$ between 4 and 9 for both the RNN and ANN models. The mean number of hidden neurons $\bar{H}$ is simply defined as the rounded average of hidden neurons H_k used to build each transfer-bloc $f_k(\cdot)$, i.e., $\bar{H} = \text{round}\left(\sum_{k=1}^{m} H_k\right)$. These values for $\bar{H}$ were not chosen by chance, but as a result of the rule presented in section 2.5.3, where the expected number of hidden neurons was around 6.

From figure 2.10(a) we can see that changing the number of hidden neurons around this value has little impact on the RNN model. For example, for a relative error of about 8% (a reasonable value of RRMSE according to previous works [22, 23]), the change in the cumulative distribution of RRMSE is below 6%. On the contrary, figure 2.10(b) shows the important influence of the number of hidden neurons on the quality of the estimation for the ANN model. Indeed, for the same value of RRMSE, the



(a) CDF RRMSE for $\bar{H} = 4$ to 9 with RNNs

(b) CDF RRMSE for $\bar{H} = 4$ to 9 with ANNs

Figure 2.10 — Cumulative distribution of the relative error as function of the mean number of hidden neurons $\bar{H}$ in the Abilene dataset, for (a) the RNN model and (b) the ANN model.

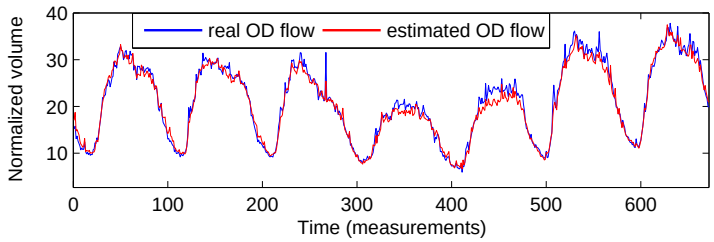
cumulative distribution of the relative error can vary almost a 30% for the TMs in the validation set, even for a change of only 1 hidden neuron, from 5 to 6 or from 8 to 9 for example. This strong sensitivity with respect to the neural topology makes of the ANN-TME an estimation method that has to be highly tuned to obtain proper results, seriously limiting its usefulness.

2.5.5 Evaluation of the RNN-TME method

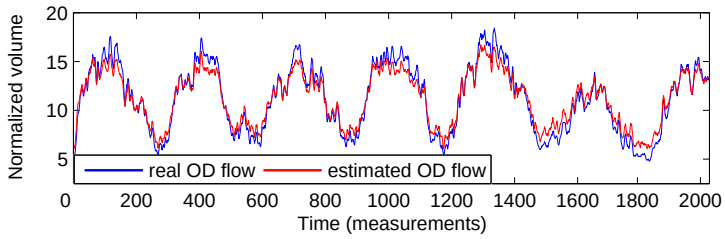
In the evaluation of the RNN-TME method, we consider the estimation of 1 week of OD-flows traffic from two operational networks: the Abilene network and the GEANT network. As we did before, we take 8 consecutive days of traffic from Abilene, using the first 24 hours of measurements (288 patterns) as the learning dataset and the following 7 days (2016 patterns) for validation.

Note that the sampling rate used in the GEANT dataset is three times slower compared to the $5'$ time scale used in Abilene and in many ISP networks to collect SNMP measurements [34]. In order to have the same size of learning datasets in both networks, we take 10 consecutive days of measurements from GEANT, using the first 3 days for learning (288 patterns) and the remaining 7 days for validation (672 patterns). Note however that the same result can be achieved by interpolating intermediate measurements in a 24hs learning dataset. As usual, we assume that traffic flows X_t are just known during the learning period of the RNNs models, and consider the SNMP measurements Y_t as the input known data.

Figure 2.11 depicts the real and estimated values of the normalized volume of a single OD-flow k, namely $z_t(k)$, for 1 week of traffic in 2.11(a) GEANT and 2.11(b) Abilene. In both cases the estimation is accurate and stable during the 7 days of the validation period, and even though there is an evident and important traffic decrease during the weekend, the estimation is still correct. This result is a-priori



(a) Single OD-flow in GEANT



(b) Single OD-flow in Abilene

Figure 2.11 — 1 week of OD-flow traffic volume estimation using the RNN-TME approach for (a) an OD-flow in GEANT and (b) an OD-flow in Abilene.

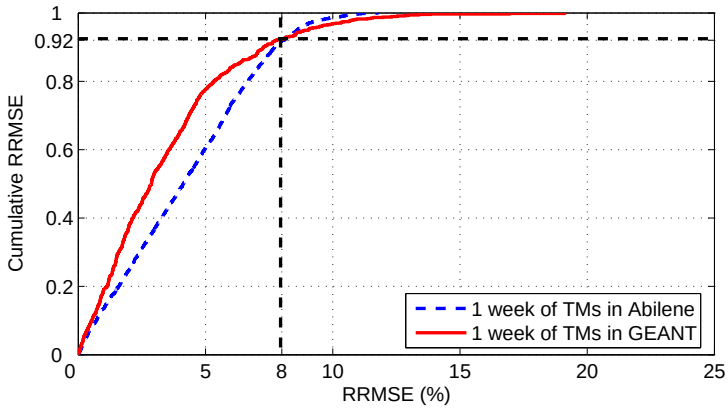


Figure 2.12 — Cumulative distribution of the RRMSE(t) for 1 week of TMs estimated with the RNN-TME approach in GEANT and Abilene.

quite impressive, especially because the learning period is probably not that long so as to cover all the possible input and output cases. However, the key issue in the learning process of each transfer-block $f_k(\cdot)$ is that, in fact, we are not learning any function but one very particular, which strongly depends on the structure of the routing matrix R. Each block $f_k(\cdot)$ is nothing but a particular pseudo-inverse of $R(i,\cdot)$. If we can correctly learn $f_k(\cdot)$ for a certain learning dataset, then this transfer-block should perform correctly even for new data, not seen before. The obvious drawback is that the performance of the method depends on the particular characteristics of R.

Figure 2.12 presents the cumulative distribution of the RRMSE for the validation week in GEANT and Abilene. The relative estimation error for the TM is below 8% for more than 90% of the samples in the validation dataset. The mean values of the

RRMSE for the evaluation period are 3.46% for GEANT and 4.22% for Abilene. As before, taking as benchmark the relative errors obtained in previous works [22, 23], we may conclude that the obtained results are highly satisfactory. In spite of this, we should say once again that these results may be difficult to generalize, due to the particular characteristics of the routing matrix R that was used.

2.6 Principal Components Analysis for TME

In this section we shall describe another mixed TME technique that will be used as benchmark for our methods. The method, introduced in [37], uses a Principal Component Analysis (PCA) technique to reduce the dimensionality of the TM and thus produce a simple least mean squares estimate from SNMP measurements. Evaluations performed in [23] show that the PCA approach is one of the most accurate methods so far proposed in the literature, which justifies our choice.

The PCA method tackles the TM estimation problem by reducing the dimensionality of the TM X_t. PCA is a data-driven dimension reduction technique that captures the maximum energy (or variability) in the data into a minimum set of new axes called *principal components* (PCs). In [37], authors used PCA to study the intrinsic dimensionality of a set of consecutive TMs, and found that the entire set of m OD-flows, when examined over long time scales (days to weeks), can be accurately captured by low dimensional representations, using a reduced number of PCs. Using this low dimensional representation, [23] proposes a simple least mean squares solution to the TM estimation problem.

Formally, we shall define $\mathbf{X}$ as the $p \times m$ matrix of p consecutive TMs, $\mathbf{X} = \{X_1, X_2, \ldots, X_p\}^T$. For technical reasons, we shall consider that the matrix $\mathbf{X}$ is a column-centered matrix, i.e., each column of $\mathbf{X}$ has zero mean. Using PCA, we can express $\mathbf{X}$ in terms of its m PCs $\mathbf{v}_i$, $i = 1, \ldots, m$ (assuming there are no singularities, we shall come back to this issue later):

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{2.45}$$

where $\mathbf{V}$ is a $m \times m$ matrix with PCs $\mathbf{v}_i$ as columns, $\mathbf{D}$ is a $m \times m$ diagonal matrix with non-negative real numbers on the diagonal, and $\mathbf{U}$ is a $p \times m$ matrix. Let us clearly define each of these matrices. PCA consists in a simple change of basis vectors, in such a way that the new basis vectors coincide with the directions of maximal variance in $\mathbf{X}$. The new basis vectors $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ are the PCs of $\mathbf{X}$. Each PC $\mathbf{v}_i$ can be easily computed using its definition. The first PC $\mathbf{v}_1$ is the vector that points in the direction of maximal variance of $\mathbf{X}$, and thus it can be computed as:

$$\mathbf{v}_1 = \arg \max_{||\mathbf{v}||=1} ||\mathbf{X}\mathbf{v}||^2 \tag{2.46}$$

Note that vectors $\mathbf{v}_i$ form the new basis, and for convenience we just consider orthonormal vectors. To compute the second PC $\mathbf{v}_2$, we look for the direction along which residuals variance is maximized. The residual is the difference between the original data $\mathbf{X}$ and the data mapped onto the first PC $\mathbf{v}_1$, i.e., $\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T$. By construction, we restrict the search of $\mathbf{v}_2$ to all directions orthogonal to $\mathbf{v}_1$. Proceeding iteratively, the k-th PC $\mathbf{v}_k$ can be computed as:

$$\mathbf{v}_k = \arg \max_{||\mathbf{v}||=1,\, <\mathbf{v},\mathbf{v}_i>=0} ||(\mathbf{X} - \sum_{i=1}^{k-1} \mathbf{X}\mathbf{v}_i\mathbf{v}_i^T)\mathbf{v}||^2 \tag{2.47}$$

Computing all the principal components of $\mathbf{X}$ is equivalent to finding the eigenvectors of the observed covariance matrix $\mathbf{C_X} = \frac{1}{p}\mathbf{X}^T\mathbf{X}$. Let us assume that $\mathbf{C_X}$ is a full rank matrix. In that case, $\mathbf{X}^T\mathbf{X}$ can be expressed as:

$$(\mathbf{X}^T\mathbf{X})\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{2.48}$$

where $\mathbf{\Lambda}$ contains the m eigenvalues λ_i, corresponding to the m orthonormal eigenvectors $\mathbf{v}_i$ of $\mathbf{X}^T\mathbf{X}$, arranged as columns of the matrix $\mathbf{V}$. Let us now construct the $p \times 1$ vectors $\mathbf{u}_i$, $i = 1 \ldots m$, which correspond to the orthonormal projection of $\mathbf{X}$ onto each eigenvector $\mathbf{v}_i$:

$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{X}\mathbf{v}_i, \text{ with } ||\mathbf{u}_i|| = 1 \tag{2.49}$$

In our particular problem, each element $j = 1, \ldots, p$ of vector $\mathbf{u}_i$, namely $\mathbf{u}_i(j)$, corresponds to the projection of the m OD-flows of the j-st TM $X_j \in \mathbf{X}$ onto eigenvector $\mathbf{v}_i$, and thus we shall name each vector $\mathbf{u}_i$ as *eigenflow* i. If we define $\mathbf{U}$ as the $p \times m$ matrix with eigenflows as columns, and $\mathbf{D}$ as the $m \times m$ diagonal matrix with values $\sqrt{\lambda_i}$ in the diagonal, we can rewrite (2.49) as:

$$\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{D} \tag{2.50}$$

Because $\mathbf{V}$ is orthonormal, we can multiply both sides of (2.50) by $\mathbf{V}^{-1} = \mathbf{V}^T$ to arrive at the final form of the decomposition of $\mathbf{X}$ in its m principal components $\mathbf{V}$, as described in (2.45).

So far we have assumed that matrix $\mathbf{C_X}$ has rank m; however, it might be the case that $\mathbf{C_X}$ has only $r < m$ orthonormal eigenvectors, where r is the rank of the matrix. When the rank of $\mathbf{C_X}$ is less than m, $\mathbf{C_X}$ is said to be singular, which basically means that data lies on a smaller dimension subspace. In this case, the previous analysis we did about the PCA decomposition of $\mathbf{X}$ still remains valid, provided three slight modifications: (i) the first r elements in the diagonal of $\mathbf{D}$ correspond to the square root of the r eigenvalues λ_i, whereas the additional $m - r$ elements in the diagonal are zero; (ii) the first r columns of $\mathbf{V}$ correspond to the r eigenvalues $\mathbf{v}_i$, and the remaining $m - r$ columns are orthonormal vectors used to "fill up" $\mathbf{V}$; finally, (iii) similar to $\mathbf{V}$, matrix $\mathbf{U}$ has $m-r$ additional orthonormal vectors as columns.

The interesting thing about the PCA decomposition in (2.45), is that now it is possible to produce a low dimensional representation of the TMs in $\mathbf{X}$ by only considering the top-k PCs $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$, corresponding to the k directions that capture the largest energy of $\mathbf{X}$, with $k << m$. If we define the $k \times k$ diagonal matrix $\mathbf{D}_k$ as the matrix

with the k largest elements of $\mathbf{D}$, and the $m \times k$ matrix $\mathbf{V}_k$ as the matrix with the corresponding top-k PCs, we can approximate any TM $X_t \in \mathbf{X}$ by:

$$X_t \approx \mathbf{V}_k \, \mathbf{D}_k \, \nu_t \tag{2.51}$$

where the $k \times 1$ vector ν_t has the values of the k most significant eigenflows at time $t = 1, \ldots, p$, $\nu_t = \{\mathbf{u}_1(t), \mathbf{u}_2(t), \ldots, \mathbf{u}_k(t)\}$. With this approximated representation, we have to estimate the value of a reduced number of parameters to compute an estimated TM. In order to estimate the k values of ν_t from the SNMP measurements vector Y_t, we use equations (2.1) and (2.51) to obtain:

$$Y_t \approx F \, \nu_t \tag{2.52}$$

where $F = R \, \mathbf{V}_k \, \mathbf{D}_k$ is a $r \times k$ matrix. Given the small number of columns in F, it is easy to compute a least mean squares estimate of ν_t from SNMP measurements. In fact, equation (2.52) represents now a well-posed estimation problem, and thus we can compute the estimate $\hat{\nu}_t$ simply as:

$$\hat{\nu}_t = (F^T F)^{-1} F^T \, Y_t \tag{2.53}$$

Combining (2.51) and (2.53), we can finally compute the k-Principal Components Analysis Estimate (PCAE-k) for TM X_t:

$$\boxed{\hat{X}_t^{PCAE-k} = \left(\mathbf{V}_k \, \mathbf{D}_k \, (F^T F)^{-1} F^T\right) Y_t} \tag{2.54}$$

Traffic matrix estimation using PCA assumes that both matrices $\mathbf{V}$ and $\mathbf{D}$ are known and stable in time. In [37], authors use a prior set of TMs $\mathbf{X}_o$ obtained by 24 hours of direct OD-flow measurements to compute these matrices, and show that the decomposition remains reasonably stable in time. However, in [23] authors claim that these matrices should be periodically recomputed to provide accurate results in the long term.

In addition, the quality of the PCA estimation depends on the number of PCs k used to build $\mathbf{V}_k$. As it should be evident for the reader, using more PCs provides better results, but increments the dimensionality of the traffic model and the size of the corresponding decomposition matrices, increasing the complexity of the algorithm. As we show in section 2.7, and even if in the general case the intrinsic dimensionality of the TM is low [37], a large number of PCs might be required to estimate a TM in some particular situations.

At first glance, an inattentive reader may believe that the PCAE method and the SMLE approach described in section 2.3 are basically the same; far from being true, there is a crucial difference between both approaches. In the one hand, PCAE is a

completely non-parametric data-driven approach and almost no assumptions about the underlying data model are made. In fact, results highly depend on the particular set of TMs $\mathbf{X}$ used for calibration. On the other hand, the SMLE method completely depends on a parametric model with strong assumptions about the TM, and thus it does not depend on the particular set of TMs used for calibration.

2.7 Comparative Analysis

In this section we present a comparative analysis of the different TME algorithms presented in this chapter, considering not only their estimation accuracy but also some other significant implementation-related issues. We shall analyze two different case-studies. In the former, we evaluate the performance of the methods in a normal-operation scenario, where traffic is free of anomalies. The latter case-study corresponds to an abnormal scenario, where an unexpected and large volume traffic anomaly occurs. Finally, we present an study on the numerical complexity of the algorithms used by each TME method.

2.7.1 TME for Normal-Operation Traffic

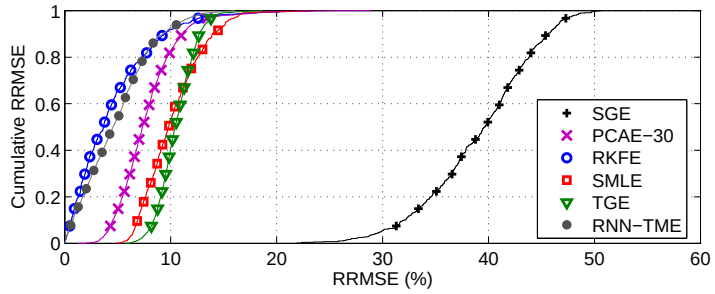
In this case-study we shall consider only anomaly-free traffic, which corresponds to the most usual traffic behavior. The Splines-Based Maximum Likelihood Estimation method (SMLE), the Recursive Kalman Filter Estimation method (RKFE), and the Random Neural Networks TME (RNN-TME) method are compared against the Simple Gravity Estimation method (SGE), the Tomo-Gravity Estimation method (TGE), and the Principal Components Analysis Estimation method (PCAE).

The evaluation is conducted using a validation dataset composed of 672 consecutive TMs from Abilene. The three mixed TME methods, namely the RNN-TME, the RKFE and the PCAE methods, use 24hs of direct OD-flow measurements for calibration purposes, corresponding to 288 TMs gathered exactly before the validation set. The SMLE method uses 1h of SNMP measurements to calibrate the SB model. The TGE and SGE methods do not require calibration. As regards the PCAE method, we consider $k = 30$ principal components to describe OD-flows traffic, and thus we shall use PCAE-30 as a reference to this method.

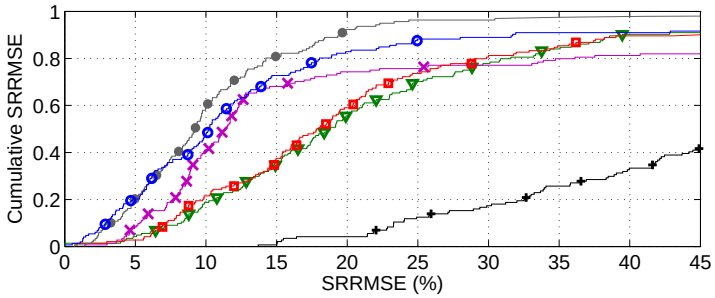
Estimation Method	Mean RRMSE (%)
RKFE	4.48
RNN-TME	4.95
PCAE	6.53
SMLE	10.3
TGE	11.2
SGE	39.1

Table 2.2 — Mean RRMSE values (%) for 672 TMs in Abilene.

Figure 2.13 presents the comparative performance of the 6 TME methods. From figure 2.13(a), we can see that the RNN-TME and the RKFE produce estimation relative errors below 10% for approximately 90% of the TMs. In the case of the PCAE method, approximately 80% of the TMs are estimated with less than 10% of relative error. This result drops to nearly 55% for the SMLE method, 40% for the TGE method, and to 0% for the SGE method.



(a)



(b)

Figure 2.13 — (a) Cumulative RRMSE(t) and (b) Cumulative SRRMSE(k) for 672 measurements in Abilene, for the SMLE, the RKFE, the TGE, the SGE, the RNN-TME and the PCAE methods.

Table 2.2 presents the obtained mean values of the temporal relative error. The performance obtained with the three mixed methods clearly outperforms that obtained with the pure SNMP methods. Our implementations provide similar and sometimes slight better results than those reported in [22, 23]. The supremacy of mixed methods w.r.t. pure SNMP methods seems quite evident, given that mixed methods use better and more rich data for calibration purposes. While temporal results show the accuracy of the TME methods, the reader should remember that in this performance index we are not comparing the small-volume OD-flows. In section 2.3.1 we claimed that in general, the TME methods produce quite poor estimation results for OD-flows with low volume of traffic [21, 23].

In order to unveil this performance issue, we shall analyze the estimation error produced by the TME methods for each single OD-flow of the TM during the complete duration of the validation dataset. For this purpose, we shall introduce the Spatial Relative Root Mean Squared Error for OD-flow k, defined as:

$$\text{SRRMSE}(k) = \frac{\sqrt{\sum_{t \in T_{\text{val}}} \left(x_t(k) - \hat{x}_t(k)\right)^2}}{\sqrt{\sum_{t \in T_{\text{val}}} x_t(k)^2}}, \quad \forall k = 1, \ldots, m \tag{2.55}$$

Different from the temporal error RRMSE(t), which represents a summary of the error produced in the estimation of a TM, the spatial RRMSE summarizes the error produced in the estimation of each single OD-flow over its lifetime.

Figure 2.13(b) depicts the cumulative distribution of the spatial error SRRMSE(k) produced in the estimation of the 132 individual OD-flows, during the 672 consecutive time indexes $t \in T_{\text{val}}$.

The first interesting observation is that spatial errors are fairly more spread than temporal errors, which is not surprising at all. Given that traffic variations in the validation dataset correspond mainly to normal operation traffic, the errors produced at different instances of the TM are quite similar and independent of the particular time of evaluation. On the other hand, it seems quite reasonable to accept that some OD-flows are more difficult to estimate than others, due basically to the different characteristics of each particular OD-flow and the particular features of each TME method. Note that in all cases, many OD-flows are simply not possible to be estimated with a reasonable error.

The second interesting conclusion that can be drawn from figure 2.13(b) is that, despite the poor results that are obtained for many OD-flows, the RNN-TME estimation method outperforms the rest of the estimation algorithms, even for low-volume OD-flows. Indeed, the RNN-TME method produces spatial relative errors smaller than 20% for about 90% of the 132 OD-flows. This is a direct consequence of the learning technique used in this estimation method, where a single RNN is trained for each single OD-flow, capturing its particular characteristics.

2.7.2 TME in the Presence of Volume Anomalies

In the second case-study, we shall study the performance of the different TME methods to estimate an OD-flow in the presence of volume anomalies. Volume anomalies represent large and abrupt traffic variations due to unexpected events. Figure 2.14 depicts the normalized traffic volume of a single OD-flow that experiences a brutal traffic volume augmentation due to a BGP reroute in Abilene. Before time 370 traffic belongs to normal operation and little traffic flows in this particular OD-flow, but after this time a BGP egress-point shift causes traffic from other OD-flows with the same origin node to suddenly shift towards the same destination node, causing a sustained volume increase during about 18hs until time 580, when normal operation is regained for this OD-flow.

Similar to previous evaluations, we use the first 24hs of direct OD-flow measurements X_t (i.e. $T_{\text{learn}} = \{1, .., 288\}$) for training and calibration of the RNN-TME method, the RKFE method, and the PCAE method, when traffic follows a normal operation behavior. The real value of X_t is only assumed to be known during this training period, $\forall t \in T_{\text{learn}}$. As we explained before, we use a single hour of SNMP measurements Y_t to calibrate the SMLE method, considering the traffic from the first hour of the evaluation. The SGE and the TGE methods are directly applied.

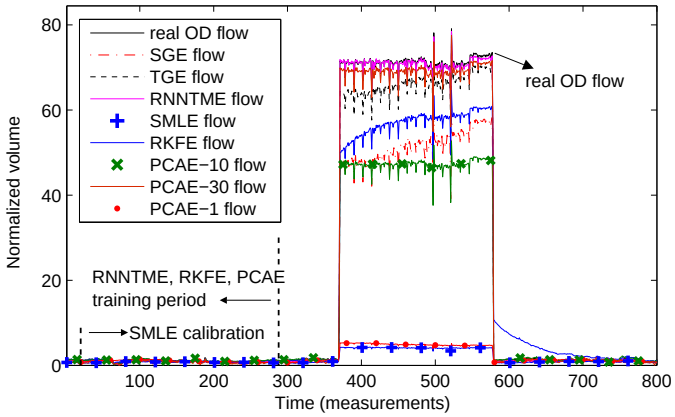


Figure 2.14 — Normalized OD-flow volume estimation under a large volume variation due to a BGP egress-point shift.

From figure 2.14 we can see that all methods can properly track the OD-flow volume evolution before the occurrence of the anomaly. However, only the RNN-TME and the PCAE-30 methods achieve a proper estimation of the anomalous OD-flow traffic volume after time 370. The RNN-TME method obtains a mean relative estimation error of 1.3% during the occurrence of the anomaly, between times 370 and 580. On the other hand, the PCAE-30 method incurs in a mean relative estimation error of 3.2%. The reason for this success relies on the the fact that no traffic model is assumed by both the RNN-TME and the PCAE approaches.

Very interesting is the tracking power achieved by the RNN-TME method, which is not a-priori justified, especially because no anomalous traffic patterns were present in the learning step. This evaluation permits to exemplify the strong dependence of the RNN-TME method on the particular structure of R that we mentioned before. In fact, one of the links that are crossed by the anomalous OD-flow is only used by this particular OD-flow, and as we have explained before, the learning process results in this case in simply setting the RNN weights so as to copy at the output a scaled copy of the corresponding input. As regards the accuracy obtained by the PCAE method, we can simply affirm that the analyzed OD-flow was captured in the sub-space defined by the first 30 principal components.

The rest of the estimation methods impose certain assumptions on the underlying traffic characteristics that are clearly modified in the event of a volume anomaly. For example, the recursive and adaptive estimation of the anomalous traffic volume produced by the RKFE method shows that the transition matrix calibrated under normal operation conditions is no longer appropriate during the anomaly. The RKFE estimation accurately converges before and after the anomaly, but presents a large convergence gap during its occurrence, producing a mean relative estimation error of 26.1%. The SGE and TGE methods rely on a gravity stability assumption that does not hold during the anomaly, and thus they produce a mean relative error of 29.7% and 10.5% respectively.

As we have previously discussed in section 2.6, the performance of the PCAE-k method highly depends on the number of principal components k used to describe the TM, which is an important drawback of the approach. In this particular evaluation we can clearly appreciate this dependence. The mean relative error multiplies by a factor of 10 when changing k from 30 to 10, going from 3.2% to 32.7%. An interesting result is obtained when using only 1 PC, as the anomaly goes almost undetected by the PCAE-1 method. This clearly suggests that the volume anomaly occurs in one OD-flow that is not described by this single PC. It is not surprising that the SMLE method produces a very similar result, given that the underlying SB traffic model is only valid for anomaly-free traffic.

With this simple evaluation we can gain an intuition on how these traffic models will be used for anomaly detection in the following chapter. Even though in this chapter we are using the SB traffic model to estimate a TM, the original objective of the model is to produce traffic residuals sensitive to volume anomalies. These traffic residuals are simple the traffic obtained after removing the anomaly-free traffic, correctly described by the SB model. We shall come back to this issue in chapter 3.

2.7.3 Numerical Complexity

In this section we propose to analyze and compare the computational complexity of each TME algorithm, measured in terms of the number of basic operations (i.e., sums and products) involved in the estimation of a complete TM. The main idea of this analysis is to study the scalability of each estimation method as regards the size of the network, basically defined by the number of OD-flows m to estimate. The sampling frequency of SNMP measurements is in the order of some minutes, generally between $5'$ and $10'$, and so the algorithms must be capable of producing an estimate in the order of the minute. The learning cost is not considered in this analysis, mainly because we assume that the calibration of the algorithms represents an off-line task. Table 3.3 summarizes the computational complexity of each algorithm.

Let us begin by the Simple Gravity and Tomo-Gravity methods. The SGE method uses the total aggregated traffic entering at and leaving from every edge node in the network to compute an estimate of each OD-flow volume from (2.2); the total aggregated traffic at each node is provided by SNMP measurements, and thus the total cost of the SGE method is in the order of m^2 basic operations in the worst case. The TGE method uses a Gravity estimate as a point of departure, computing an additional pseudo-inverse matrix R_{inv} to improve results. Using a simple SVD approach, this computation implies rm^2 basic operations. The pseudo-inverse matrix computation can be performed off-line and then be used for TME, so to be as fair as possible we will not consider the associated cost. Finally, the projection of the residual Y_t' onto R_{inv} involves a total of mr operations. Therefore, we can say that the total cost of the TGE method is also $\mathcal{O}(m^2)$.

Method	n^{o} basic operations
SGE	$\mathcal{O}(m^2)$
TGE	$\mathcal{O}(m^2)$
SMLE	$\mathcal{O}(rqm)$
RKFE	$\mathcal{O}(m^3)$
PCAE	$\mathcal{O}(rkm)$
RNN-TME	$\mathcal{O}(m)$

Table 2.3 — Computational complexity of the different TME algorithms. The number of operations corresponds to the estimation of a complete TM with m OD-flows and r links, and it does not include the operations involved in the learning/calibration of the methods.

From equation (2.11), the estimation of a TM with the SMLE method involves approximately $rq^2 + qr^2 + rqm + rm$ basic operations, consisting of exclusively matrix multiplications, and thus its computational cost is $\mathcal{O}(rqm)$. The RKFE method implies to update the Kalman gain, the estimation covariance error and the residual error. This involves matrix multiplications and inversions, and in the worst case these are $m \times m$ square matrices; thus the associated cost of the RKFE approach is $\mathcal{O}(m^3)$.

The operations involved in the PCAE method are quite similar to those in the SMLE method, and the associated cost is $\mathcal{O}(rkm)$. Finally and according to section 2.5.3, the computational cost of the RNN-TME method is $\mathcal{O}(m)$.

From this simple analysis we can see that the RNN-TME method is by far the fastest and the least constrained algorithm regarding scalability issues and on-line operation. The RNN-TME approach has an estimation cost which is linear in m. The SGE and TGE methods have similar computational cost among them, quadratic in m. The SMLE and PCAE methods have a computational cost which we can say is somehow linear in m. This is due to the fact that, even if the number of OD-flows m may grow high in a large-scale network, the number of links r generally grows at a much slower rate. Additionally, both methods consists in parsimonious representations, hence q and k are generally very small w.r.t. m and r. Finally, the RKFE method is the most expensive in terms of number of operations, having a cost which is almost cubic in m.

2.8 Conclusions

In this chapter we have proposed and analyzed different traffic models for a Traffic Matrix, particularly using aggregated data to study its behavior. This analysis led us to study the well-known Traffic Matrix Estimation problem. Probably one of the most interesting conclusion of this chapter is that we have shown that the field of Traffic Matrix Estimation can still be improved with more accurate, faster and more stable techniques as the ones we have presented, encouraging the development and implementation of new techniques to lighten routers tasks in the future, offering other possibilities to networks operators.

In the first part of the chapter we have introduced a novel parsimonious traffic model that correctly captures the anomaly-free behavior of the TM. The model assumes quite strong hypotheses about the characteristics of the different OD-flows that compose the TM, and even though they are quite difficult to verify in a general network scenario, we have shown with real data from operational networks that the model is accurate enough so as to provide a proper picture of the TM from SNMP measurements. The SMLE method that was developed using this model presents estimation results comparable to those provided by the well known and highly accepted Tomo-Gravity Estimation approach, but with a paramount advantage, that of using a linear model to describe OD-flows traffic. As we explained before, this will allow us to develop anomaly detection algorithms with robust mathematical properties in the following chapter, something difficult to achieve with the TGE method.

In the second part of the chapter we have conducted an in-depth analysis of a recursive TME method developed in previous works, based on the use of Kalman filters. This study allowed us to better understand some drawbacks and conception problems of the original approach, for which we have proposed simple yet effective improvements. These improvements provide not only more accurate results, but also a more stable operation, which comes directly from the flexible model used in the recursive estimation.

In the last part of the chapter we have presented a TME method based on statistical learning techniques, more precisely, based on a new type of neural network known as the Random Neural Network. We showed that this kind of neural network provides more robust results for TME, compared to a traditional Artificial Neural Network model. The RNN-TME method has shown to be particularly attractive to estimate low-volume OD-flows, a task that other techniques can not generally achieve due to large errors. Finally, we have shown that the method outperforms previous works in terms of computational complexity, a paramount advantage when considering large-scale networks and scalability issues.

Regarding the comparative analysis between the different TME methods, it is very hard to assert whether there is a single method that outperforms the others or not. In fact, probably the best solution would be to consider different TME methods, depending on the networking context and the particular task. For instance, evaluations showed that the RNN-TME method is more accurate to estimate both large and small volume OD-flows than the benchmark approaches. However, the learning step is much more complex than in other methods, becoming even overkilling if routing updates are often. Additionally, we saw that the method is highly dependent on the particular structure of the routing matrix, an undesirable side-effect of statistical learning applied to this problem.

The SMLE method provides proper results with a very short learning step, and does not require direct OD-flow measurements for calibration. This makes it particularly interesting for networks without flow measuring technology or with a dynamic routing scheme (note that in large-scale networks, dynamic routing may say routing updates every 12hs, i.e. long-time scales). However, the SMLE approach relies on a parametric traffic model with strong assumptions that may not be verified in every network. The RKFE method also provides very good estimation results, and its learning step is much simpler than in RNN-TME, but its computational cost may impose scalability issues in very large-scale networks.

As we explained before, the Splines-Based model was not originally aimed for TME, but for Anomaly Detection. In the next chapter we exploit the SB traffic model in this direction, constructing anomaly detection and localization algorithms with optimality properties on top of it.

CHAPTER 3

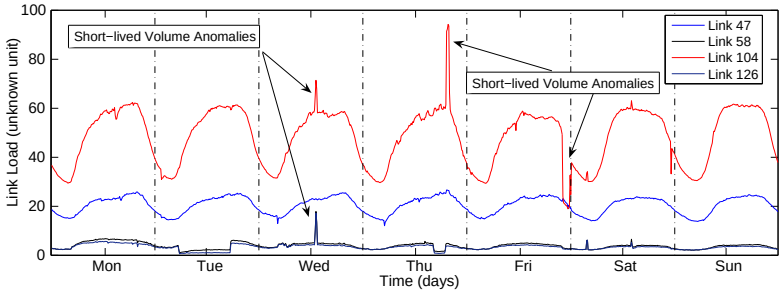
Optimal Anomaly Detection and Localization

Despite the massive growth of end-user applications and access bandwidth in the near future, the Internet is not going to collapse under the weight of future traffic volume. However, according to the study provided by Cisco's global IP traffic forecast for the years 2006 until 2011 [3, 4], one of the most difficult challenges for network operators will be to correctly manage the large and unexpected congestion problems at the core network caused by volume anomalies.

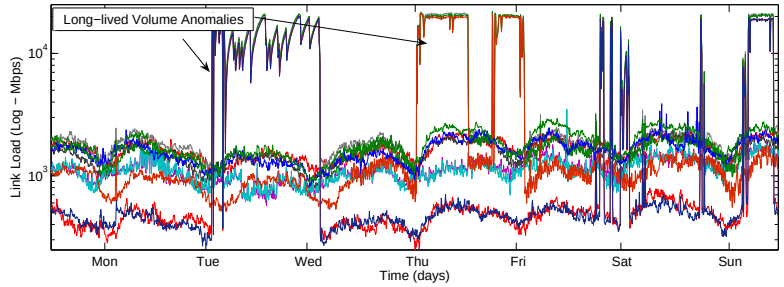
Volume anomalies represent large and sudden variations in OD-flows traffic. These variations arise from unexpected events such as flash crowds, network equipment failures, network attacks, and external routing modifications among others. Besides being a major problem itself, a volume anomaly may have an important impact on the overall performance of the affected network. Large-scale monitoring systems are currently deployed in ISP and large enterprise networks to fight back against these unexpected events. In this chapter, we focus on two central aspects of traffic monitoring for volume anomaly detection: (i) the rapid and accurate detection of volume anomalies, and (ii) the localization of the origins of the detected anomalies.

The first issue corresponds to the anomaly detection field, a difficult and extensively studied problem. Anomaly detection in data networks consists of identifying patterns that deviate from normal traffic behavior, thus it is intimately related to traffic modeling. In order to detect abnormal behaviors, accurate and stable traffic models should be used to describe what constitutes an anomaly-free traffic behavior. This is indeed a critical step in the detection of anomalies, because a rough or unstable traffic model may completely spoil the correct detection performance and cause many false alarms.

As we claimed in the Introduction of the thesis, different types of network anomalies can be detected depending on the monitored data. In this chapter we focus on network-wide volume anomaly detection, analyzing network traffic at the TM level. In particular, we use SNMP measurements to detect volume anomalies in the OD-flows that compose a TM, aiming to conceive light monitoring algorithms. As we showed in chapter 2, this is a challenging task, simply because the TM is not directly observable from these coarse-grained data.



(a) Tier-2 ISP network



(b) Abilene network

Figure 3.1 — Network volume anomalies in large-scale IP networks.

The TM is a volume representation of OD-flows traffic, and thus the types of anomalies that we can expect to detect from its analysis are volume anomalies. Figure 3.1 depicts the occurrence of short-lived (a couple of hours at most) and long-lived volume anomalies in 3.1(a) four monitored links from a commercial international Tier-2 network and 3.1(b) several links from the Abilene network. As each OD-flow typically spans multiple network links, a volume anomaly in one single OD-flow is simultaneously visible on several links. As we will see, this multiple evidence can be exploited to localize the anomalous OD-flows.

The algorithms that we develop in this chapter use the linear and parsimonious Splines-Based traffic model presented in chapter 2 to model the anomaly-free traffic behavior of the TM and to overcome the observability problems that arise from using SNMP measurements to analyze the TM. The linear SB model has another major virtue that will be exploited throughout this chapter: it allows to treat the anomaly detection problem as a change detection problem with nuisance parameters, represented in practice by the anomaly-free traffic. This permits to design optimal algorithms using the principles of decision theory.

Optimality support is fundamental in the conception of general algorithms, not tied to any particular network and more importantly, independent of individual evaluations in particular network and traffic scenarios. In-house methods may work rather well in certain scenarios, but without a principled and generalizable support they can be easily rebutted; this is in fact the case of the most celebrated anomaly detection and localization approach, as recently evidenced in [76].

The second issue that we address is the localization of the origins of a detected anomaly. This belongs to a more general field known as fault localization or fault *isolation*. The localization of an anomaly consists in inferring the exact location of the problem from a set of observed anomaly indications. This represents another critical task in network monitoring, given that a correct localization may represent the difference between a successful or a failed countermeasure.

In this chapter we shall assume that traffic anomalies are exogenous unexpected events (flash crowds, external routing modifications, external network attacks) that *significantly* modify the volume of one or multiple OD flows within the monitored network. For this reason, the localization of the anomaly consists in finding the OD-flows that suffer such a variation, referred from now on as the *anomalous* OD-flows. The method that we develop locates the anomalous OD-flows from SNMP measurements, taking advantage of the multiple evidence that these anomalous OD-flows leave through the traversed links.

To provide strong evidence on the effectiveness of the proposed methods, all the algorithms are validated using real traffic data from the three different backbone networks used in chapter 2: the Internet2 Abilene backbone network, the European GEANT academic network, and a commercial international Tier-2 network. Additionally, we compare our algorithms against well-known works in the field, showing that similar or even better performance can be achieved with thorough theoretical foundation.

The remainder of this chapter is organized as follows. In Section 3.1 we present the state of the art in the field of anomaly detection and localization in IP networks. Section 3.2 presents a brief taxonomy of the different volume anomalies likely to arise in a large-scale network. In section 3.3 we design a volume anomaly detection algorithm for optimal detection, maximizing the correct detection rate for a bounded false alarm rate. Section 3.4 presents a recursive algorithm for simultaneously detecting and locating volume anomalies in a particular anomalous OD-flow within the TM, minimizing the maximum mean detection/localization delay for given bounds in the false localization and false alarm rates. In Section 3.5 we describe two well-known anomaly detection and localization methods proposed in the literature that will be used as benchmark for our algorithms. The former of these methods is based on the PCA technique described in section 2.6, the latter uses the recursive Kalman filtering technique previously described in 2.4 to detect volume anomalies. In section 3.6 we present the evaluation of the detection/localization algorithms, comparing their performance against the two benchmarking methods in different networks. Section 3.7 discusses complexity and implementation issues of the proposed algorithms. Finally, section 3.8 concludes the chapter.

3.1 State of the Art

The problem of anomaly detection in IP networks has been extensively studied during the last decade; the rich literature available to date is undeniable evidence of this. However, the field still represents a fertile research area due to its high complexity and to the lack of optimal and general solutions to the problem.

The anomaly detection literature treats the detection of general anomalous traffic behaviors [61, 72, 73, 75, 78, 80] as well as specific kinds of network and traffic anomalies. A basic list includes flash crowd events [59, 60], network failures [53, 54, 55, 62, 77], network attacks [63, 65, 66, 79], and large traffic shifts [67] among others. Many of these works operate on individual and independent time series, analyzing traffic at a particular network link, particular device readings or particular packet characteristics with classical forecasting and outliers analysis methods.

For example, [62] uses exponential smoothing EWMA (Exponentially Weighted Moving Average) and Holt-Winters forecasting techniques to detect anomalous behaviors in router readings. [61] analyses frequency characteristics of flow traffic and SNMP measurements using a wavelets based filtering approach, exposing anomalies as sharp variations in the filtered data variance. [64] builds compact summaries of flow traffic data using the notion of *sketch*, applying then the same forecasting techniques used in previous works (ARIMA, Holt-Winters, etc.) on top of such summaries to detect significant forecast errors. [63] uses spectral analysis techniques over TCP flows for DoS (Denial of Service) detection, using traffic traces from a single network link. [67] uses BGP and SNMP data to detect large traffic shifts, using EWMA, seasonal analysis and Holt-Winters over single time series to filter-out periodic and trend components, detecting anomalies as impulse functions. [59] characterizes flash crowds in Web servers and provides a network aware clustering approach to distinguish these events from DoS attacks, proposing an adaptive CDN (Content Delivery Network) architecture to fight back against these extreme events.

[77] represents one of the first papers that uses multiple time series for anomaly detection, synthesizing information from multiple MIB variables at a single router to improve results. Contrary to these works, we treat the anomaly detection problem from a network-wide perspective, exploiting spatial correlations across the time series of traffic from all the links of a network.

Network-wide anomaly detection has also been treated in different works [72, 73, 74, 75, 78, 80]. The methods proposed in [73, 74, 75] make use of rich IP-flow and packet data to detect anomalies, but this data that can be too costly to collect and to process [13]. [72] detects and classifies anomalies by jointly analyzing the distribution of OD-flows and traffic features like IP addresses and ports. Authors use PCA and the subspace method to analyze the ensemble of OD-flows and the corresponding traffic features in a network. The subspace approach is not new

and was firstly introduced in the field of fault diagnosis for chemical engineering processes [68, 69]. Authors in [74] use the idea of sketch proposed in [64] and the PCA approach to identify anomalous traffic flows. [75] proposes a recursive method to detect anomalies in multivariate time series, using the number of packets and the number of individual IP-flows aggregated in a TM as input data. On the contrary, our methods make use of easy to collect coarse-grained SNMP link measurements to detect and locate volume anomalies in OD-flows.

The use of SNMP measurements to detect volume anomalies in OD-flows has been considered in [72, 78, 80], but none of these works has provided a complete and reliable solution to the problem. [78] uses a Kalman-filtering approach to track the evolution of OD-flows from SNMP measurements, detecting anomalies as large prediction errors. The method requires a long training phase where direct anomaly-free OD-flow measurements are used to calibrate the underlying model. As we explained in chapter 2, the assumed model has a particular structure that may require several periodical recalibrations to provide reliable results, which makes the method too costly to implement from a practical point of view. Besides, the paper does not tackle the anomaly localization problem.

Only [72, 80] treat the problem of both anomaly detection and localization in OD-flows from SNMP measurements. Authors in [72] use the PCA approach and the subspace method proposed in [68, 69] to separate SNMP measurements into a normal subspace and an anomalous subspace, where anomalies are detected. The use of the PCA technique and the subspace method has probably become the most famous approach for network-wide anomaly detection in recent years. However, the approach is a pure data-driven in-house method, and recent works [75, 76] have shown categorical evidence about its serious shortcomings for anomaly detection and localization in data networks.

Finally, our approach falls into the same category as [80], where anomalies are inferred from SNMP measurements by combining network tomography and anomaly detection techniques. Authors in [80] use similar methods to those applied in previous works to detect volume anomalies in OD-flows: Fourier and Wavelet analysis, ARIMA modeling, and PCA decomposition. The localization of anomalies is performed via different heuristics which are not evaluated from a complexity perspective and that might be too time-consuming for on-line application. In fact, all evaluations performed in [80] are conducted off-line over individual datasets. Unlike [80], we provide detection and localization algorithms that can be applied in an on-line fashion with solid theoretical support on their optimality properties, a feature absent in the previously described works.

3.2 Network Volume Anomalies Taxonomy

In this section we present a brief taxonomy of the different anomalies than can be encountered in any large-scale IP network and that we intend to detect with our methods. Table 3.1 presents a basic classification of these anomalies, according to [71, 78].

Traffic anomalies can be clustered by their nature in four different categories: (i) unusual end-user behavior (flash crowds, alpha flows), (ii) malicious end-user behavior (DoS, DDoS, port/network scans, worms propagation), (iii) network failures and (iv) external traffic engineering effects. Since we propose to analyze network traffic at the TM granularity, which is basically a volume representation of traffic, the first distinction that we consider is about volume anomalies. Volume anomalies consist in large and sudden traffic augmentations that are visible at the OD-flow traffic granularity.

Port and network scanners are software applications designed to probe a network host or group of hosts for open ports, which could eventually be compromised by an attacker. Worms consists of self-propagating code that spreads across a network by exploiting security flaws. Both kinds of network attacks are generally invisible at an OD-flow traffic resolution, and thus they will not be considered in the list of anomalies that we intend to detect.

Flash crowds and alpha flows correspond both to unusual, non-malicious end-user behaviors. A flash crowd is an unusual and large demand for a resource or service in the network. It generally occurs when a web site catches the attention of a large number of people and gets an unexpected and overwhelming surge of traffic. Flash crowds can be predictable, for example in the event of a scheduled football game or a software release, or simply unpredictable, like in breaking-news events. A flash crowd is characterized by multiple instances of large OD-flows, directed to a single destination. Alpha flows are represented by large transfers of data at unusual high rates from a single origin to a single destination, thus involving a single OD-flow. Bandwidth-measurement experiments conducted in research networks are instances of alpha flows.

Denials of Service (DoS) and Distributed DoS (DDoS) are flooding attacks against a single destination. These attacks may involve either a single source of attack, which corresponds to a DoS attack, or many distributed sources of attack, corresponding to a DDoS attack. Distributed attacks occur when a large group of end-hosts are compromised by an attacker (usually by means of network scanners and worms) who commands them to jointly flood a victim.

Anomaly	Volume	Unusual	Malicious	TE	Failures
Flash Crowd	•	•			
Alpha Flow	•	•			
DoS	•		•		
DDoS	•		•		
Scan			•		
Worm			•		
Outages	•				•
Ingress Shift	•			•	
Egress Shift	•			•	

Table 3.1 — Different Network and Traffic Anomalies in IP Networks.

Network failures like link or node failures and outages generally cause a volume drop in the affected OD-flows. In dynamic load-balancing or internal routing reconfiguration scenarios, an outage may also cause strong congestion situations in the network, because many OD-flows have to be re-routed on the same network links.

An egress shift occurs when the destination of an OD-flow moves from one node to another. This can happen if there is a change in a BGP peering policy, or even a failure, as many OD-flows can have multiple possible exit points from an ISP. Policy changes can also cause a shift of ingress point for a particular destination.

The algorithms that we develop in the following sections do not intend to classify the different kinds of volume anomalies that we have described. From now-on, we assume that volume anomalies are represented by a sudden and large traffic augmentation $\boldsymbol{\varphi}$ in one or more OD-flows of the TM X_t, and design methods to detect them from SNMP measurements Y_t.

3.3 Optimal Volume Anomaly Detection

In this chapter we shall use the parsimonious SB model previously introduced in chapter 2 to remove the contribution of the anomaly-free traffic into the SNMP measurements vector Y_t, producing residuals sensitive to volume anomalies. We shall treat the detection of volume anomalies as a change detection problem with a nuisance parameter, represented by the anomaly-free traffic. This allows to infer anomalies in X_t directly from Y_t, without the need of a previous TME step. This approach improves accuracy and reduces detection lags, because it does not drag possible errors from previous steps.

The goal of the proposed detection algorithm is to detect the presence of an additive anomaly $\boldsymbol{\varphi}$ in one or more OD-flows of the TM X_t from the SNMP measurements vector Y_t, with the highest probability of detection for a given upper bounded probability of false alarm. The detection of this anomalous variation can be treated as a hypothesis testing problem, considering two possible traffic situations or hypotheses: the null hypothesis $\mathcal{H}_0$, where OD-flows are anomaly-free, and the alternative hypothesis $\mathcal{H}_1$, where OD-flows present a volume anomaly and thus traffic is no longer characterized by the anomaly-free-traffic model (2.8). For every new SNMP measurement, the method has to choose between $\mathcal{H}_0$ and $\mathcal{H}_1$ with the "best detection performance". We shall explain below what do we mean by best detection performance.

In order to continuously adapt the decision thresholds of the method, the previously introduced anomaly-free SB traffic model is slightly modified, explicitly considering the temporal variation of the covariance matrix Σ. The Gaussian noise $\boldsymbol{\xi}_t$ is now assumed to have a covariance matrix $\gamma_t^2\Sigma$; the matrix $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$ is assumed to be known and stable in time. The scalar γ_t is unknown and serves to model the mean level of OD-flows volume variance. Considering equation (2.9), the previous hypothesis testing problem can be formulated as follows:

$$\mathcal{H}_0 = \{Z \sim \mathcal{N}(\boldsymbol{\varphi} + H\boldsymbol{\mu}, \gamma_t^2\, I);\ \boldsymbol{\varphi} = 0,\ \boldsymbol{\mu} \in \mathbb{R}^q\} \tag{3.1}$$

$$\mathcal{H}_1 = \{Z \sim \mathcal{N}(\boldsymbol{\varphi} + H\boldsymbol{\mu}, \gamma_t^2\, I);\ \boldsymbol{\varphi} \neq 0,\ \boldsymbol{\mu} \in \mathbb{R}^q\} \tag{3.2}$$

where $\boldsymbol{\varphi}$ represents an anomaly. Note that we have intentionally removed the time index t from Z and $\boldsymbol{\mu}$, explicitly stating that the test is applied for a single measurements vector $Z = Z_t$ at a certain time t.

In the anomaly detection problem, the modeled anomaly-free traffic $\boldsymbol{\mu}$ is considered as a nuisance parameter since (i) it is completely unknown, (ii) it is not necessary for the detection, and (iii) it could possibly mask the anomalies. In order to remove the nuisance parameter from the detection problem, the standardized measurements vector Z is projected onto the left null space of H, using the projection matrix $P_H^{\perp} = I - H(H^TH)^{-1}H^T$. Briefly speaking, we remove the "interference" of $\boldsymbol{\mu}$ from the problem. For this reason it is possible to chose between $\mathcal{H}_0$ and $\mathcal{H}_1$, provided that the projection of the anomaly $\boldsymbol{\varphi}$ onto the left null space of H is nonzero. For example,

suppose that a volume anomaly of size θ occurs in OD flows j and k; then it is easy to see that $\boldsymbol{\varphi} = \Phi^{-\frac{1}{2}}\,\mathbf{r}\,\theta$, where $\mathbf{r}$ stands for the sum of the normalized columns $\mathbf{r}_j$ and $\mathbf{r}_k$ of the routing matrix R.

The quality of a statistical test is defined by the false alarm rate and the power function. The aforementioned testing problem is difficult to analyze, because (i) $\mathcal{H}_0$ and $\mathcal{H}_1$ are composite hypotheses and (ii) there is an unknown nuisance parameter $\boldsymbol{\mu}$. A composite hypothesis refers to a statistical hypothesis that does not completely specify the probability distribution of the test statistic, i.e. it does not reduce to a single point into the probability space. There is no general way to test between composite hypotheses with a nuisance parameter. Therefore, we shall consider the following evaluation approach [84].

Let K_α be the class of tests $\phi(Z) : \mathbb{R}^r \mapsto \{\mathcal{H}_0, \mathcal{H}_1\}$ with an upper bounded maximum false alarm probability, $K_\alpha = \{\phi \ : \sup_{\boldsymbol{\mu}} \Pr_{\boldsymbol{\varphi}=0,\boldsymbol{\mu}}(\phi(Z) = \mathcal{H}_1) \leqslant \alpha\}$, with $0 < \alpha < 1$. The probability $\Pr_{\boldsymbol{\varphi}=0,\boldsymbol{\mu}}$ stands for the measurements vector Z being generated by the distribution $\mathcal{N}(H\boldsymbol{\mu}, \gamma_t^2\, I)$, and α is the prescribed upper bound for the probability of false alarm. The power function or hit rate is defined by the probability of correct detection $\beta_\phi(\boldsymbol{\varphi}, \boldsymbol{\mu}) = \Pr_{\boldsymbol{\varphi}\neq 0,\boldsymbol{\mu}}(\phi(Z) = \mathcal{H}_1)$. A priori, the power function depends on the parameter $\boldsymbol{\varphi}$ as well as on the nuisance parameter $\boldsymbol{\mu}$, which is highly undesirable.

In our work we use the statistical test $\phi^* : \mathbb{R}^r \mapsto \{\mathcal{H}_0, \mathcal{H}_1\}$ of [91, 85], inspired by the fundamental paper of Wald [84]. To solve this problem, Wald [84] proposes a test $\phi^*(\cdot) \in K_\alpha$, which has uniformly best constant power (UBCP) in the class K_α over a certain family of surfaces $\mathcal{S}$. The adaptation of Wald's theory to the problem with nuisance parameters in the case of problem (3.1) - (3.2) has been done in [91, 85] by using the theory of invariant tests. Here, the family of surfaces of constant power $\mathcal{S} = \{S_c : c \geq 0\}$ is defined by $S_c = \{\boldsymbol{\varphi} : \|P_H^\perp \boldsymbol{\varphi}\|^2 = c^2\}$. The UBCP invariant test realizes the best possible constant power $\beta_{\phi^*}(\boldsymbol{\varphi}, \boldsymbol{\mu}) = \beta_{\phi^*}(\boldsymbol{\varphi}', \boldsymbol{\mu}), \forall \boldsymbol{\varphi}, \boldsymbol{\varphi}' \in S_c$ and $\beta_{\phi^*}(\boldsymbol{\varphi}, \boldsymbol{\mu}) \geqslant \beta_\phi(\boldsymbol{\varphi}, \boldsymbol{\mu})$ over the tests $\phi \in K_\alpha$ with a given false alarm rate α. Finally, the threshold λ_α is chosen to satisfy the false alarm rate α, $\Pr_{\boldsymbol{\varphi}=0,\boldsymbol{\mu}}(\Lambda(Z) \geqslant \lambda_\alpha) = \alpha$, where $\Lambda(Z)$ is defined in (3.3).

The test $\phi^*(\cdot)$ decides between $\mathcal{H}_0$ and $\mathcal{H}_1$ with the best detection probability for a bounded false alarm rate, which represents the major advantage of our approach. The test is designed as follows, where $\|\cdot\|$ represents the Euclidean norm:

$$\phi^*(Z) = \begin{cases} \mathcal{H}_0 \ \text{if} \ \Lambda(Z) = \|P_H^\perp Z\|^2 / \gamma_t^2 < \lambda_\alpha \\ \mathcal{H}_1 \ \text{else} \end{cases} \tag{3.3}$$

As we show in section 3.6, this strong theoretical support also has a major impact in practice, providing results that outperform previous proposals. The Optimal Spline-Based Detection method developed in this section will be referred to as the OSBD method in the rest of the thesis.

3.4 Optimal Sequential Volume Anomaly Detection and Localization

In this section we introduce an optimal volume anomaly detection algorithm that has also the ability of locating the anomaly, i.e. finding which is the particular OD-flow responsible for the abnormal links traffic variation. We consider the same simplifying hypothesis as in [72], considering only "localized" anomalies, namely anomalies in a single OD-flow at a time. Even if this assumption restricts the applicability of our algorithm, previous studies have shown that most OD-flow volume anomalies arise in single OD-flows [71]. Different from section 3.3, we now seek to detect and locate an additional anomalous volume θ in one single OD-flow k. This traduces into an additive change $\boldsymbol{\theta} = \mathbf{r}_k \theta$ in the SNMP measurements vector Y_t, where θ corresponds to the size of the anomaly and $\mathbf{r}_k$ is the k-th normalized column of R. The vector $\mathbf{r}_k$ defines the manner in which the anomaly adds traffic to each link in the network, and thus it can be seen as a signature of the anomaly.

Instead of maximizing the probability of anomaly detection for a bounded false alarm probability, we design an algorithm that minimizes the maximum mean detection/localization delay for an upper bounded probability of false localization and a lower bounded mean time between consecutive false alarms, a usual measure of the false alarm rate. The detection/localization delay is another crucial design criterion; indeed, the faster the detection and localization, the faster the resolution of the problem.

The problem of detecting and locating a volume anomaly that occurs at an unknown time t_0 is a particular case of a classical change detection/isolation problem, where the objective is to compute an alarm time T at which a change of type $\nu \in \{1, 2, \ldots, m\}$ in the probability distribution of a random sequence of measurements is detected. The alarm time T corresponds to the time when an anomaly in OD-flow ν is detected and located. Before going into the details of the particular algorithm, let us formally define the optimality minimax criterion that we use in the design. The optimality criterion consists of minimizing the maximum mean delay for detection/localization, given by:

$$\overline{\mathbb{E}}(T) = \sup_{t_0 \geq 1, 1 \leq k \leq m} \mathbb{E}^k_{t_0}(T - t_0 | T \geq t_0), \tag{3.4}$$

where $\mathbb{E}^k_{t_0}(T - t_0 | T \geq t_0)$ denotes the conditional expectation of $T - t_0$ when the event $\{T \geq t_0\}$ is true and the k-th change type occurs at time t_0, subject to the following constraints: (i) a lower bound for the mean time between two false alarms:

$$\mathbb{E}_0(T) \geq \upsilon \tag{3.5}$$

where υ is a prescribed lower bound and $\mathbb{E}_0(\cdot)$ denotes the expectation when all measurements have the same probability density function f_0, corresponding to anomaly-free traffic; (ii) an upper bound for the maximum probability of false localization:

$$\max_{1\leq k\leq m}\max_{1\leq j\neq k\leq m}\sup_{t_0\geq 1}\Pr^k_{t_0}(\nu=j|T\geq t_0)\leq\eta \tag{3.6}$$

where $\Pr^k_{t_0}(\nu = j|T \geq t_0)$ corresponds to the probability that the decision is j whereas the true change type is $k \neq j$. In brief, we require that the maximum mean detection/localization delay given by (3.4) should be *as small as possible*, subject to performance bounds on the mean time between consecutive false alarms and the maximum probability of false localization.

In order to construct an algorithm that verifies this minimax criterion, we shall treat the detection/localization of a volume anomaly that occurs at an unknown time t_0 as a sequential hypothesis testing problem, where the null hypothesis $\mathcal{H}_0 = \{$OD-flows are anomaly-free$\}$ ($t_0 = +\infty$) is tested against m alternatives $\mathcal{H}^k_{t_0} = \{$the k-th OD-flow presents an anomalous additional amount of traffic θ from time $t_0\}$, $k = 1, \ldots, m$. Sequential approaches are used to minimize the number of observations needed to decide among the hypotheses. The sequential hypothesis testing problem can be written as:

$$\begin{array}{rcl}\mathcal{H}_0 & : & Z_t\sim\mathcal{N}(H\,\boldsymbol{\mu}_t,\gamma_t^2\,I)\,,\quad t=1,2,..\\ \mathcal{H}^k_{t_0} & : & \left\{\begin{array}{l} Z_t\sim\mathcal{N}(H\,\boldsymbol{\mu}_t,\gamma_t^2\,I),\ t=1,..,t_0-1,..\\ Z_t\sim\mathcal{N}(H\,\boldsymbol{\mu}_t+\Phi^{-\frac{1}{2}}\,\mathbf{r}_k\,\theta,\gamma_t^2\,I),t=t_0,..\end{array}\right.\end{array} \tag{3.7}$$

where Z_t is the standardized measurements vector. As we did before, we can remove the nuisance parameter $\boldsymbol{\mu}$ from the detection problem. In order to only keep the anomalies-sensitive part of Z_t, we compute the residual process $U_t = WZ_t$, using a linear transformation W into a set of $r - q$ linearly independent variables. The matrix W^T is the linear rejector that eliminates the anomaly-free traffic by projection onto the left null space of H, built from the first $r - q$ eigenvectors of $P^{\perp}_H$ corresponding to eigenvalue 1. The rejector verifies the following relations: $WH = 0$, $W^TW = P^{\perp}_H$ and $WW^T = I_{r-q}$. Hypotheses $\mathcal{H}^k_{t_0}$ can be thus simplified by filtering the anomaly-free traffic:

$$\mathcal{H}^k_{t_0}:\left\{\begin{array}{l} U_t\sim\mathcal{N}(0,\gamma_t^2\,I)\,,\quad t=1,..,t_0-1,..\\ U_t\sim\mathcal{N}(\mathbf{u}_k\,\theta,\gamma_t^2\,I)\,,\quad t=t_0,t_0+1,..\end{array}\right.$$

where $\mathbf{u}_k = W\,\Phi^{-\frac{1}{2}}\,\mathbf{r}_k$ corresponds to the signature in the residuals of a change in OD flow k.

The recursive algorithm proposed in [89, 90] perfectly fits this detection/localization problem, with one useful feature, that of minimizing the mean number of samples needed to detect a change and decide among the different change types with bounded false alarm and false localization rates. This algorithm is asymptotically optimal, i.e. it asymptotically minimizes the maximum mean delay for detection/localization $\overline{\mathbb{E}}(T)$,

when both the false alarm and the false localization rates go to 0: $\max\{\upsilon^{-1}, \eta\} \to 0$. The output of the recursive detection/localization algorithm is twofold: (i) the alarm or stopping time T_r, which corresponds to the instant when an alarm is raised, and (ii) a decision ν_r, which corresponds to the type of change that the algorithm decides for among the m possible change types:

$$\begin{aligned} T_r &= \min_{1 \leqslant k \leqslant m} \{T_r(k)\}, \quad \nu_r = \arg \min_{1 \leqslant k \leqslant m} \{T_r(k)\} \\ T_r(k) &= \inf \{t \geqslant 1 : s_t(k) \geqslant 0\}, \quad k = 1 \dots m \\ s_t(k) &= \min_{0 \leqslant j \neq k \leqslant m} [g_t(k,j) - h_{k,j}], \quad k = 1 \dots m \end{aligned} \tag{3.8}$$

with $g_t(k,j) = g_t(k,0) - g_t(j,0)$. The recursive functions $g_t(k,0)$ are defined by

$$g_t(k,0) = (g_{t-1}(k,0) + u_t(k,0))^+ \tag{3.9}$$

$$u_t(k,0) = \log \frac{f_k(U_t)}{f_0(U_t)} \tag{3.10}$$

where $(x)^+ = \max(x, 0)$, $g_0(k,0) = 0$ for every $1 \leqslant k \leqslant m$ and $g_t(0,0) = 0$ for all t. The function f_0 represents the probability density function of residuals under anomaly-free behavior, and f_k is the probability density function of residuals $U_{t_0}, U_{t_0+1}, ..$ after a change of type k. The thresholds $h_{k,j}$ are chosen by the following formula:

$$h_{k,j} = \begin{cases} h_{\mathrm{d}} & \text{if } 1 \leqslant k \leqslant m \quad \text{and } j = 0 \\ h_{\mathrm{i}} & \text{if } 1 \leqslant k, j \leqslant m \quad \text{and } j \neq k \end{cases} \tag{3.11}$$

where h_{d} and h_{i} are the detection and localization thresholds. Basically, the anomaly detection is performed by comparing the m recursive functions $g_t(k,0)$ against the detection threshold h_{d}, while the anomaly localization is performed by comparing the difference between these m recursive functions with the localization threshold h_{i}. $T_r(k)$ is the first time when the alternative hypothesis $\mathcal{H}_{t_0}^k$ is chosen by the sequential test as the most likely hypothesis. The stopping time T_r corresponds to the earliest of all the times $T_r(k)$, with $1 \leqslant k \leqslant m$. The detected anomaly is declared in OD-flow k if the earliest of all these times was $T_r(k)$.

The choice of the detection and localization thresholds h_{d} and h_{i} is discussed in [89], with practical comments and simulation results about the effectiveness of such thresholds. In practice, the detection threshold h_{d} is fixed so as to achieve the desired false alarm rate. As it follows from [89], some statistical issues of the recursive algorithm can be solved by choosing $h_{\mathrm{d}} \geq h_{\mathrm{i}}$, and thus we will generally consider $h_{\mathrm{i}} = h_{\mathrm{d}}$. In other words, given the desired false alarm rate, we fix h_{d} and take the biggest value of h_{i} so as to minimize the false localization rate.

A final remark about the computation of the probability density functions in (3.10): f_0 is nothing but a Gaussian density function of law $\mathcal{N}(0, \gamma_t^2 I)$, where I is the identity matrix of dimensions $(r-q) \times (r-q)$ in this case. As regards f_k, the amplitude of the anomaly θ is completely unknown, and we must assume a certain distribution for it in order to correctly define f_k. Given that we are dealing with volume anomalies, it is reasonable to assume that the amplitude θ is uniformly distributed between two defined bounds θ_1 and θ_2. In this case, it is easy to see that f_k is simply a Gaussian mixture density. The bounds are introduced just for technical reasons and they can be chosen arbitrarily when dealing with volume anomalies. However, it is possible to control the sensitivity of the algorithm to detect small traffic changes instead of volume anomalies, see [90] for additional details. The choice of the bounds has little impact as regards anomaly localization, because the signature is based on the direction of the anomaly and not on its amplitude.

The optimal sequential volume anomaly detection and localization algorithm presented in this section will be referred as the Sequential Spline-Based (SSB) method in the rest of the thesis.

3.5 Benchmarking Methods for Anomaly Detection and Localization

In this section we describe two well-known anomaly detection and localization methods proposed in the literature that will be used as benchmark for the OSBD and the SBB methods. The former of these methods is based on the PCA technique described in section 2.6, the latter uses the recursive Kalman filtering TME method previously described in 2.4 to detect volume anomalies as large estimation errors.

3.5.1 Anomaly Detection and Localization with PCA and the Sub-Space Method

The well-known PCA approach for anomaly detection and localization is chosen as benchmark due to its relevance in the anomaly detection literature [72, 37, 71, 74, 76]. This approach comes directly from the theory developed for subspace-based fault detection in multivariate process control [68, 69, 70]. The PCA approach and the subspace method consist of a decomposition of the SNMP measurements into a Principal Components (PCs) basis, separating traffic into a "normal subspace" that captures the anomaly-free traffic behavior, and an "anomalous subspace" that provides residuals sensitive to anomalies.

Similar to the PCA decomposition presented in section 2.6, the method considers a SNMP measurements matrix $\mathbf{Y} \in \mathbb{R}^{p \times r}$ of p consecutive SNMP measurements vectors, $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_p\}^T$, where each column of $\mathbf{Y}$ represents a time series of p samples of consecutive SNMP measurements for each link $l = 1, \ldots, r$. Using PCA, the r PCs of $\mathbf{Y}$ are computed and separated into two disjoint sets: the set of "normal components", composed of the first k PCs, and the set of "anomalous components", composed of the remaining $r - k$ PCs. The idea behind this approach is that traffic anomalies are sparse in $\mathbf{Y}$, and so the first components of the PCA transformation will correctly capture the anomaly-free traffic behavior, while anomalies will be visible in the remaining components.

The space spanned by the set of normal components is the "normal subspace" $\mathcal{S}$ and the space spanned by the anomalous components is the "anomalous subspace" $\hat{\mathcal{S}}$. Given $\mathcal{S}$ and $\hat{\mathcal{S}}$, every SNMP measurements vector $Y \in \mathbf{Y}$ can be separated into the modeled traffic Y_{model} and the residual traffic Y_{residual} by simple projection onto $\mathcal{S}$ and $\hat{\mathcal{S}}$ respectively:

$$\begin{aligned} Y &= Y_{\text{model}} + Y_{\text{residual}}, \qquad Y \in \mathbf{Y} \\ Y_{\text{model}} &= \mathbf{P}\mathbf{P}^T Y \\ Y_{\text{residual}} &= \left(I - \mathbf{P}\mathbf{P}^T\right) Y \end{aligned} \tag{3.12}$$

where $\mathbf{P} \in \mathbb{R}^{r \times k}$ stands for the matrix with the first k PCs as column vectors and $\mathbf{P}\mathbf{P}^T$ represents the projection matrix onto the normal subspace. The anomaly detection is performed in the residual traffic Y_{residual}, looking for changes in the squared norm of residuals that exceed a certain threshold δ_α, defined according to the desired false alarm rate α:

$$||Y_{\text{residual}}||^2 \leqslant \delta_\alpha \tag{3.13}$$

As regards the use of PCA and the subspace method for anomaly localization [72], authors consider only the set of anomalies that arise due to unusual traffic in a single OD-flow, and thus the anomaly localization consist in finding the anomalous OD-flow among the m OD-flows of the TM. Let us identify the set of hypothesized anomalies as $\{A_k,\, k = 1, \ldots, m\}$, where A_k corresponds to an anomaly in OD-flow k. When an anomaly A_k occurs, the SNMP measurements vector is represented by:

$$Y = Y^* + \mathbf{r}_k\, \theta \tag{3.14}$$

where Y^* represents the SNMP measurements vector for normal-operation traffic conditions and which is unknown when the anomaly occurs. As before, θ is the size of the anomaly and $\mathbf{r}_k$ defines the manner in which this anomaly adds traffic to each link in the network.

The approach to identify the anomalous OD-flow proposed in [72] consists in first estimating the anomaly-free traffic volume for each hypothesized anomaly A_k, which will be referenced as Y_k^*, and then selecting the hypothesized anomaly that minimizes the projection of Y_k^* onto the anomalous sub-space $\hat{\mathcal{S}}$. Briefly speaking, the approach chooses the hypothesis that explains the largest amount of residual traffic.

Authors in [72] propose to estimate Y_k^* by eliminating the effect of the anomaly from Y, subtracting the anomalous traffic contribution from the links associated with anomaly A_k. For doing so, they first estimate the size of the anomaly, referred as $\hat{\theta}$:

$$\hat{\theta} = \arg\min_{\theta} ||Y_{\text{residual}} - \mathbf{Q}\, \mathbf{r}_k\, \theta||^2 \tag{3.15}$$

where $\mathbf{Q} = \left(I - \mathbf{P}\mathbf{P}^T\right)$ is the projection matrix onto the anomalous sub-space. The estimation $\hat{\theta}$ corresponds to the volume that minimizes the distance between the residual traffic Y_{residual} and the projection of the additional anomalous traffic onto $\hat{\mathcal{S}}$. The result of (3.15) is a simple least-squares estimate, given by:

$$\hat{\theta} = \left(\mathbf{r}_k^T\, \mathbf{Q}^T \mathbf{Q}\, \mathbf{r}_k\right)^{-1} \mathbf{r}_k^T\, \mathbf{Q}^T\, Y_{\text{residual}} \tag{3.16}$$

This leads to the best estimate of Y^* assuming anomaly A_k:

$$\begin{aligned} Y_k^* &= Y - \mathbf{r}_k\,\hat{\theta} \\ &= Y - \mathbf{r}_k\,\left(\mathbf{r}_k^T\,\mathbf{Q}^T\mathbf{Q}\,\mathbf{r}_k\right)^{-1}\mathbf{r}_k^T\,\mathbf{Q}^T\,Y_{\text{residual}} \\ &= \left(I - \mathbf{r}_k\,\left(\mathbf{r}_k^T\,\mathbf{Q}^T\mathbf{Q}\,\mathbf{r}_k\right)^{-1}\mathbf{r}_k^T\,\mathbf{Q}^T\,\mathbf{Q}\right)\,Y \end{aligned}$$

After computing the m possible values for Y_k^*, the method choses the anomaly A_k that minimizes the norm of the projection of Y_k^* onto $\hat{\mathcal{S}}$:

$$k = \arg\min_{i=1,\ldots,m} ||\mathbf{Q}\,Y_i^*||^2 \tag{3.17}$$

The reader should note that this anomaly localization approach is nothing but a quite round-about heuristic without a clear justification of why it should work properly. The heuristic has also some inherent problems, for example that of dragging the estimation errors introduced in (3.16). In addition, previous works [76] claim that the heuristic can unduly trigger alarms in some OD-flows much more frequently than others, which also questions its usefulness.

3.5.2 Anomaly Detection with Kalman Filters

The method proposed in [78] for anomaly detection uses the recursive Kalman filtering technique presented in section 2.4, but instead of computing an estimated TM, it uses the Kalman filter to detect volume anomalies as large estimation errors. Since the Kalman filter method can properly track the evolution of the TM in the absence of anomalies, estimation errors should be small most of the time. A volume anomaly is then flagged when the estimation error exceeds certain detection threshold.

The method analyses the estimation error $\varepsilon_t = \mathrm{e}_{t|t} = X_t - \hat{X}_{t|t}$, where $\hat{X}_{t|t}$ represents the estimated TM using all the SNMP measurements until time t, namely $\{Y_t, Y_{t-1}, Y_{t-2}, \ldots\}$. The problem with this approach is that the real value of the TM, namely X_t, can not be directly measured and thus ε_t can not be computed. Fortunately, the Kalman filter equations (2.15) and (2.16) permit to compute the estimation error indirectly, using the innovation process η_{t+1}. The innovation process represents the difference between the observed SNMP measurements vector Y_{t+1} and its predicted value, obtained from the predicted TM $\hat{X}_{t+1|t}$:

$$\eta_{t+1} = Y_{t+1} - R\,\hat{X}_{t+1|t} \tag{3.18}$$

Under the Kalman filtering hypotheses, the innovation process is a zero-mean Gaussian process, whose covariance matrix $\Gamma_{t+1} = \mathbb{E}(\eta_{t+1}\,\eta_{t+1}^T)$ can be easily derived from equations (2.14), (2.15), and (2.16):

$$\Gamma_{t+1} = R\,P_{t+1|t}\,R^T + Q_{v_{t+1}} \tag{3.19}$$

where $P_{t+1|t}$ is the covariance matrix of the prediction error $X_{t+1} - \hat{X}_{t+1|t}$ and $Q_{v_{t+1}}$ is the covariance matrix of the observation noise process V_{t+1}. In [78], authors use a least squares estimate to infer the estimation error ε_{t+1} from the innovation process η_{t+1}:

$$\varepsilon_{t+1} \approx -K_{t+1}\, R\, P_{t+1|t}\, S_{t+1}^{-1}\, K_{t+1}\, \eta_{t+1} \tag{3.20}$$

where $S_{t+1} = K_{t+1}\, \Gamma_{t+1}\, K_{t+1}^T$ is the covariance matrix of the residual TM, namely $\hat{X}_{t+1|t+1} - \hat{X}_{t+1|t}$. Using (3.21), authors in [78] propose a statistical test following the Neyman-Pearson theorem to detect volume anomalies. The approach basically consists of constructing the following test:

$$\frac{\left|\left|K_{t+1}\, R\, P_{t+1|t}\, S_{t+1}^{-1}\, K_{t+1}\, \eta_{t+1}\right|\right|^2}{\left|\left|P_{t+1|t+1}\right|\right|} > T_d \tag{3.21}$$

where $\|\cdot\|$ represents the Euclidean norm. Briefly speaking, the test compares the estimation error obtained from the SNMP measurements to that obtained from the recursive filter, calibrated for anomaly-free traffic behavior. When the ratio between both quantities exceeds the detection threshold T_d, the algorithm raises an anomaly alarm.

3.6 Performance Evaluation

In this section we present the evaluation of the anomaly detection and localization algorithms presented in sections 3.3 and 3.4, comparing their performance to that obtained with the two benchmarking methods in different operational networks. The datasets correspond to those previously described in section 2.3.1, including the Abilene network, the GEANT network, and a Tier-2 ISP network.

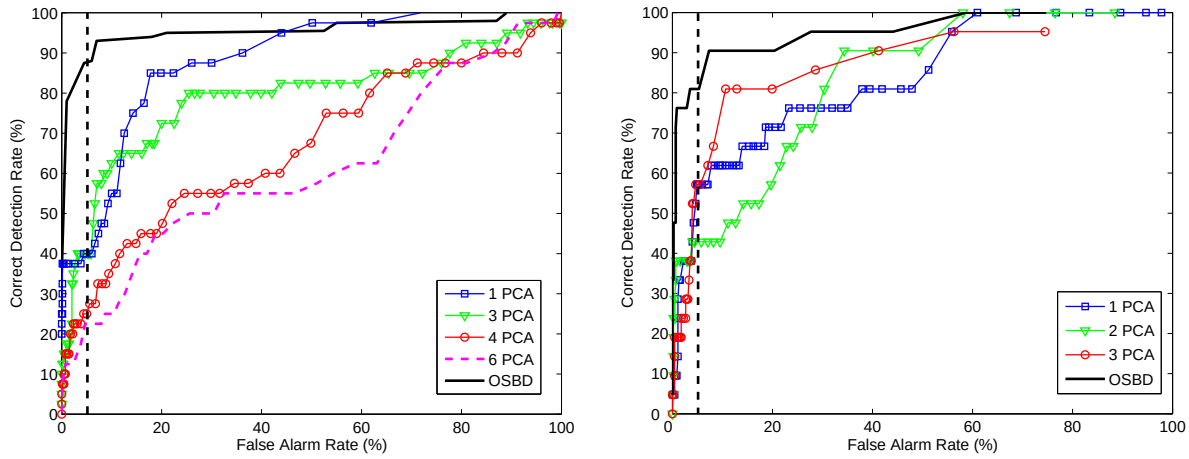
We first compare the anomaly detection performance of the OSBD method against the PCA approach in Abilene and GEANT. Then we evaluate the SSB anomaly detection/localization method in Abilene and in the Tier-2 network. Finally, we compare the performance of the SSB method against the PCA approach and the Kalman-based anomaly detection method, using artificial volume anomalies introduced in Abilene. In all cases, we show that the performance of our algorithms obtained in the practice is in agreement with the thorough theoretical foundation.

3.6.1 Numerical Evaluation of the OSBD Method

In this evaluation we shall use two validation datasets, the former is composed of 720 consecutive SNMP measurements from Abilene, the latter corresponds to 720 measurements from GEANT. The learning datasets for the OSBD method consist of 1 h of anomaly-free SNMP measurements, gathered immediately before the validation datasets. As regards the PCA approach, the method is not designed to work on-line as presented in [72] and in section 3.5.1. Instead, the analysis is performed off-line over the complete validation dataset $\mathbf{Y} = \{Y_{t_1}, Y_{t_2}, \ldots, Y_{t_{720}}\}^T$. Thus, the learning and validation datasets are the same for the PCA approach.

For the sake of false alarm and correct detection rates evaluation, the set of "true" anomalies is manually identified in each testing dataset. Manual inspection declares an anomaly in an OD-flow if the unusual deviation intensity of the guilty OD-flow leads to an increase of traffic (i) larger than 1.5% of the total amount of traffic on the network and (ii) larger than 1% of the amount of traffic carried by the links routing this guilty OD-flow, for each of these links. This rule is based on the conclusions about large traffic changes drawn in [80]. Hence, only large volume anomalies are considered as "true anomalies". 40 measurements of the Abilene testing dataset are affected by at least one significant volume anomaly. In the case of the GEANT testing dataset, 36 anomalous measurements are identified.

Different from the PCA approach, the OSBD method is applied to the SNMP measurements of each validation dataset in an on-line fashion, sequentially running the test defined in (3.3) for every new "incoming" SNMP measurement $Y_{t_1}, Y_{t_2}, \ldots, Y_{t_{720}}$. For the detection purpose, it is crucially important to have a good estimate of γ_t. This parameter is easily estimated from the learning dataset by using the Maximum Likelihood estimate of noise variance in residuals U_t [82]. Since this



(a) ROC Curves for Abilene

(b) ROC Curves for GEANT

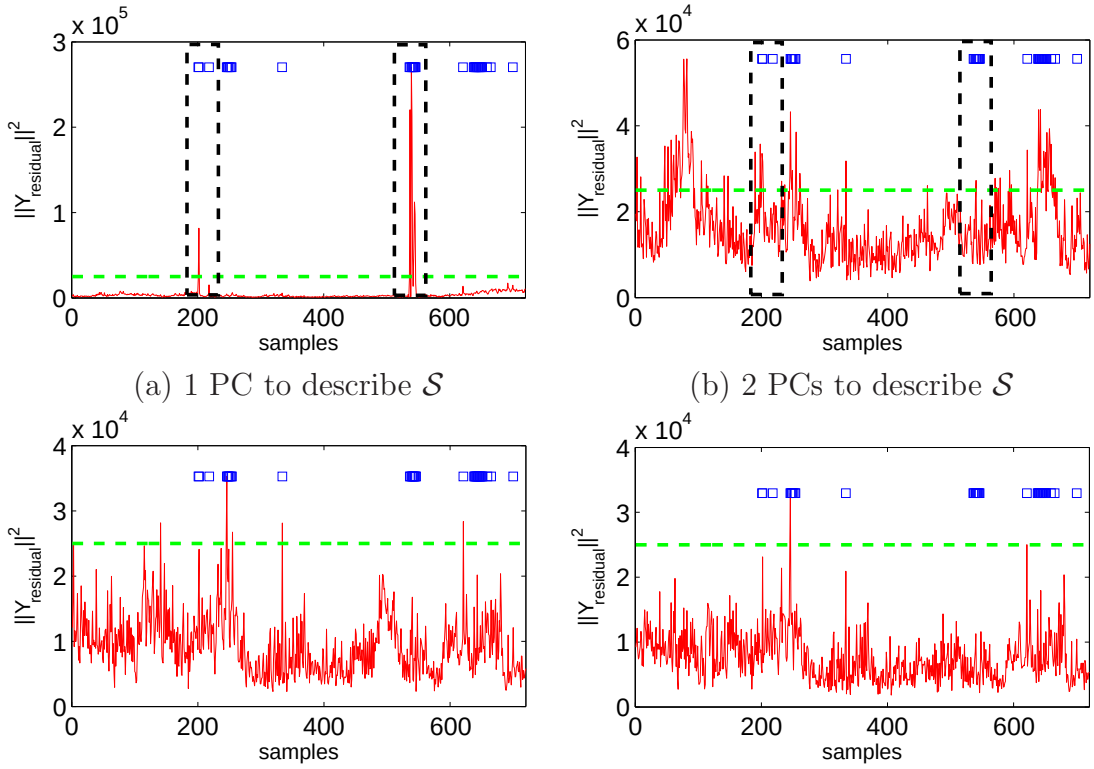
Figure 3.2 — Correct detection rate vs false alarm rate for the Optimal Spline-Based Detection method (OSBD - solid line) and the PCA approach, considering a different number of k first PCs $\mathbf{v}_k$ to model the normal sub-space.

parameter can slowly vary in time, its value is updated during the test: at time t, if no anomaly has been declared in the last hour, γ_t is estimated by its value one hour before.

Figure 3.2 depicts the ROC curves for the OSBD and the PCA methods in the Abilene and the GEANT datasets, showing the correct detection rate β for different values of the false alarm rate α, corresponding to different values of detection threshold. The ROC curves allow to compare the accuracy of both approaches and the sensitivity of each detection method w.r.t. the variation of the detection thresholds, showing the existing trade-off between the correct detection and the false alarm rates.

In the PCA approach, a different number of k first PCs $\mathbf{v}_k$ is used to model the normal sub-space. Results obtained with the PCA approach in the Abilene dataset are quite far from those obtained with the OSBD method; the PCA test presents more than 2 times lower detection rates for a reasonable false alarm rate, below 5%. For example, for a false alarm rate $\alpha = 1\%$, the OSBD method correctly detects almost 80% of the anomalies, while this value drops to nearly 40% for the best performance of the PCA approach, using 1 PC to model the normal sub-space. Results are quite similar for the GEANT dataset, but in this case the best performance of the PCA approach is attained using 3 PCs to model the normal sub-space.

Figure 3.2 also evidences the lack of consistency of the PCA approach as regards the number of PCs used to model the anomaly-free traffic; for the same dataset, results are quite different when this number slightly varies. For the different datasets, the number of PCs that provides better results also differs, which makes it difficult to generalize results. As it is shown in recent works [76], the PCA approach has to be highly tuned for each particular dataset in order to provide reliable results, making



(a) 1 PC to describe $\mathcal{S}$

(b) 2 PCs to describe $\mathcal{S}$

(c) 4 PCs to describe $\mathcal{S}$

(c) 5 PCs to describe $\mathcal{S}$

Figure 3.3 — Temporal evolution of $||Y_{\text{residual}}||^2$, using a different number of first PCs to model $\mathcal{S}$. The squares indicate when an anomaly truly occurs. The dotted line depicts the detection threshold. Large anomalies pollute the normal sub-space and are not detected with the PCA approach. (a) Both large volume anomalies at samples 200 and 540 are correctly detected using 1 PC to describe $\mathcal{S}$. (b) Large volume anomalies are not detected using a 2 PCs representation of $\mathcal{S}$.

it inapplicable in a general real scenario. In fact, the main problem with current in-house methods is the difficulty to generalize their results.

The last important observation is that the OSBD method provides highly accurate results with a remarkably short learning-step, reinforcing the stability properties of the underlying parametric anomaly-free-traffic model and the robustness of the approach. On the contrary, the PCA approach provides a completely data-driven model for anomaly-free traffic, resulting in the aforementioned shortcomings.

3.6.2 Limitations of PCA for Anomaly Detection

Let us try to explain why the PCA approach may provide such a bad detection performance. There are at least three major problems regarding the PCA approach: (i) its performance strongly depends on the number of first PCs used to describe the normal subspace; (ii) the approach is data-driven, and (iii) the application of the PCA analysis directly to possibly "contaminated" data (i.e., data with volume anomalies)

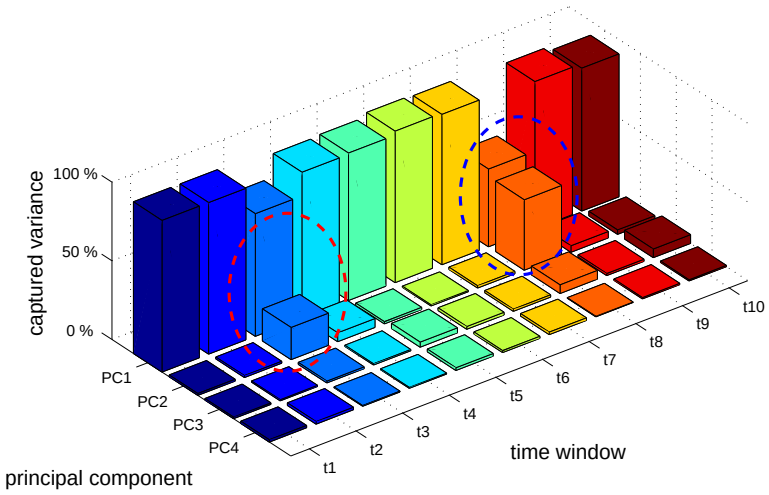


Figure 3.4 — Temporal evolution of the total variance captured by each PC $\mathbf{v}_i$, $||\mathbf{Y}\mathbf{v}_i||^2$. Each time window $t_{j=1..10}$ consists of 6hs of SNMP data. Large volume anomalies may inadvertently pollute the normal subspace at t_3 and t_8.

may pollute the normal subspace. These problems were also analyzed and verified in [75, 76], we shall exemplify them in the Abilene dataset.

Let us begin by the first issue; in [72], the separation between the normal and the anomalous PCs is performed using a simple ad-hoc threshold-based separation method that is highly tuned for each dataset and cannot therefore be generalized, turning the PCA approach inapplicable in a general scenario. Figure 3.3 depicts the temporal evolution of $||Y_{\text{residual}}||^2$, using a different number of first PCs to describe the normal subspace, from 1 to 5.

The dotted line represents the detection threshold; the squares indicate the times when an anomaly truly occurs, according to the manual inspection. It can be appreciated that the false positive rate is very sensitive to small differences in the number of PCs used to describe the normal subspace. The ROC curves in figure 3.2 show that there is no single PCA representation for the Abilene dataset that offers a good balance between correct detection and false alarm rates.

Regarding the second and third issues, the PCA approach is data-driven, which means that the PCA decomposition strongly depends on the considered SNMP measurements matrix $\mathbf{Y}$. This is a serious problem, simply because the characteristics of each measurements matrix $\mathbf{Y}$ vary from one to the other, and a deep analysis of the data must be conducted to avoid inconsistencies.

The PCA approach assumes that traffic anomalies are sparse in $\mathbf{Y}$, and thus the normal subspace can be correctly described by the first PCs that capture the highest level of "energy". Figure 3.4 depicts the temporal evolution of the variance captured by each PC $\mathbf{v}_i$, $||\mathbf{Y}\mathbf{v}_i||^2$, considering consecutive matrices $\mathbf{Y}$ spanning 6hs of traffic each. For each of these matrices we compute the set of PCs, which means that every 6hs the PCs are recomputed.

In almost every time window, the first PC captures the highest energy, justifying the use of 1 single PC to describe the normal subspace. However, large volume anomalies at time windows t_3 and t_8, also visible in figure 3.3(a), contribute to a large proportion of the captured energy; in this case, a second PC may be added as a descriptor of the normal subspace. Since this second component corresponds in fact to an anomaly, the normal subspace is inadvertently polluted. In figure 3.3(b), both large anomalies at t_3 and t_8 are not detected due to this effect.

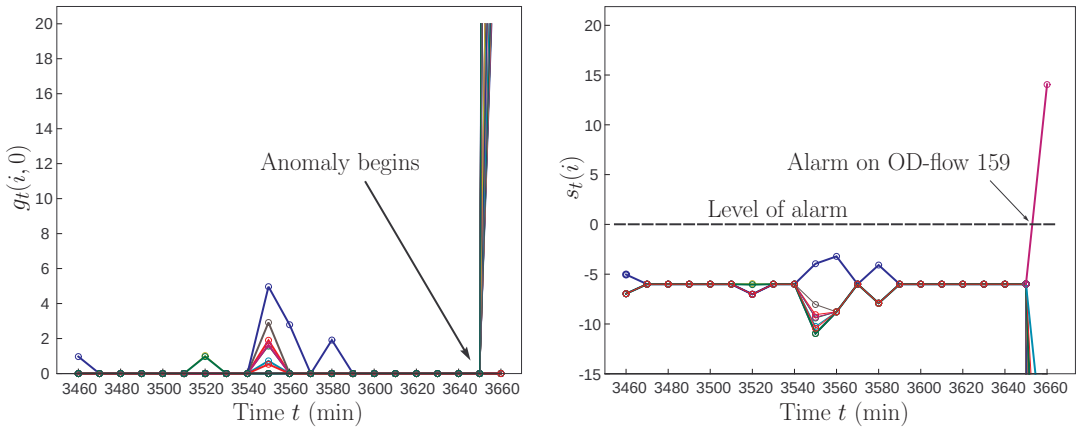
Our algorithms rely on a model which is not data-driven and has a very short learning-step. The effect of a training step over polluted data does not represent a problem to our short-learning approach, as it is quite simple to assure a 1 h anomaly-free time period.

3.6.3 Evaluation of the SSB Detection/Localization Method

Le us first demonstrate the ability of the SSB method to detect and locate an OD-flow volume anomaly from SNMP measurements in two different networks, the commercial Tier-2 ISP network and the Abilene network. Figures 3.5 and 3.6 show a typical realization of functions $s_t(i)$ and $g_t(i,0)$ defined in (3.8) and (3.9) respectively, both for a Tier-2 network and the Abilene network. Functions $s_t(i)$ are used to "monitor" the OD-flows; when $s_t(i)$ exceeds the threshold 0, OD-flow i is declared anomalous.

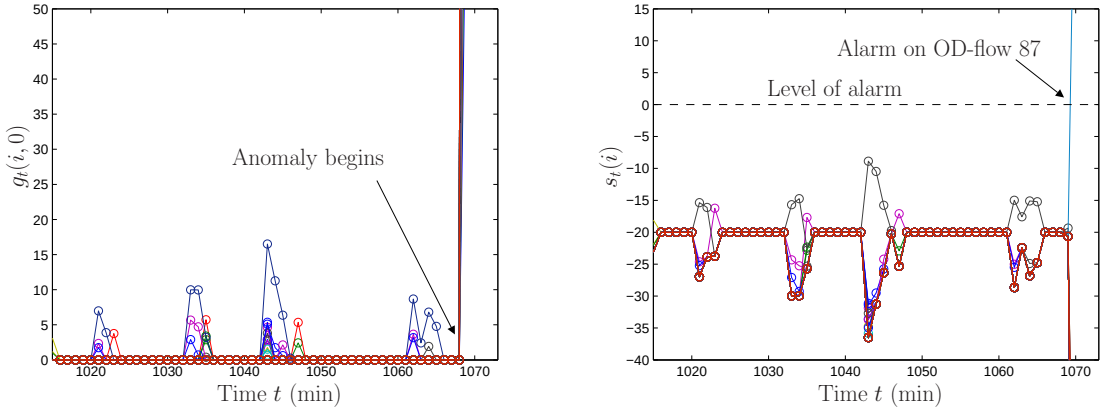
The anomaly in the Tier-2 network begins at time 3660 min, and at time 1070 min in Abilene. Note that after this time, several recursive functions $g_t(i,0)$ rapidly grow in both network scenarios. Each function $g_t(i,0)$ is associated with OD-flow i and when this function increases, it means that OD-flow i is suspected of carrying an abnormal amount of traffic. Contrary to $g_t(i,0)$, only function $s_t(159)$ associated to anomalous OD-flow 159 increases and finally exceeds the threshold 0 in the Tier-2 network. In the case of Abilene, the anomaly is correctly located in OD-flow 87.

Functions $s_t(i)$ permit to locate the anomalous OD-flow among all the OD-flows associated to functions $g_t(i,0)$ that have rapidly increased. The volume anomalies detected in these examples correspond to abrupt and massive volume augmentations, and thus functions $s_t(i)$ only need 1 observation to detect and locate the anomalous OD-flow. Since the underlying sampling rates of both datasets are 10' and 5' for the Tier-2 and the Abilene networks respectively, the detection lag corresponds to a delay of 10' and 5' in each case. Note however that our algorithm in not intrinsically



(a) Recursive functions $g_t(i, 0)$ (b) Functions $s_t(i)$

Figure 3.5 — Typical realizations of anomaly detection/localization functions in a Tier-2 ISP network.

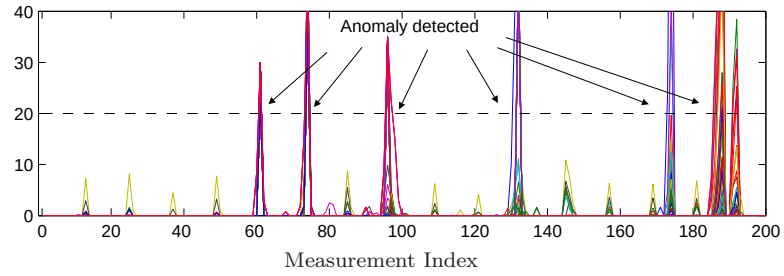


(c) Recursive functions $g_t(i, 0)$ (d) Functions $s_t(i)$

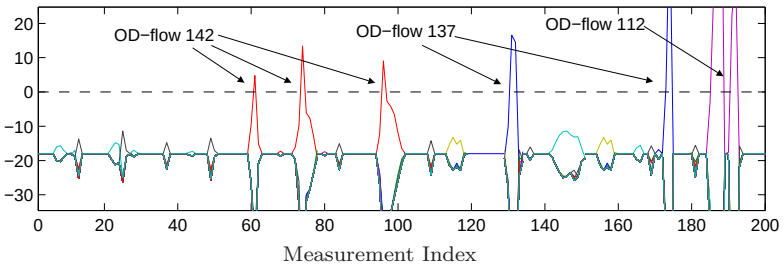
Figure 3.6 — Typical realizations of anomaly detection/localization functions in the Abilene network.

tied to any particular sampling rate, thus this detection delay would be even shorter if the sampling rates were higher. An interesting observation of this evaluation is that the SSB method achieves accurate results in both datasets, even though the respective anomaly-free traffic behaviors are quite different between these two networks.

Let us now compare the performance of the SSB method to continuously detect and locate volume anomalies in real-time against the two benchmark methods described in section 3.5, the Kalman-Based (KB) method and the PCA method. In order to use the PCA method to detect and locate volume anomalies in real-time, we shall consider a sequential implementation of the PCA approach, which we will referenced as the Sequential PCA method (SPCA). This sequential extension of the PCA approach comes from the authors of the former PCA method in [37, 72], but the method was never evaluated in their anomaly detection work [72]. The idea is straightforward; the



(a) Recursive functions $g_t(i, 0)$



(b) Functions $s_t(i)$

Figure 3.7 — On-line volume anomaly detection and localization in Abilene, using the Sequential Spline-Based method.

PCs and the corresponding projection matrix $\mathbf{PP}^T$ are built off-line from a certain time window $[t_1, t_n]$ of SNMP measurements $\mathbf{Y} = \{Y_{t_1}, Y_{t_2}, \ldots, Y_{t_n}\}^T$. Subsequently, every new arriving measurement Y_t at time $t > t_n$ is processed on-line using this projection matrix.

The SPCA method is used to both detect and locate volume anomalies, using the same algorithms presented in 3.5.1. The KB method is only used for volume anomaly detection as presented in [78], thus we do not intend to use it for anomaly localization. In order to use the SSB method to continuously detect and locate volume anomalies, the algorithm statistics are reset to 0 after each anomaly detection, i.e., $g_t(i, 0)$ is set to 0 after a change detection at time t, $\forall i = 1, \ldots, m$. Figure 3.7 shows how the SSB method works on-line, continuously detecting and locating volume anomalies in Abilene.

The testing dataset used for the evaluation consists of 864 consecutive SNMP measurements from the Abilene network. Instead of manually identifying the set of true volume anomalies, we introduce synthetic volume anomalies into this set. Indeed, in order to test the volume anomaly localization algorithms, we need to know exactly which is the anomalous OD-flow. Additionally, we need to be sure that a volume anomaly only occurs at a particular OD-flow at one time, so as to fulfill the simplifying hypothesis of single OD-flow anomalies.

Method	Detected	False Alarms	Located
SSB	93.9 %	1.4 %	90.8 %
KB	90.8 %	1.3 %	n/a
SPCA (1 PC)	76.9 %	1.9 %	73.9 %
SPCA (3 PCs)	53.9 %	1.7 %	49.2 %

Table 3.2 — Results of the detection and localization for 864 SNMP measurements in Abilene, composed of 65 OD-flow volume anomalies.

We follow a similar procedure as that described in [78] to introduce 63 large synthetic volume anomalies. The basic idea of this procedure consists in extracting the long-term trend from each OD-flow, adding a Gaussian noise to these "smoothed" OD-flows and finally adding the synthetic volume anomalies to this "anomaly-free" smoothed dataset. These anomalies correspond to short-lived volume changes in particular single OD-flows. We additionally add 2 short-lived volume anomalies that span multiple OD-flows at the same time, in order to analyze the response of the single OD-flow volume anomaly localization algorithms in that case.

Table 3.2 presents the comparative performance of the three algorithms. As before, the learning dataset for the SSB method consists of 1 h of anomaly-free SNMP measurements. As in [37], the training dataset for the SPCA method consists of 1 week of SNMP measurements, gathered immediately before the validation dataset and not necessarily free of volume anomalies. Similarly to [78], the learning dataset for the KB method consists of 24hs of anomaly-free direct OD-flow measurements.

The detection thresholds for the three methods are set so as to achieve a false alarm rate of about 1% in the validation dataset. As we have previously stated in section 3.4 and considering the observations in [89], the localization threshold of the SSB method is set to the same value as the detection threshold, i.e., $h_{\mathrm{i}} = h_{\mathrm{d}}$ in equation (3.11). In order to appreciate the sensitivity of the SPCA method to the dimensionality of the normal subspace, we consider two different representations for $\mathcal{S}$, using 1 and 3 PC(s) respectively.

The SSB method correctly detects 61 out of the 65 volume anomalies, producing a total of 12 false alarms on the 864 measurements of the validation dataset. From the 61 detected anomalies, 59 are correctly located in the particular anomalous OD-flows. The 2 volume anomalies that are not correctly located correspond to those anomalies that span multiple OD-flows simultaneously. In this case the algorithm certainly produces an alarm, but the localization step can not correctly distinguish between the anomalous OD-flows. In the following section we discuss an approach to solve this problem.

Detection results are similar for the KB method, which correctly detects 59 anomalies with only 11 false alarms. Obtained results are less accurate with the SPCA approach and many anomalies go undetected. Using 1 PC to construct the normal subspace, the SPCA method correctly detects 50 volume anomalies while triggering 16 false alarms. The detection threshold of the SPCA approach can be tuned so as to correctly detect 89% of the anomalies, but the false alarm rate climbs to approximately 6% in that case, a value almost 5 times bigger than the rest of the methods.

The SPCA method has a similar problem to locate multiple OD-flows anomalies. Studies presented in [76] show that correctly identifying the anomalous OD-flows with the PCA approach is inherently difficult. Results are quite poor when using 3 PCs to model $\mathcal{S}$, only detecting 35 volume anomalies and locating 32. These results are consistent with the sensitivity analysis and the highlighted shortcomings of the PCA approach presented in [76, 75].

3.7 Discussion

In this section we shall focus on complexity and implementation issues of the presented methods, discussing advantages and disadvantages of our proposals with respect to previous works, as well as some possible extensions for the anomaly localization algorithm.

3.7.1 Complexity Analysis

Numerical complexity and memory storage are central issues for on-line anomaly detection. Most of previous works on network-wide anomaly detection have conceived methods for off-line detection [72, 80], mining anomalies in large snapshots of data rather than treating every single measurement sequentially. These methods can be used for diagnosis of volume anomalies after their occurrence, but are rather useless for an ISP if anomaly mitigation or any other kind of countermeasure is the objective. On the contrary, both Spline-Based methods can be used for on-line anomaly detection, and thus we should assess their complexity. Let us compare the numerical complexity of these algorithms against those used for comparison in section 3.6.3, the Kalman-Based method and the Sequential version of the PCA approach.

The OSBD method stores two matrices in memory, the matrix $\Phi^{-\frac{1}{2}}$, with $\Phi = R\Sigma R^T$, and the projection matrix $P_H^{\perp} = I - H(H^TH)^{-1}H^T$, with $H = \Phi^{-\frac{1}{2}}RS$. This represents a total of $3r^2/2$ variables ($P_H^{\perp}$ is symmetric), where r is the number of links in the network. The computation of $\Phi^{-\frac{1}{2}}$ and $P_H^{\perp}$ involves matrix multiplications and inversions, and thus the associated cost is $\mathcal{O}(r^3)$. There is an additional cost in the learning phase of the SB methods, related to the Tomo-Gravity estimate used to construct the splines basis S. The cost of the TGE method in the learning phase is similar to that of a least-squares method, which implies $\mathcal{O}(rm^2)$ operations to estimate a TM.

All these matrices are computed off-line during the learning phase and do not affect the scalability and on-line applicability of the method. The on-line application involves three consecutive operations at every time t: the "whitening" of the SNMP measurements vector $Z_t = \Phi^{-\frac{1}{2}}Y_t$, the projection of the obtained vector onto the left null space of H, and the computation of the norm of this projection. All these operations have a complexity $\mathcal{O}(r^2)$.

Memory usage is similar in the case of the SSB method. The matrix $\Phi^{-\frac{1}{2}}$ is also stored, but instead of saving the projection matrix $P_H^{\perp}$ the rejector W is kept in memory, built from the first $r-q$ eigenvectors of $P_H^{\perp}$. Given the recursive structure of the SSB method, m additional variables are kept in memory, which corresponds to the m recursive functions $g_t(i,0)$, $i = 1,\ldots,m$. For anomaly localization purposes, the m anomaly signatures $\mathbf{u}_k \in \mathbb{R}^{(r-q)\times 1}$ are also stored.

The singular value decomposition (SVD) of $P_H^{\perp}$ has a computation complexity of $\mathcal{O}(r^3)$, and as before, the construction of the splines basis involves $\mathcal{O}(rm^2)$ operations. In the on-line detection/localization phase, residuals $U_t = WZ_t$ are firstly computed and then used to update the m recursive functions $g_t(i, 0)$ according to (3.9) and (3.10). Finally, the m functions $s_t(i)$ used for anomaly localization are computed according to (3.8). These steps involve approximately $\mathcal{O}(r^2)$ operations for anomaly detection and $\mathcal{O}(m^2)$ additional operations for anomaly localization.

The SPCA method keeps the symmetric projection matrix $\mathbf{Q} = (I - \mathbf{P}\mathbf{P}^T)$ in memory, which accounts for $r^2/2$ variables. The anomaly localization in the SPCA method consists of a greedy search for a particular anomaly signature, each represented by a normalized column of the routing matrix $\mathbf{r}_k \in \mathbb{R}^{r\times 1}$ that must also be saved in memory. The construction of $\mathbf{P}\mathbf{P}^T$ relies on computing the SVD of the SNMP measurements matrix $\mathbf{Y} \in \mathbb{R}^{p\times r}$, where p is the number of consecutive SNMP measurements considered, a number usually much bigger than r; for example, $p = 1008$ and $r = 49$ in [72], corresponding to 1 week of traffic. This SVD has a numerical complexity in $\mathcal{O}(pr^2)$.

The use of the SPCA for anomaly detection involves the projection of the SNMP measurements vector onto the anomaly subspace and the computation of the norm of this projection, with a numerical complexity in $\mathcal{O}(r^2)$. As regards anomaly localization, the greedy search consists of constructing m possible anomaly explanations, with an approximated cost of $\mathcal{O}(r^2)$ operations each, thus additionally adding $\mathcal{O}(mr^2)$ operations.

Finally, the KB method complexity corresponds to that of the standard Kalman filter recursive equations. The method must store in memory an $m \times m$ state transition diagonal matrix that models the evolution of the anomaly-free traffic matrix, the routing matrix R, and the noise covariance matrices associated with the observation and the evolution processes; this last is also a diagonal matrix. This accounts for a total of $2(r^2 + m)$ variables in memory. The recursive nature of the Kalman filter implies to keep in memory two additional matrices, the $m \times r$ Kalman gain matrix and the $m \times m$ prediction error covariance matrix.

The learning process of the KB method as proposed in [78] consists of a recursive Expectation Maximization (EM) approach. There are many different EM algorithms, but in all cases the resolution involves matrix operations with a numerical complexity of $\mathcal{O}(m^3)$ for the estimation of an $m \times 1$ vector. The use of the KB for on-line anomaly detection implies to update the Kalman gain, the estimation covariance error and the residual error. This involves matrix multiplications and inversions, and thus the associated cost is $\mathcal{O}(m^3)$.

Table 3.3 builds a raw summary of the numerical complexity and memory storage restrictions for the algorithms discussed above. Memory usage is similar in all cases, with a slightly higher requirement for the KB approach. While the SPCA method

Method	nº vars. mem.	nº ops. learn	nº ops. on-line	
OSBD	$\mathcal{O}(r^2)$	$\mathcal{O}(rm^2)$	$\mathcal{O}(r^2)$	n/a
SSB	$\mathcal{O}(mr)$	$\mathcal{O}(rm^2)$	$\mathcal{O}(r^2)$	$\mathcal{O}(m^2)$
SPCA	$\mathcal{O}(mr)$	$\mathcal{O}(pr^2)$	$\mathcal{O}(r^2)$	$\mathcal{O}(mr^2)$
KB	$\mathcal{O}(m^2)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$	n/a

Table 3.3 — Numerical complexity and memory usage for different on-line anomaly detection algorithms. On-line operations are divided into detection operations and localization operations.

works with an $p \times r$ matrix in the learning phase, the SB and the KB methods use $m \times m$ matrices and thus they may require more operations for learning issues, depending on the relation between m and p.

As regards on-line applicability, we see that the KB method is largely more expensive than the rest of the algorithms for anomaly detection, which comes directly from using the Kalman filter with large matrices. Finally, anomaly localization involves a similar number of operations for the SSB and the SPCA methods. The important conclusion that can be drawn from table 3.3 is that the SB algorithms that we propose in this work have both similar or even smaller numerical complexity for on-line anomaly detection/localization than those proposed to date.

3.7.2 Implementation Issues

We shall now discuss some important issues related to a real implementation of the proposed algorithms in a large-scale operational network. Table 3.4 presents a comparative analysis of some implementation significant features between the SB methods, the KB method, and the SPCA method. Let us discuss each of the compared items.

All the methods use SNMP measurements as input data for anomaly detection, making it possible, at least a priori, to detect volume anomalies in OD-flows without necessity of direct flow monitoring technology. This is a key feature regarding the development of light monitoring systems. However, the KB method needs anomaly-free direct OD-flow measurements for calibration purposes, loosing this advantage.

The learning data for the SB methods consists of anomaly-free SNMP measurements, while the SPCA method uses SNMP measurements not necessarily free of anomalies for calibration (collected "raw" data). There is a major difference in the duration of the learning step, which has important consequences. As we have shown in the evaluation section, the SB methods just need one hour of SNMP measurements to achieve reliable results. The KB method uses 24hs of OD-flow measurements to calibrate the underlying anomaly-free traffic model, and the SPCA method uses as much as 1 week of SNMP measurements to build the normal and anomalous subspaces.

Feature Under Comparison	SB	KB	SPCA
Input Data	SNMP	SNMP	SNMP
Learning Data	SNMP anomaly-free	TM anomaly-free	SNMP
Learning Period Length	1 hs.	24 hs.	1 week
Assumptions	strong	weak	significant
Scalability	yes	poor	yes
Supports Dynamic Routing	partially	no	no
Supports Missing Data	yes	no	yes

Table 3.4 — Implementation issues in on-line anomaly detection/localization.

The use of raw SNMP measurements in the SPCA approach is certainly useful, but as it has already been shown in previous works [76, 75] and as we have shown in section 3.6.2, there is an undeniable associated risk of learning contamination, which is definitely magnified by the lengthly learning step. The remarkably short learning step of the SB methods makes it easy for network operators to calibrate the underlying SB model without risks of contamination, as it is quite easy to collect 1 hour of SNMP measurements free of volume anomalies.

The assumptions involved in deriving the SB anomaly-free traffic model are quite strong with respect to the rest of the algorithms. Nevertheless, the validation of the SB model in three different large-scale networks shows that these assumptions are correctly verified in quite different network topologies and traffic scenarios (commercial traffic as well as research-oriented traffic). The KB method makes little assumptions on the underlying traffic model and assumes the classical Kalman filter hypotheses to be correctly verified. In practice, the Kalman filter is well known for being robust to model imprecisions, and thus we claim that the KB assumptions are weak. The SPCA method is a pure data-driven method and makes no assumptions about traffic characteristics. However and as it is pointed out in [76], there are quite significant assumptions in the heuristics used for anomaly localization that have no a priori justification and can unduly trigger alarms in some OD flows much more frequently than others.

The numerical complexity analysis previously performed shows that both SB methods as well as the SPCA method are easily scalable with the size of the network, while poor scalability can be expected from the KB method.

There is no discussion about the impacts of routing modifications over the SPCA method in the former papers [37, 72] and a constant routing matrix is used, both in the theoretical development and in the evaluation. The authors of [78] claim that the KB method can be easily extended to work with time varying routing matrices, but no discussion is provided on the involved challenges and current proposal does not support dynamic routing. The main challenge with routing modifications is that intradomain routing modifications can modify the incoming OD-flows traffic distribution due to interdomain traffic shifts. In fact, it is well known that hot-potato routing can induce

interdomain routing changes due to intradomain routing modifications. In this sense, all algorithms must be re-calibrated when an intradomain routing modification occurs, and the only methods that have a learning period length in the time scale of a routing modification are the SB methods, thus we claim that the SB anomaly detection methods can partially support routing modifications.

A similar analysis can be done regarding the application of the methods to non-stationary OD flows. Non-stationarities in traffic flows may render the underlying anomaly-free-traffic model non-longer adequate, motivating a model recalibration. The key issue is how to detect when a new recalibration must be done. In [23], authors propose a very simple heuristic to achieve this task for the underlying models of the SPCA and KB methods, monitoring the innovation process η_t of the traffic model. The decision rule is straightforward: if the innovation process is above certain threshold, a recalibration is triggered. To avoid unnecessary and expensive recalibrations due to short-lived volume anomalies, authors propose to monitor η_t during periods of 24hs, and only perform a new calibration if η_t has exceeded the threshold more than some fraction of the time.

A similar heuristic could be directly applied to the SB methods. However, there are some clear drawbacks of this approach. The first problem is related to long-lived anomalies, which may not be filtered even with a 24hs window of measurements. In fact, in this case it is not possible to distinguish between an anomaly and a model that has drifted. The second problem is that the recalibration could come many hours late, seriously affecting the performance of the detection algorithm. Our SB methods have once again the lead in this subject, due to the short and "cheap" SNMP-based learning period of the underlying model. A very simple heuristic to avoid drifting from an accurate model would be to proceed in a similar way to section 3.6.1: simply recalibrate the model if no anomaly has been declared in the last hour. Evaluations about the temporal stability of the SB model showed that this is not necessary even for several consecutive days in the real datasets that we used. Even so, we have shown that if necessary, our method can effectively be re-calibrated every hour, and thus we claim that the SB anomaly detection methods support non-stationary traffic.

The last item that we discuss concerns missing data; all algorithms use SNMP measurements as input, which has known practical limitations due to missing data and synchronization problems when collecting SNMP readings network-wide. In fact, the simultaneous collection of SNMP readings is practically impossible in very large-scale networks. The SPCA and the SB methods assume temporal independence between consecutive SNMP measurements, and thus the only impact that missing data has is a delayed verdict. In practice, it is easy to verify that all SNMP router readings are available at time t before applying the detection/localization algorithms; in case there are missing readings at time t, the methods have to delay the analysis until the following time step when data is complete.

As regards desynchronized readings, the problem is similar to missing data, and the best the algorithms can do is to delayed the analysis as before. Both problems condition the smallest feasible time scale on which the proposed methods might be used, but this is an implementation issue that depends on the particular network and thus it is impossible to give an order of this smallest time-scale. A possible solution to alleviate the problem of missing and desynchronized SNMP readings is to use oversampling and averaging, both commonly used in signal processing to reduce the effect of noisy measurements. However, this analysis is beyond the scope of our work.

Regarding the KB method, it strongly relies on the temporal dependence between consecutive SNMP measurements, and thus it can be heavily influenced by missing data. The Kalman filter can be modified so as to cope with missing data, but current KB implementation [78] does not support this practical limitation.

3.7.3 Multiple Anomaly Localization

To conclude with the discussion section, we propose some possible extensions to the presented anomaly localization algorithm. In this paper we have assumed the same simplifying hypothesis as in [72], considering only "localized" anomalies, namely anomalies in a single OD-flow at a time. However, the localization algorithm can be extended, at least in theory, to locate multiple consecutive OD-flow volume anomalies.

The multiple hypotheses $\mathcal{H}_{t_0}^k$ in (3.7) can be reformulated so as to consider multiple combinations of consecutive anomalous OD-flows as additional hypotheses to test. For example, suppose that we want to detect single OD-flow volume anomalies as well as volume anomalies that span two OD-flows at the same time. In this case, we have to add to $\mathcal{H}_{t_0}^k$ all the hypotheses that consider a volume anomaly at OD-flow i and at OD-flow j at the same time, for $0 \leqslant i \neq j \leqslant m$. This accounts for $\mathcal{C}_2^m = m!/2!(m-2)! \approx m^2/2$ additional hypotheses to test. In this case, the set of anomaly signatures is composed not only by the m single normalized columns of the routing matrix $\mathbf{r}_k$, $k = 1, \ldots, m$ but also by $\mathcal{C}_2^m$ signatures that include the two normalized columns of the routing matrix associated with the two anomalous OD-flows.

This procedure is the same as the one discussed in [72], but the idea comes from the former work of the PCA approach for fault diagnosis [68]. The problem with this approach is that the number of hypotheses to deal with, and consequently the number of decision functions $s_t(i)$ to compute grows highly and becomes very difficult to manage in a practical implementation. It is important to stress that the PCA approach [68, 72] suffers from exactly the same problem as regards anomaly localization, as the heuristics employed have a numerical complexity in the same order as our methods. The localization of multiple consecutive anomalous OD flows is out of the scope of our work.

3.8 Conclusions

In this chapter we have addressed the problem of network-wide volume anomaly detection and localization in large-scale IP networks. The following list highlights the main characteristics of the proposed solution and our major contributions to the field:

(1) Presented methods rely on coarse-grained, easily available SNMP data to detect and locate network-wide volume anomalies in OD-flows traffic. This is a main advantage in order to develop light monitoring systems without the necessity of direct flow measurement technology, particularly in the advent of the forecast massive traffic to analyze in the near future.

(2) We have shown how to use the linear, parsimonious, SB traffic model presented in chapter 2 to remove the anomaly-free traffic from the anomaly detection problem, motivating our original approach of treating the detection and localization of volume anomalies as a statistical change detection/localization problem with a nuisance parameter. This a-priori simple characteristic allows to construct optimal algorithms for volume anomaly detection and localization.

(3) We have developed different methods for volume anomaly detection and localization with a paramount advantage with respect to previous works in the field, that of having solid optimality properties in terms of false alarm rate, detection/localization delay, and false localization rate. This represents a major breakthrough in the field and the most important contribution of this chapter. We argue that optimality support is fundamental in the conception of general algorithms, not tied to any particular network or evaluation.

(4) Using extensive data from three real backbone networks we have shown that the theoretical optimality properties of the proposed algorithms are verified in practice, providing results that outperform current network-wide anomaly detection/localization methods in a wide variety of network topologies and traffic scenarios.

(5) The complexity analysis has shown that our algorithms are more efficient than current methods to perform anomaly detection and localization on-line with even better results. We believe that a real implementation of our optimal algorithms could be envisaged without any modifications to current technology.

We expect that the proposed solutions in this chapter will stimulate in the future the development of anomaly detection algorithms with a solid theoretical background, allowing a robust growth of the network monitoring field. We believe that the results of decision theory applied to the field of network monitoring are still not sufficient and worthy to extend. This chapter contributes to bridging the gap between these two fields.

CHAPTER 4

Routing Optimization Under Traffic Uncertainty

Internet traffic is highly dynamic and difficult to predict in current network scenarios. As we have discussed and evidenced in chapters 2 and 3, traffic variations in large-scale networks present not only a slow predictable component due to normal traffic usage patterns (e.g. daily demand fluctuation) but also an abrupt and unpredictable behavior represented by volume anomalies. As we have explained in the Introduction, the future traffic scenario points in the same direction: more heterogeneous and bandwidth aggressive applications, spontaneous and difficult to predict traffic events, emergent business models which modify known traffic patterns. These evidences and forecasts coupled with the difficulty involved in the measurement of traffic in a large-scale network makes of network traffic a highly uncertain quantity.

So far in the thesis we have addressed two of the main sources of this traffic uncertainty problem, considering different approaches. Let us briefly recall them. In chapter 2, we addressed the problem of inferring the traffic demand of a complete Autonomous System from aggregated and easily available data, modeling and analyzing traffic behavior with a wide variety of techniques. In chapter 3 we took a step forward, using some of these traffic models to automatically detect and localize unexpected traffic events. The question that we pose after this analysis is what can we do to fightback this traffic variability and uncertainty? Recall that one of the main drivers in networking, if not the most important, is to provide services with some level of performance, in the wide sense of the concept, from connectivity to Quality of Service. Consequently, we adopt in this chapter a different perspective to the problem, and focus the attention on how to adapt the network to perform properly under this traffic uncertainty.

The performance of the Internet itself depends, in large part, on the operation of the underlying routing protocols. Routing optimization becomes increasingly challenging due to the dynamic and uncertain nature of current traffic. In this chapter, we explore different routing optimization mechanisms, paying special attention to the traffic uncertainty issue. Today's routing protocols in IP networks compute routing configurations based on the network topology and some rough knowledge of traffic demands (e.g., worst-case traffic, average traffic, long-term forecasts), without regard to the current traffic load on routers and links or possible traffic misbehaviors. In this scenario, the responsibility to adapt the routing scheme to the prevailing traffic

falls on network operators and network management systems. Large-scale networks are usually over-provisioned, and routing modifications due to traffic variations are not that often. However, the evolution and deployment-rate of broadband access technologies (e.g. Fiber To The Home) is such that the assumption of infinitely provisioned core links will soon become obsolete, and simply upgrading link capacities may no longer be an economically viable solution. Recent studies in the field of routing optimization agree that today's approach may no longer be suitable to manage current and future traffic patterns.

In the light of this scenario, network operators search for *reliable* routing mechanisms. In the most general form, the term network *reliability* refers to the ability of the network to perform and maintain its functions in the case of component failures. In our case, we extend the idea of network reliability to the case of unexpected traffic events, represented by volume anomalies. Two almost antagonist approaches have emerged in the recent years to cope with both the traffic increasing dynamism and uncertainty and the need for cost-effective solutions: Robust Routing (RR) and Dynamic Load-Balancing (DLB). Briefly, the Robust Routing approach copes with traffic uncertainty in an off-line preemptive fashion, computing a fixed routing configuration that is optimized for a large set of possible traffic demands. On the other hand, dynamic load balancing delivers traffic among multiple fixed paths in an on-line reactive fashion, adapting to traffic variations.

In Robust Routing, traffic uncertainty is taken into account directly within the routing optimization, computing a single routing configuration for all traffic demands within some *uncertainty set* where traffic is assumed to vary. This uncertainty set can be defined in different ways, depending on the available information: busy-hour traffic, largest values of links load previously seen, a set of previously observed traffic demands (measured in the previous day, in the same day of the previous week, etc.), etc. The criteria to search for this unique routing configuration is generally to minimize the maximum link utilization over all demands of the corresponding uncertainty set. While this routing configuration is not optimal for any single traffic demand within the set, it minimizes the worst case performance over the whole set, providing performance bounds.

Dynamic Load Balancing copes with traffic uncertainty and variability by splitting traffic among multiple paths on-line. In this dynamic scheme, each origin-destination pair of nodes within the network is connected by several a priori configured paths, and the problem is simply how to distribute traffic among these paths in order to optimize a certain cost function. DLB is generally defined in terms of a link-cost function, where the portions of traffic are adjusted by each origin-destination pair of nodes in order to minimize the total network cost.

Those who promote DLB highlight among others the fact that it represents the most resource-efficient solution, adapting to current network load in an automated and decentralized fashion. Those who advocate the use of RR claim that there is actually no need to implement supposedly complicated dynamic routing mechanisms, and that the incurred performance loss for using a single routing configuration is negligible when compared with the increase in complexity. An interesting characteristic of RR relies on the use of a single fixed routing configuration, avoiding possible instabilities due to routing modifications. In practice, network operators are reluctant to use dynamic mechanisms and prefer fixed routing configurations, as they claim they get a better feeling of what is going on in the network. In fact, most operational networks fall into this category.

In this chapter we study the problem of Routing Optimization under traffic uncertainty. As we did in chapters 2 and 3, we focus on a single AS, adapting the intradomain routing configuration to traffic dynamics. We begin firstly by analyzing the Robust Routing paradigm, introducing its main features and providing evidence of its advantages with respect to traditional IP routing optimization techniques. Secondly, we show that despite achieving routing reliability with relatively low performance loss, RR presents various drawbacks and conception problems as it is currently proposed. We present and evaluate different variants of RR to alleviate these shortcomings, keeping the robustness of the approach against traffic uncertainty. These variants include three different proposals: a Multi-Hour Robust Routing (MHRR) approach, a Reactive Robust Routing (RRR) approach, and an End-to-End Quality of Service-based Robust Routing (QoS-RR) approach.

The first shortcoming that we identify in a Robust Routing paradigm is the associated cost-efficiency. Using a single routing configuration for long-time periods can be highly inefficient. The definition of the uncertainty set in RR defines a critical trade-off between performance and robustness: larger sets allow to handle a broader group of traffic demands, but at the cost of routing inefficiency; conversely, tighter sets produce more efficient routing schemes, but subject to poor performance guarantees. Based on expected traffic patterns, we show that it is possible to adapt the uncertainty set and build a Multi-Hour yet robust routing scheme that outperforms the stable approach. For the case of volume anomalies, we propose a dynamic extension of RR known as Reactive Robust Routing (RRR). The RRR approach uses the sequential volume anomaly detection/localization method introduced in chapter 3 to rapidly detect and localize abrupt changes in traffic demands and decide routing modifications. We propose a load balancing approach for RRR, in which traffic is balanced among fixed paths according to a centralized entity that controls the fractions of traffic sent on each path. Naturally, we present a comprehensive comparative analysis between the RRR approach and different DLB algorithms. In such study we provide substantial evidence on the virtues and shortcomings of both mechanisms, on the one hand by quantifying the performance loss of RRR with respect to DLB, and on the other hand by analyzing the temporal response of RRR and DLB under significant and unpredicted traffic variations.

The second drawback that we identify in current Robust Routing is related to the objective function it intends to minimize. Optimization under uncertainty is generally more complex than classical optimization, which forces the use of simpler optimization criteria such as the maximum link utilization (MLU), i.e., minimize the load of the most utilized link in the network. The MLU is by far the most popular Traffic Engineering objective function, but clearly it is not the most suitable network-wide optimization criterion; setting the focus too strictly on MLU often leads to worse distributions of traffic, adversely affecting the mean network load and thus the total network end-to-end delay, an important QoS indicator. It is easy to see that the minimization of the MLU in a network topology with heterogeneous link capacities may lead to poor results as regards global network performance.

To avoid this issue, we firstly propose to minimize the mean link utilization instead of the MLU. The mean link utilization provides a better image of network-wide performance, as it does not depend on the particular load or capacity of each single link in the network but on the average value. Despite this advantage, a direct minimization of the mean link utilization does not assure a bounded MLU, which is not practical from an operational point of view. Thus, we minimize the mean link utilization while bounding the MLU by a certain utilization threshold a priori defined. This adds a new difficult to set constraint to the problem, namely how to define this utilization threshold. We further improve our proposal by providing a multiple objective optimization criterion, where both the MLU and the mean link utilization are minimized simultaneously. We evaluate the improvements of our proposals from a QoS perspective, using the mean path end-to-end queuing delay as a measure of global performance. The evaluations presented in this chapter show that this approach attains better global performance from an end-to-end quality of service perspective.

Combining the different extensions of the traditional Robust Routing paradigm, we develop a robust approach to rapidly fightback the side-effects of volume anomalies, strengthening the global QoS of the network in the event of strong and abrupt congestion situations.

This remainder of this chapter is organized as follows. Section 4.1 presents the state of the art in the field of Robust Routing and Dynamic Load Balancing. In section 4.2 we analyze the Robust Routing paradigm, comparing its performance against traditional routing optimization techniques. The first contribution of this chapter is presented in section 4.3, where we introduce and evaluate a Multi-Hour extension for Robust Routing. In section 4.4 we introduce a new proposal to rapidly adapt the routing configuration in the event of volume anomalies, known as the Reactive Robust Routing (RRR). The idea in RRR is to reduce the impacts of these large traffic variations on the network's global performance. Different rerouting strategies are proposed for RRR, including complete and partial routing reconfiguration and reactive load balancing. Section 4.6 presents a variant of RR that attains better global performance, particularly regarding end-to-end Quality of Service metrics like end-to-end delay. In section 4.6 we present a comprehensive comparative study between RRR and DLB. In this section

we describe some traditional DLB schemes and additionally propose new variants to improve their performance. Finally, section 4.7 presents the lessons learnt from this study and some concluding remarks.

4.1 State of the Art

Given a single known Traffic Matrix (TM), Routing Optimization consists in computing a set of origin-destination paths and a traffic distribution among these paths in order to optimize some performance criterion, usually expressed by means of a cost function. This is a well known multi-commodity flow problem, easily solved by linear programming techniques [96]. However and as we have explained before, in practice, traffic demands vary in time and are usually difficult to measure, turning routing optimization into a challenging problem.

Traditional routing optimization approaches have addressed the problem relying on either a small group of *representative* TMs or estimated TMs to compute optimal and reliable routing configurations. These techniques usually maintain a history of observed TMs in the past, and optimize routing for the representative traffic extracted from the observed TMs during a certain history window. In this sense, we shall refer to traditional algorithms as Prediction-Based Routing. The most well-known and traditionally applied prediction-based routing approach is a simple rule-of-thumb: optimize routing for a worst-case traffic scenario, using for example the busy-hour TM seen in the history of traffic [94]. This may not be a cost-effective solution, but operators have traditionally relied on the over-provisioning of their networks to support the associated performance loss. In the search for a better solution, authors in [97] propose to optimize routing for an estimated TM. This approach can provide more efficient solutions, but it highly depends on the goodness of the estimation technique, and as we show in next section, it may even result in unreliable routing configurations for the real traffic. Other papers like [98, 99] have proposed to optimize routing for multiple TMs simultaneously, using for example a finite number of TMs from a previous day, from the same day of a previous week, etc. In these works, authors derive different methods to find routing configurations with good average and worst case performance over these TMs. Note that in this approach, the worst case is only among the samples used in the optimization, and not among all possible TMs. All these traditional approaches tend to work reasonably well, but they certainly require a *leap of faith* from operators and a lot of management effort to ensure robustness against unexpected traffic variations.

A different approach has emerged in the recent years to cope with the traffic increasing dynamism and the need for cost-effective solutions, Dynamic Load-Balancing (DLB) [110, 111, 112, 114]. In DLB, traffic is split among fixed a priori established paths in order to optimize a certain cost function. The two most well-known proposals in this area are MATE and TeXCP. In MATE [110], a convex link cost function is defined, which depends on the link capacity and the link load. The objective is to minimize the total network cost, for which a simple gradient descent method is proposed. TeXCP [111] proposes a somewhat simpler objective: minimize the biggest utilization each traffic demand obtains in its paths. A rough description of the algorithm is that origin nodes iteratively increase the portion of traffic sent through the path with the smallest utilization. Another DLB scheme which has the same

objective but a relatively different mechanism is REPLEX [112]. In [114], authors use a link cost function based on measurements of the queuing delay, which results in better global performance from a QoS perspective. DLB presents a desirable property, that of keeping routing adapted to dynamic traffic. However, DLB algorithms present a trade-off between adaptability and stability which might be particularly difficulty to address under significant and abrupt traffic changes.

The Robust Routing paradigm (RR), in all of its *flavors* [101, 102, 103, 104, 105, 106, 107, 108] represents another recently proposed solution to the routing optimization under traffic uncertainty problem. The objective in RR is to find a fixed routing configuration that fulfills a certain criterion for an infinite set of traffic demands, known as *uncertainty set*. The criterion is generally the one that minimizes the Maximum Link Utilization (MLU) over the uncertainty set of demands. In [101], authors capture traffic variations by introducing a polyhedral uncertainty set of demands, applying linear programming techniques to compute an optimal stable routing for all demands within this set. [103] applies this robust technique to compute a robust MPLS routing configuration without depending on traffic demand estimation, and discusses corresponding methods for robust OSPF optimization. The same approach is extended to explicitly manage potential traffic shifts due to BGP reroutes in [106]. Oblivious Routing [102] also defines linear algorithms to optimize worst-case MLU for different sizes of traffic uncertainty sets, aiming to handle dynamic changes. Authors in [105] propose a two-phase routing scheme that allows to use fixed pre-configured bandwidth paths to handle traffic variations, bounded by a Hose model. In the Hose model [100], the total amount of traffic that enters/leaves an ingress/egress node in the network is bounded. Using this routing scheme, [107] develops linear programming formulations to minimize the MLU. [108] analyses the use of RR through a combination of traffic estimation techniques and its corresponding estimation error bounds, in order to shrink the uncertainty set of traffic demands. In [104] authors introduce a RR mechanism that optimizes routing for predicted demands, bounding worst-case MLU to ensure acceptable efficiency under unexpected traffic events. A major drawback of RR is its inherent dependence on the definition of the uncertainty set: larger sets allow to handle a broader group of traffic demands, but at the cost of routing inefficiency; conversely, tighter sets produce more efficient routing schemes, but subject to poor performance guarantees. Another problem related to RR is that optimization under uncertainty is generally more complex than classical optimization, which forces the use of simpler optimization criteria.

As regards a comparative study between RR and DLB, to the best of our knowledge the only work that performs certain analysis is [104]. In this work, authors compare the performance of their RR mechanism with a dynamic approach which they claim models the behavior of mechanisms such as MATE and TeXCP. Given a time-series of traffic demands, this dynamic approach consists of computing an optimal routing for each traffic demand i and evaluate its performance with the following traffic demand $i+1$. There are two important shortcomings of this DLB simulation. Firstly, adaptation in DLB is iterative and never instantaneous. Secondly, in all DLB mechanisms paths

are set a priori and remain unchanged during operation. This is not the case in their dynamic approach, where each new routing optimization may change not only traffic portions but paths themselves. For these reasons, we believe that the comparison provided in [104] is biased against dynamic schemes.

4.2 Prediction-Based Routing and Robust Routing Optimization

In order to present the Routing Optimization problem, we shall begin by introducing the notation used in this chapter. Similar to chapters 2 and 3, the network topology is defined by n nodes and a set $L = \{l_1, \ldots, l_r\}$ of r links, each with a corresponding capacity c_i, $i = 1, \ldots, r$. The Traffic Matrix (TM) $X_t = \{x_t(k)\}$ denotes the traffic volume of each OD flow $k = 1, \ldots, m$ at time t, being $m = n.(n-1)$. Let $N = \{\text{OD}_1, \ldots, \text{OD}_m\}$ be the set of m OD pairs of nodes, associated with the m OD flows of traffic. We consider a multi-path network topology, where each OD flow $x_t(k)$ can be arbitrarily split among a set of origin-destination paths P_k. We shall use r_p^k as the portion of traffic flow $x_t(k)$ sent through path $p \in P_k$, where $0 \leqslant r_p^k \leqslant 1$ and $\sum_{p \in P_k} r_p^k = 1$. Let λ_l^p be an indicator variable that takes value 1 if path p traverses link l and 0 otherwise, and $Y_t = \{y_t(1), \ldots, y_t(r)\}$ the links traffic load at time t. Let us first consider a single-time TM X, which permits us to omit the time subindex. Recall that X and Y are related through the routing matrix R, a $r \times m$ matrix $R = \{r_l^k\}$ where $r_l^k = \sum_{p \in P_k} \lambda_l^p \,.\, r_p^k$. The variable r_l^k indicates the fraction of OD flow $x(k)$ routed through link l. This results in the so far so used relation $Y = R \,.\, X$.

Given X, the multi-path routing optimization problem consists in selecting the set of paths P_k for each OD pair k and computing the routing matrix R, in order to optimize a certain objective function $f(X, R)$. A simplified version of this problem is the optimal load-balancing problem, in which a routing matrix R is computed, but for a *given* fixed set of paths. In other words, in load-balancing P_k is already given, and the optimization is done with respect to the values r_p^k only. The most popular TE objective function $f(X, R)$ has traditionally been the Maximum Link Utilization (MLU) $u_{\max}$, defined as:

$$u_{\max}(X, R) = \max_{l \in L} \{u_l\}$$

where $u_l = y(l)/c_l$ stands for the link utilization; a value of u_l close to 1 indicates that the link is operating near its capacity. Overloaded links tend to cause QoS degradation (e.g. larger delays and packet losses, throughput reduction, etc.), so MLU represents a reasonable measure of network performance, quite basic and indirect but very used in practice. Network operators usually prefer to keep links utilization values relatively low in order to support sudden traffic increases and link/node failures.

The computation of a multi-path routing configuration that minimizes $u_{\max}$ is an instance of the classical multi-commodity flow problem, which can be formulated as a simple linear program [92]. For a single known traffic matrix X, the problem can be easily solved by linear programming techniques [96]. The system defined in (4.1) represents an arc-path formulation to this multi-path routing optimization problem.

$$
\begin{aligned}
&\text{minimize} && u_{\max} && && (4.1)\\
&\text{subject to:} && && && \\
&\sum_{k\in N}\sum_{p\in P_k} \lambda_l^p \,.\, r_p^k \,.\, x(k) && \leqslant && u_{\max}.c_l \qquad \forall\, l \in L && (4.2)\\
&\sum_{p\in P_k} r_p^k && = && 1 \qquad \forall\, k \in N && (4.3)\\
&u_{\max} && \leqslant && 1 && (4.4)\\
&r_p^k && \geqslant && 0 \qquad \forall\, p \in P_k,\ \forall\, k \in N && (4.5)
\end{aligned}
$$

All problem constraints are linear. The set of constraints (4.2) define the concept of u_{max}: the total traffic across each link l is bounded. Constraints (4.3) express the multi-path property of the routing: each OD flow k can be transmitted through different paths p in P_k, and every flow must be completely routed. Finally, constraint (4.4) specifies that routing must be stable. From an algorithmic point of view, this is an easy to solve linear programming problem. The classical way of solving problem (4.1) is by column generation [94], where a *column* represents a new candidate path. Rather than explicitly enumerating all paths in the network, the algorithm begins with a small subset of paths (e.g., the shortest-hop routing) and then sequentially adds new paths to the problem to improve the solution, based on the reduced cost of the paths. However, as we have previously discussed, traffic demands are uncertain and difficult to predict in current scenario, and all we can expect is to find the real value of the TM within some bounded uncertainty set.

In a robust perspective of the multi-path routing optimization problem, demand uncertainty is taken into account within the routing optimization, computing a single routing configuration for all demands within some uncertainty set $\mathbb{X}$. The idea of traffic uncertainty set comes from the context of Virtual Private Network (VPN) provisioning, where the traffic exchanged between interconnected customer sites is bounded, according to certain traffic profiles. The conventional approach to define these traffic profiles is by means of the *pipe model*. In this model, the volume of traffic in each OD flow k of the TM is bounded by some fixed upper-bound $x_{\max}(k)$, such that:

$$\mathbb{X} = \{X \in \mathbb{R}^m,\ x(k) \leqslant x_{\max}(k),\ \forall k = 1 \ldots m,\ X \geqslant 0\}$$

Another very popular model to define a traffic uncertainty set is the *hose model* [100]. In the hose model, the total volume of traffic that leaves from an origin node and enters at a destination node in the network is bounded:

$$\mathbb{X} = \left\{ X \in \mathbb{R}^m,\ X \geqslant 0,\ \begin{array}{rcll} \sum\limits_{j=1, j\neq i}^{m} x(i,j) & \leqslant & x_{\max}^{\text{out}}(i), & \forall i = 1 \ldots m \\ \sum\limits_{i=1, i\neq j}^{m} x(i,j) & \leqslant & x_{\max}^{\text{in}}(j), & \forall j = 1 \ldots m \end{array} \right\}$$

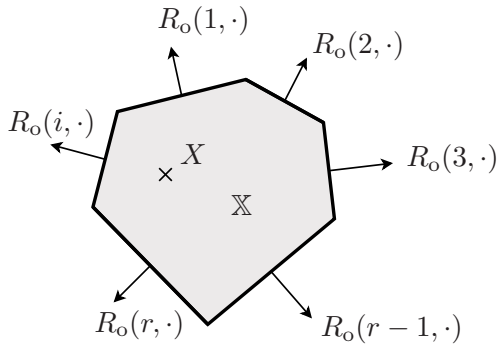


Figure 4.1 — The uncertainty set $\mathbb{X}$ as a polytope.

where $x(i,j)$ stands for the traffic volume exchanged between an origin node i and a destination node j. There are some other approaches to define an uncertainty set of traffic demands. For example, in [103], the authors define an uncertainty set based on the convex intersection of several TMs $X_1, X_2, \ldots, X_T$ measured at different times of the day:

$$\mathbb{X} = \boldsymbol{co}\{X_1, X_2, \ldots, X_T\}$$

where ***co*** stands for the convex-hull of the TMs. In this chapter we consider a polyhedral uncertainty set $\mathbb{X}$, more precisely a *polytope* as in [101], based on the intersection of several half-spaces that result from linear constraints imposed to traffic demands. This is probably the most general approach, as every previous model can be treated as a particular case of a polyhedral model. As an example, let us define a polyhedral uncertainty set $\mathbb{X}$ based on a given routing matrix R_o and the busy-hour links traffic load Y^{busy} obtained while using this routing matrix:

$$\mathbb{X} \;=\; \left\{X \in \mathbb{R}^m, R_o\, X \leqslant Y^{\text{busy}},\ X \geqslant 0\right\}$$

This definition of uncertainty set is highly flexible and easy to understand, and provides a linear analytical means of verifying if certain TM X belongs to $\mathbb{X}$ or not. The polytope model provides a major advantage with respect to previously defined models: routing optimization can be performed from easily available links traffic load Y without even knowing the actual value of the traffic demand X. Figure 4.1 depicts the obtained uncertainty set, based on the convex intersection of r half-spaces in the form of $R_o(i,\cdot)\, X \leqslant y(i)^{\text{busy}}$, $\forall i \in L$, where $R_o(i,\cdot)$ stands for the i-th row of the routing matrix R_o.

Now that the concept of uncertainty set has been introduced, we shall define the *Robust Routing Optimization Problem* (RROP). The RROP defined in (4.6) consists in minimizing the maximum link utilization $u_{\max}$, considering all demands within $\mathbb{X}$. The solution to this problem is twofold: on the one hand, a routing configuration R_{robust}, which consists in both the origin-destination paths and the traffic distribution, and on the other hand, a worst-case performance threshold $u_{\max}^{\text{robust}}$:

$$
\begin{array}{rcll}
\text{minimize} & u_{\max} & & (4.6)\\
\text{subject to:} & & & \\
\sum_{k\in N}\sum_{p\in P_k} \lambda_l^p \,.\, r_p^k \,.\, x(k) & \leqslant & u_{\max}.c_l \qquad \forall\, l \in L, \forall\, X \in \mathbb{X} & (4.7)\\
\sum_{p\in P_k} r_p^k & = & 1 \qquad \forall\, k \in N & (4.8)\\
u_{\max} & \leqslant & 1 & (4.9)\\
r_p^k & \geqslant & 0 \qquad \forall\, p \in P_k,\ \forall\, k \in N & (4.10)
\end{array}
$$

$$
\begin{aligned}
R_{\text{robust}} &= \operatorname*{argmin}_{R} \max_{X\in\mathbb{X}} u_{\max}(X, R)\\
u_{\max}^{\text{robust}} &= \max_{X\in\mathbb{X}} u_{\max}(X, R_{\text{robust}})
\end{aligned}
$$

The paramount contribution of Robust Routing is that, given a suitable definition of the uncertainty set, the obtained robust routing configuration R_{robust} can be used during long periods of time, without the need of routing reconfigurations. In this sense, we refer to Robust Routing as **Stable Robust Routing** (SRR). An additional property of SRR is that it provides worst-case performance bounds for all traffic demands within the uncertainty set, as $u_{\max}(X, R_{\text{robust}}) \leqslant u_{\max}^{\text{robust}}, \forall X \in \mathbb{X}$.

At first glance, the RROP looks like a really difficult to solve problem. Indeed, the inclusion of the uncertainty set largely modifies the traditional problem. In particular, the set of constraints (4.7) is now an infinite set, because the uncertainty set $\mathbb{X}$ has infinite TMs. Nevertheless, authors in [101] propose a simple algorithm to efficiently solve the problem by generalized linear programming.

The first thing to notice is that $\mathbb{X}$ is a convex set, more precisely a polytope. Any polytope can be represented as the convex-hull of its extreme points. An extreme point of a polyhedron is a point that cannot be written as a convex combination of other points in the polyhedron. Thus, if constraints (4.7) are verified for every extreme point of $\mathbb{X}$, then they are verified for every TM in $\mathbb{X}$. The interesting thing is that a polyhedron has a finite number of extreme points, and therefore, constraints (4.7) have to be verified for a finite set only. However, the number of extreme points can be still very high, and considering all the constraints related to each extreme point of $\mathbb{X}$ can be very difficult, simply because it may not be that easy to compute them all from scratch. The solution proposed in [101] consists in using a constraints generation procedure, in order to add the inequalities of type (4.7) that are not satisfied by the current solution without explicitly enumerating all the extreme points of $\mathbb{X}$ from the begging. Similar to the column generation procedure, the algorithm begins with a small subset of constraints and then iteratively adds new constraints to the problem, corresponding to the extreme points of polytope $\mathbb{X}$.

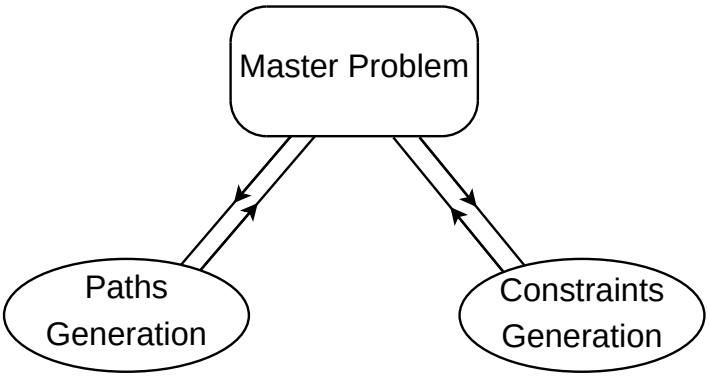


Figure 4.2 — A combined columns and constraints generation method to solve the Robust Routing Optimization Problem.

The complete algorithm to solve the RROP results in a combined columns and constraints generation method. Figure 4.2 depicts a high-level description of this method. Starting from an initial subset of paths $P_k^{(o)}$, $\forall k = 1 \dots m$ and an initial subset of traffic demands $\mathbb{X}^{(o)}$, the i-th iteration of the *Master Problem* corresponds to problem (4.6), using the set of paths $P_k^{(i)}$ and the set of extreme points $\mathbb{X}^{(i)}$ so far generated/added. Given a solution to this master problem, the *Paths Generation* problem consists in adding new paths to decrease the value of the objective function, and the *Constraints Generation* problem consists in adding new extreme points of $\mathbb{X}$ so as to completely cover the uncertainty set. The convergence of the algorithm is guaranteed for the following reasons: the number of paths that can be generated is finite, the number of extreme points of $\mathbb{X}$ is finite, no path can be generated more than once, and no constraint of type (4.7) can be added more than once.

Adding new paths only based on their reduced cost may not be the best choice from a practical point of view, since the number of paths for each OD pair is not a priori restricted and the characteristics of new paths are not controlled. For example, it would be interesting to have disjoint paths to route traffic from each single OD pair, improving resilience. For this reason we modify the Paths Generation method, both limiting the maximum number of paths in P_k and taking as new candidates the shortest paths with respect to link weights w_l^i:

$$w_l^i = \frac{1}{\epsilon + \left(1 - r_l^{k^i}\right)}$$

where $r_l^{k^i}$ corresponds to the fraction of traffic flow $x(k)$ that traverses link l after iteration i, and ϵ is a small constant that avoids numerical problems. If OD pair k uses a single path p at iteration i, $r_l^{k^i} = 1$ for every link $l \in p$, and so this path is removed from the graph where new shortest paths are computed ($w_l \to \infty$, $\forall l \in p$). While this may result in a sub-optimal performance, it allows a real and practical implementation. In case there are no disjoint paths for OD pair k, we use the standard column constraint generation method to add new paths for OD pair k.

In the following evaluations we compare the performance of the traditional Prediction-Based Routing approach and the previously described Robust Routing method. We consider two different case-studies, the former based on routing optimization for an instantaneous uncertain TM, and the latter based on routing optimization for time-varying TMs. The study is performed in Abilene, the Internet2 backbone network used in previous chapters.

4.2.1 Routing Optimization for Instantaneous Traffic

Let us first consider the problem of routing optimization for an instantaneous TM, i.e., a TM at a certain fixed time. We shall include the notion of traffic uncertainty related to the problem of traffic matrix estimation, assuming that we only know the current routing configuration and the link load measurements at this fixed time. Based on this data, we compare three different approaches: the *traditional approach* is to estimate a single TM using a TomoGravity estimation method and then perform a routing optimization based on this estimated TM. The *robust approach* consists in Robust Routing optimization for an uncertainty set that includes all the TMs that are consistent with current routing and link load measurements. The *ideal approach* consists in routing optimization for the real TM, assumed known.

Let R_{o} be the fixed routing matrix of Abilene that is available at the Abilene dataset [129]. This routing matrix is not necessarily optimized for any particular TM. Given a single snapshot of link loads Y_o, we shall define an uncertainty set $\mathbb{X}$ as:

$$\mathbb{X} = \{X \in \mathbb{R}^m, R_{\mathrm{o}}\, X \leqslant Y_{\mathrm{o}},\ X \geqslant 0\}$$

In the robust scenario, a robust routing configuration R_{robust} is computed for $\mathbb{X}$, using (4.6). In the ideal scenario, the real TM X^* is completely known, and the optimal routing configuration R_{opt} is computed for this TM, using (4.1). In the traditional scenario, a TM $\hat{X}$ is estimated from R_{o} and Y_{o}, and the routing configuration $\hat{R}_{\mathrm{opt}}$ is computed for this TM, using (4.1) once again. We compute the real value of the maximum link utilization for these three routing configurations, using the real TM X^*:

$$\begin{aligned} u^*_{\mathrm{max}} &= u_{\mathrm{max}}(X^*, R_{\mathrm{opt}}) \\ \hat{u}_{\mathrm{max}} &= u_{\mathrm{max}}(X^*, \hat{R}_{\mathrm{opt}}) \\ u^{\mathrm{robust}}_{\mathrm{max}} &= u_{\mathrm{max}}(X^*, R_{\mathrm{robust}}) \end{aligned}$$

The real TM could be in fact any point in the polytope. In order to measure the robustness of each approach against this uncertainty, we also compute the worst-case of the maximum link utilization that could be obtained with both $\hat{R}_{\mathrm{opt}}$ and R_{robust} in $\mathbb{X}$:

Time	**02:00**	**08:00**	**14:00**	**20:00**
$\hat{u}_{\max}$	1.18	1.03	1.07	1.07
$u_{\max}^{\text{robust}}$	1.07	1.14	1.15	1.13
$\hat{u}_{\max}^{\text{wc}}$	4.71	4.87	5.75	5.01
$u_{\max}^{\text{robust wc}}$	1.10	1.15	1.16	1.14

Table 4.1 — Routing performance under traffic uncertainty, relative to $u^*_{\max}$.

$$\begin{aligned} \hat{u}_{\max}^{\text{wc}} &= \max_{X \in \mathbb{X}} u_{\max}(X, \hat{R}_{\text{opt}}) \\ u_{\max}^{\text{robust wc}} &= \max_{X \in \mathbb{X}} u_{\max}(X, R_{\text{robust}}) \end{aligned}$$

We repeat the same evaluation for different snapshots during the day. For each of them, ideal, traditional and robust routing performances are compared. Table 4.1 summarizes the results of this comparison. For simplicity, we take $u^*_{\max} = 1$ as reference. Let us consider the obtained results for the 14:00 time of day (column 14:00). If the value of the traffic demand were known, the MLU would be $u^*_{\max}$. In practice, it is difficult to perfectly know the value of the traffic demand, so an estimation is used. If the routing is optimized for the estimated value $\hat{X}$, then the performance of that routing $\hat{R}_{\text{opt}}$ when the traffic demand value is X^* is $1.07u^*_{\max}$. Thus, the performance degradation due to the estimation is 7%, which is reasonable provided that the links utilization is not too large. This means that the estimate is sufficiently close to the true value of the demand to make routing possible and efficient, at least in the case of the Abilene dataset [129]. But in theory, the only thing that we can be sure of is that X^* belongs to the uncertainty set $\mathbb{X}$ and nothing proves with certainty that X^* and the estimated value $\hat{X}$ are *close*. If we take into consideration that the TM takes any value in $\mathbb{X}$, then the MLU can reach $5.75u^*_{\max}$ in the worst case and this is obviously a risk that any operator would be ready to take.

Now let us suppose that this uncertainty is taken into account preventively in the routing optimization. In that case, the MLU when the TM value is X^* is $1.15u^*_{\max}$; compared to the performance of the traditional approach, the robust routing "cost" is $1.15 - 1.07 = 0.08$, i.e., a 8% performance degradation. But the MLU in the robust routing case will always be bounded by $1.16u^*_{\max}$, whatever the value of $X \in \mathbb{X}$. Compared to the $5.75u^*_{\max}$ worst case performance of the traditional approach, it is clear that the robust approach offers a guarantee against the uncertainty on the traffic demand value, for a cost which remains reasonable (8%). Finally, figure 4.3 depicts a summary of the results. This graphical representation of the values presented in table 4.1 permits to appreciate the robustness provided by the robust routing approach with respect to a traditional TM estimation-based approach, where the traffic uncertainty related to the problem of traffic estimation can dramatically impact routing performance, even obtaining unreliable routing configurations.

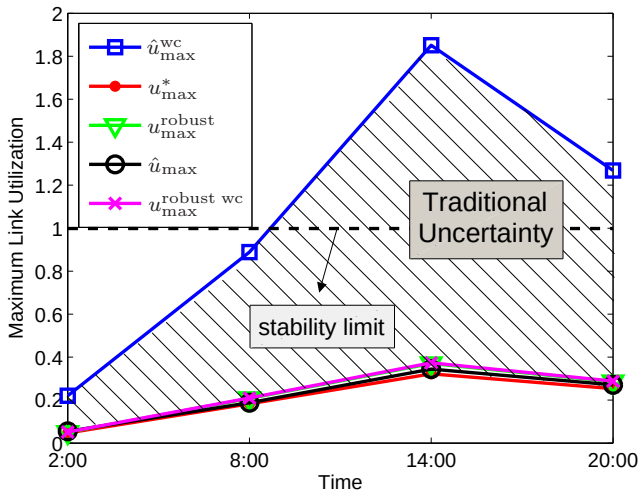


Figure 4.3 — Robustness of routing optimization facing Traffic Matrix estimation.

4.2.2 Robust Routing and Prediction-Based Routing with Time-Varying TMs

Robust optimization can also be used to handle time-varying traffic demands. Instead of working with a single snapshot TM, we assume now that we have a full time-series of TMs during certain time-window. The traditional approach to handle time-varying TMs generally consists in routing optimization for the busy-hour TM or a TM with the maximal traffic demands, obtained from historical traffic measurements. In the Stable Robust Routing approach previously described, an uncertainty set can be defined in terms of some upper bound Y^{UB} in the links traffic, extracted from traffic measured in the past.

$$\mathbb{X}\left(Y^{\mathrm{UB}}\right) = \left\{X \in \mathbb{R}^m, R_{\mathrm{o}}\, X \leqslant Y^{\mathrm{UB}},\ X \geqslant 0\right\}$$

An obvious example for this upper-bound would be the links capacities, $Y^{\mathrm{UB}} = C$, with $C = \{c_i\}_{i=1\ldots r}$. This upper-bound would ensure routing stability in terms of link utilization, but given the size and shape of the polytope it could result in a very inefficient routing configuration. A more interesting upper-bound would be the busy-hour traffic load $Y^{\mathrm{UB}} = Y^{\mathrm{busy}}$, or even some *preventive threshold* could be envisioned for unexpected traffic variations $Y^{\mathrm{UB}} = \lambda.Y^{\mathrm{busy}}$, with $\lambda \geqslant 1$.

Under normal operation conditions, it seems quite clear to admit that the difference between a traditional busy-hour-based routing approach and the Stable Robust Routing approach for a busy-hour-based polytope $\mathbb{X}(Y^{\mathrm{busy}})$ is not important. Figure 4.4 evidences this behavior. The evaluation consists of 288 consecutive TMs from the Abilene dataset, which corresponds to a 24hs period. The full line corresponds to an ideal adaptive routing, where routing configuration is re-optimized for each single TM, using (4.1). The dashed line corresponds to a fixed robust routing configuration using polytope $\mathbb{X}(Y_{220})$, which corresponds to a polytope defined on the basis of links

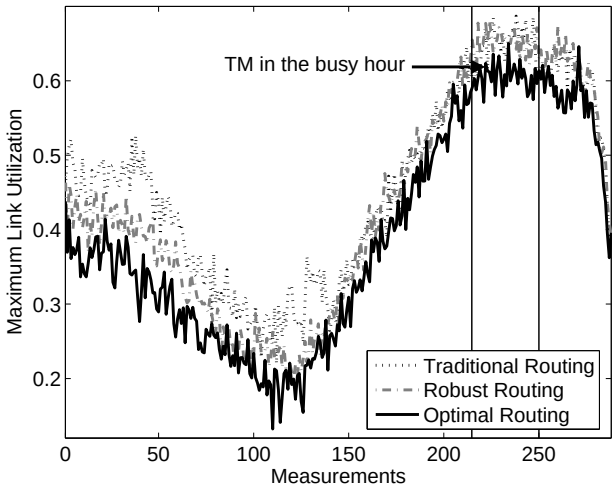


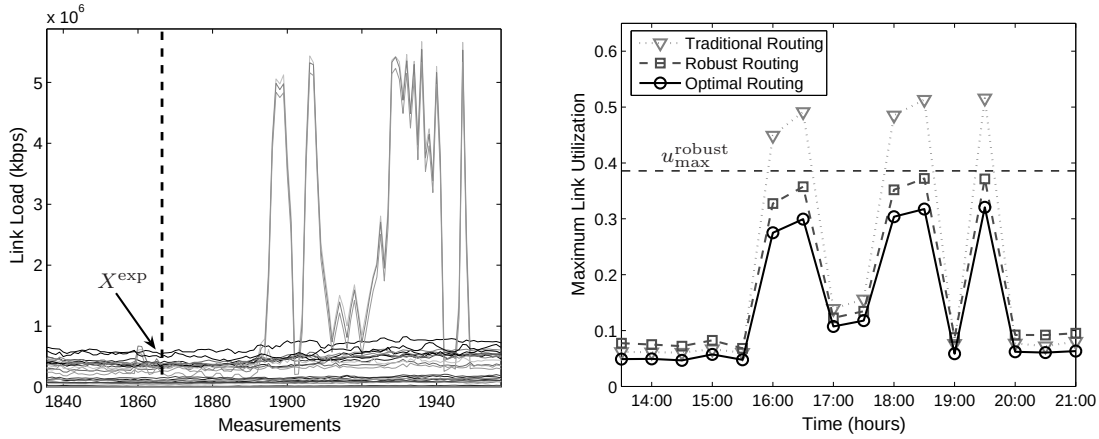
Figure 4.4 — Robust Routing with Time Varying Traffic in Normal Operation.

traffic measured during the busy-hour period. Finally, the dotted line corresponds to a fixed busy-hour-based routing, where routing configuration is optimized for X_{220} and applied during the 24hs period. Although the Robust Routing improves over the busy-hour-based routing, the benefits are only marginal.

Let us now analyze a similar scenario, but considering an unexpected abrupt traffic modification due to an external BGP routing modification. Figure 4.5.(a) depicts this situation. The network is highly under-loaded before the volume anomaly, and link utilization is below 10% for every network link. This is in fact the usual operation scenario for this temporal window of the Abilene network, which corresponds to traffic on Sunday. The Abilene network started as a research network, and at the time of this dataset the link utilization during the weekends was usually the same as the one depicted in here. In this situation, the expected TM $X^{\exp}$ is a reasonable candidate for a traditional prediction-based routing approach, where routing is optimized for this single TM; $X^{\exp}$ corresponds to the TM with index 1867 in figure 4.5.(a).

Figure 4.5.(b) presents the performance evaluation of the three routing optimization approaches previously evaluated. The full line corresponds to an optimal routing configuration, re-optimized for each single TM. The dotted line corresponds to a fixed prediction-based routing, where routing configuration is optimized for X_{1867} and applied during the complete evaluation period. The dashed line corresponds to a fixed robust routing configuration; in this case, we assume that the volume anomaly is known in advance, and build a polytope that includes this anomalous traffic. The polytope is built using an upper-bound $Y^{\mathrm{UB}} = Y^{\max(14:00-21:00)}$, which corresponds to the maximal traffic load values over the relevant time period.

Before the abrupt change, traffic remains almost constant and the MLU is similar for the three approaches. However, performance degradation for the traditional



(a) (b)

Figure 4.5 — (a) Daily traffic link load, (b) Routing evaluation.

approach reaches approximately 60% after the arrival of the unexpected event, against a 14% degradation for robust routing, both with respect to the ideal approach. The traffic demands that are responsible for these abrupt traffic modifications belong to the uncertainty set $\mathbb{X}(Y^{\max(14:00-21:00)})$, and thus the robust routing configuration is prepared to correctly handle them without surprises. In fact, the worst-case performance threshold obtained for polytope $\mathbb{X}(Y^{\max(14:00-21:00)})$ is $u_{\max}^{\text{robust}} = 0.385$, which represents a *potential* performance degradation of 19% with respect to the worst-case performance of the optimal routing approach in the evaluation period; we say that this is a potential performance degradation because no TM reaches this value in the evaluation of the robust routing. The good thing is that we can be sure that none of the TMs in $\mathbb{X}(Y^{\max(14:00-21:00)})$ can perform worse than this value of MLU.

Previous analyses show that the Stable Robust Routing approach offers performance guarantees against traffic uncertainty and unexpected traffic variations at a reasonable cost. However, considering a single routing configuration for long periods of time is clearly not a cost-effective solution and generally results in sub-optimal performance. SRR may present rather poor performance either when faced with traffic demands that do not belong to the polytope (tight uncertainty sets) or when it is designed to manage as many traffic demands as possible (big uncertainty sets). Using a single polytope to manage all traffic variations rises a difficult to address question: how do we specify the most accurate polytope? This polytope has to be sufficiently "large" to allow traffic flexibility, but should not be excessively "large" to avoid wasting network resources.

In the following section we present a very simple time-varying extension of the Stable Robust Routing approach that provides better performance. The idea is based on the notions of Multi-Hour Network Design [94]. We preserve the virtues of Robust Routing, but modify the routing configuration during time. The uncertainty set is optimally divided into several uncertainty sub-sets that better adapt to current traffic

demands, and a SRR configuration is computed for each sub-set. The partitioning algorithm allows to calculate the exact times when routing changes must be performed, simplifying network operation.

4.3 Multi-Hour Robust Routing

The problem of multi-busy-hour routing design, or simply multi-hour routing has been addressed by several researchers over the past years [15, 16, 17, 18, 93, 94, 95], especially for circuit-switched networks. The idea behind multi-hour models is to take advantage of non-coincidence of busy-hour loads between different OD flows in the network. Multi-busy-hour behavior in traffic flows is typically observed in networks that span over multiple time-zones. Multi-hour routing is a time-dependant routing method [95] in which routing is altered at a fixed point in time during the day or week. Routing configurations are determined on an off-line, preplanned basis and are implemented consistently over contiguous routing intervals.

The specific problem that we address in this section is how to build a multi-hour routing configuration that exploits multi-busy-hour traffic behaviors, maintaining the virtues of robust routing. The answer to this question seems to be quite simple, just consider a different definition of the polytope for different consecutive routing intervals of the day and compute a Stable Robust Routing configuration for each of them. The problem with this approach is how to determine the optimal moments when a new polytope has to be defined.

Based on rough knowledge of traffic variations, we propose a simple algorithm to adapt the polytope along time. We define the notion of *temporal uncertainty set*, based on predicted traffic variations extracted from past measurements. This temporal uncertainty set is optimally divided in the direction of time, and a Multi-Hour Robust routing configuration is built, considering a single SRR configuration for each sub-set. Similar to [16, 94] we assume a fully dynamically reconfigurable virtual-paths network, in which end-to-end paths can be established on-demand. We discuss the implications of such a choice in the concluding remarks of this section.

Daily traffic variations can be seen as a temporal variation of the uncertainty set. At each time slot t_j, the routing matrix R and the link load values Y_{t_j} define an uncertainty set $\mathbb{X}\left(Y_{t_j}\right) = \left\{X \in \mathbb{R}^m, R\,X \leqslant Y_{t_j},\ X \geqslant 0\right\}$. The slotted time comes from the fact that SNMP measurements Y_{t_j} are collected at discrete time intervals t_j. However, the same definition can be applied in a continuous time situation. Figures 4.6(a) and 4.6(b) explain this idea of discrete and continuous temporal variation of the uncertainty set. The union of several uncertainty sets along contiguous time slots $t_1, .., t_j, .., t_\tau$ defines the concept of *temporal uncertainty set*:

$$\mathbb{X}_t = \left\{X^{'} = (X_{t_j}, t_j) \in \mathbb{R}^{m+1},\ X_{t_j} \in \mathbb{X}\left(Y_{t_j}\right),\ t_j \in [t_1..t_\tau]\right\}$$

Assuming that this set is a union of polytopes, [109] provides a theoretical study of the optimal partitioning of $\mathbb{X}_t$, using a partitioning hyper plane. In particular, it proves that this is a NP-hard problem, except for the case where a partitioning direction is previously fixed. In such a case, the author presents a simple algorithm to approximately solve the optimal partitioning problem in polynomial-time, using a generalization of a simple dichotomy methodology.

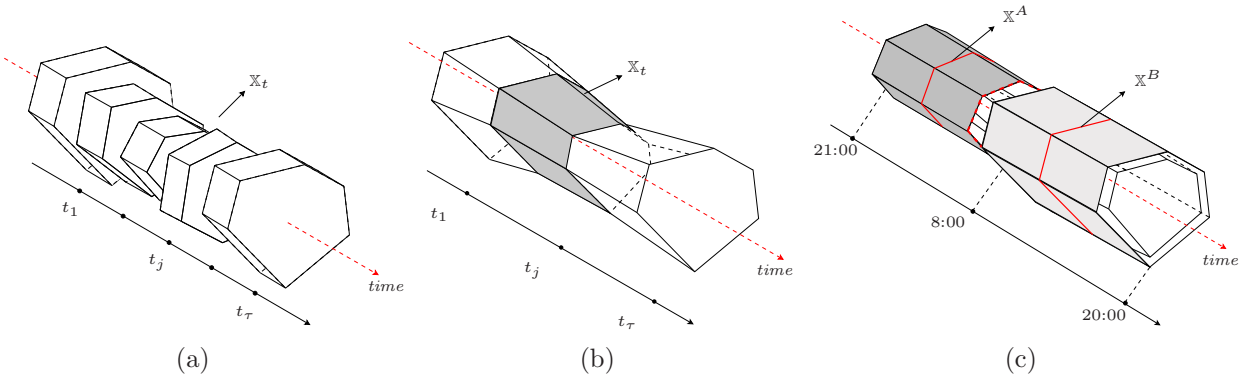


Figure 4.6 — Daily variation of the polytope $\mathbb{X}_t$, (a) discrete-time $\mathbb{X}_t$, and (b) continuous time $\mathbb{X}_t$. (c) Time partitioning of $\mathbb{X}_t$.

A partitioning hyper plane is defined by its direction vector α and a value β: $\alpha.X^{'} = \beta$. In the MHRR approach there is a particular direction for partitioning: the *time direction*. In this case, $\alpha = [0, .., 0, 1]$ and $\beta = t$. We define $h + 1$ hyperplanes at times $\{\beta_1, \beta_2, .., \beta_{h+1}\}$. Let $\mathbb{X}^i$ be the convex hull of the union $\left\{\cup_{\beta_i \leqslant t_j \leqslant \beta_{i+1}} \mathbb{X}\left(Y_{t_j}\right)\right\}$, $\forall i = 1, ., h$. The polytope $\mathbb{X}^i$ corresponds to single-time minimal convex set that contains all the traffic realizations between times β_i and β_{i+1}. Figure 4.6(c) depicts exemplifies the idea. The MHRR consists of computing the optimal times when routing should change, namely $\boldsymbol{\beta}^* = \{\beta_2^*, \ldots, \beta_h^*\}$, in order to minimize the worst case value of the MLU. Finally, a single SRR configuration is computed for each time interval $\left[\boldsymbol{\beta}_i^*, \boldsymbol{\beta}_{i+1}^*\right]$, $\forall i = 1, ., h$. The vector $\boldsymbol{\beta}^*$ is the solution to the following optimization problem:

$$\boldsymbol{\beta}^*(\mathbb{X}_t) = \arg\min_{\boldsymbol{\beta}} \left\{ \max_{i=1..h} \; u_{\max}^{\text{robust}}\left(\mathbb{X}^i\right) \right\} \tag{4.11}$$

where $u_{\max}^{\text{robust}}\left(\mathbb{X}^i\right)$ is the solution to (4.6) for polytope $\mathbb{X}^i$ and β_1, β_{h+1} are fixed a priori, as they define the considered time interval of analysis.

The interesting issue in our proposal is that we provide an objective means of computing an optimal multi-hour routing design, maintaining the robustness of the RR approach. The optimality property of our approach lies on the computation of the ideal times to switch routing. Traditional methods used in the design of multi-hour routing configurations are rather naive, relying on a couple of TMs to optimize different routing configurations [94].

The MHRR presents a trade-off between performance and routing stability. The more intervals, the more adapted the routing becomes. However, the number of intervals should be bounded as many routing changes may lead to instabilities and performance degradation. In a general case, 2 sub-sets are enough to handle the usual daily variation.

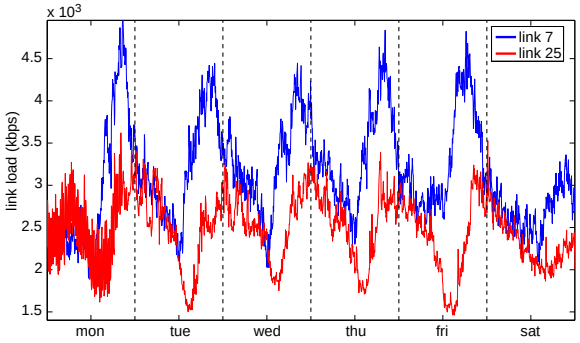


Figure 4.7 — Multi-busy-hour behavior in the traffic of two links in Abilene.

4.3.1 Multi-Hour Robust Routing Evaluation

The SRR and the MHRR approaches are compared in Abilene. The Abilene network spans over different time zones in the US, so a priori it could be possible to find multi-busy-hour behaviors in its OD-flows traffic. However, there is no such a behavior among the largest volume OD-flows, and there is a slight temporal shift between some low-volume OD-flows. This was already clear from chapter 2, where the Spline-Based model was validated with traffic from the same dataset. Figure 4.7 shows the traffic load in two links of Abilene with different geographical location. The traffic in these links corresponds to very low-volume OD-flows, and thus the link utilization is almost null. The Abilene dataset that we used in our studies [129] mainly corresponds to experimental traffic, which explains the lack of important traffic shifts among flows.

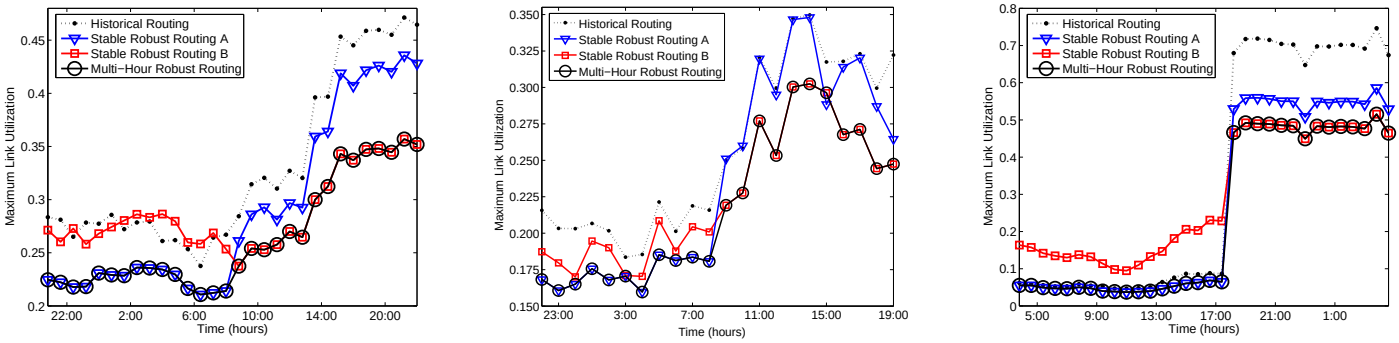
In order to evaluate the performance of the Multi-Hour approach, we scale these small OD-flows that present multi-busy-hour behavior by an important magnifying factor. In this scenario, the time variation of the polytope is not a simple homothety, and a multi-hour routing scheme applied during the day can be used to improve routing performance.

As we did before, let R_o be the historical routing matrix used in Abilene, available at [129]. We consider a single time partitioning representing two routing intervals, $\beta_1 = 21{:}00$, $\beta_2 = \beta^*$ and $\beta_3 = 20{:}00$, where β^* is the solution to (4.11). The smallest polytope that includes all possible traffic realizations over that period is computed for each time interval:

$$\mathbb{X}^{\,\mathrm{LTL}} = \{X \in \mathbb{R}^m, R_o.X \leqslant Y^{\mathrm{LTL}},\ X \geqslant 0\}$$

$$\mathbb{X}^{\,\mathrm{HTL}} = \{X \in \mathbb{R}^m, R_o.X \leqslant Y^{\mathrm{HTL}},\ X \geqslant 0\}$$

where $Y^{\mathrm{LTL}} = Y^{\max(14:00-\beta^*)}$ and $Y^{\mathrm{HTL}} = Y^{\max(\beta^*-20:00)}$, i.e., the maximum traffic load values for each link, during the Light Traffic Load (LTL) and the High Traffic Load (HTL) periods respectively. Figure 4.6.(c) depicts both polytopes. $\mathbb{X}^{\,\mathrm{LTL}}$ includes



(a) Expected daily behavior (I). (b) Expected daily behavior (II). (c) Anomalous unexpected event.

Figure 4.8 — Routing performance, Stable vs. Multi-Hour Robust Routing.

all traffic demands between 21:00 and β^*, and $\mathbb{X}^{\text{HTL}}$ between β^* and 20:00. For each polytope, a SRR configuration is computed from (4.6), which will be referenced as R^{LTL}_{robust} and R^{HTL}_{robust}. In order to compare the stable and the multi-hour approaches, both routing configurations are applied during the whole evaluation period. The routing performance obtained with R_{o} is also included, represented by the dotted line with label *Historical Routing* in figure 4.8.

Figure 4.8(a) compares the routing performance between these two RR configurations, regarding MLU. Polytope $\mathbb{X}^{\text{LTL}}$ is better suited for smaller loads, so R^{LTL}_{robust} performs better during the first half of the day, when network load is lower. However, when traffic increases, demands that do not belong to $\mathbb{X}^{\text{LTL}}$ produce higher link utilization values than those obtained with R^{HTL}_{robust}. The MHRR consists of computing the optimal times when routing must be changed, using the corresponding routing configuration depending on the time of the day. In this evaluation, the obtained value of β^* is around 8:00. Assuming a fully dynamically reconfigurable network scenario, the MHRR configuration corresponds to R^{LTL}_{robust} before β^* and R^{HTL}_{robust} after. In this evaluation, the MHRR approach presents a performance improvement of about 16% with respect to the SRR approach before β^*, reaching a near 20% of over-efficiency after β^*. Figure 4.8(b) presents a similar evaluation scenario, but in this case the performance improvements are less evident. The optimal partition value is $\beta^* \approx$ 9:00, and the performance improvement rounds a 10%.

As a final evaluation, we consider a scenario where traffic demands experience an important volume anomaly. Figure 4.8(c) presents an abrupt change in volume at time 18:00, resulting in a value of MLU almost 14 times higher. In this case, we assume that this change is known in advance; note that in the general case, it is not possible to predict these abrupt changes, and the application of the MHRR is questionable. It is not surprising that the optimal time for changing routing obtained with the partitioning algorithm is $\beta^* \approx$ 18:00. The MHRR approach definitely outperforms the SRR in this experience, presenting a MLU between 10% and 60% smaller during the whole evaluation period.

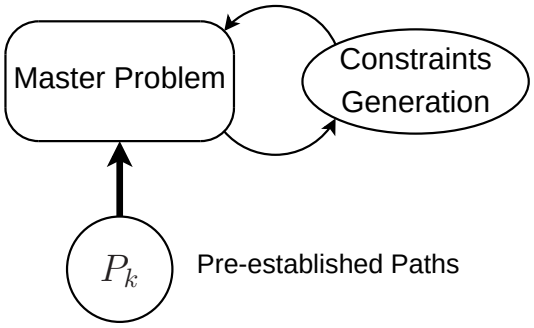


Figure 4.9 — Robust Routing Optimization Problems with Pre-established Paths P_k.

The Multi-Hour Robust Routing method that we propose in this chapter consists in routing reconfiguration, which means that both the set of established paths P_k and the traffic distribution fractions among these paths r_p^k may vary from one routing period to another. Changing the set of paths under the assumption of fully dynamically reconfigurable end-to-end paths is straightforward from a theoretical point of view. However, modifying the entire routing configuration of a large-scale network is in practice a challenging task. This problem can be easily alleviated using load-balancing instead of a full routing reconfiguration. In load-balancing, paths remain fixed and the only modification is related to the fraction of traffic sent through each of these paths.

Extending the MHRR approach to the case of load-balancing is straightforward: the set of paths $P_k, \forall k = 1 \dots m$ obtained in the first Stable Robust Routing optimization from problem (4.6) remains the same in the following SRR optimizations. In order to use a pre-established set of paths in RROP, the algorithm depicted in figure 4.2 is slightly modified. Solving RROP for a given set of pre-established paths P_k consists of only adding new extreme points of polytope $\mathbb{X}$ in (4.6), thus using only the constraints generation problem as depicted in figure 4.9.

The path diversity obtained in previous evaluations was rich enough and path modifications between consecutive routing optimizations were usually rare. For this reason we believe that results will not significantly vary from those presented in the case of using load balancing in MHRR. The evaluations of MHRR confirm once again that using a single Robust Routing configuration is not a cost-effective solution when traffic is relatively dynamic. It is clear from our study that some form of dynamism is necessary; the MHRR represents a step in this direction.

4.4 Reactive Robust Routing

The Multi-Hour Robust Routing approach presented in previous section offers a robust and efficient routing method, given a rough knowledge of the daily uncertainty set. In this sense, the MHRR approach can be used as a *proactive* technique to handle dynamic traffic. However, in the presence of volume anomalies it is no longer possible to apply the method, simply because the daily uncertainty set is completely unknown. Despite being one of its most important features, we have shown that relying on a single Stable Robust Routing configuration to address both usual traffic behavior and volume anomalies would be an inefficient solution. In the final remarks of [103], authors claim that it is not clear whether highly dynamic traffic should be addressed either using proactive or reactive methods. From our point of view, a *reactive* approach could be used as a complementary strategy to enhance Robust Routing, responding to abrupt and large traffic changes with an effective routing reconfiguration.

In this section we present a unified routing solution to handle both traffic behaviors, known as the Reactive Robust Routing (RRR) approach. The idea behind this solution is to combine both proactive and reactive complementary approaches to deal with dynamic traffic demands, separately treating expected traffic fluctuations and unpredictable traffic behaviors. Figure 4.10 presents a high-level description of the Reactive Robust Routing method.

The RRR approach uses the MHRR to handle expected variations in traffic demands, and the Sequential Spline-Based (SSB) volume anomaly detection/isolation algorithm presented in chapter 3 to deal with unexpected volume anomalies. The method exploits the isolation ability of the SSB algorithm to accurately adapt routing after the anomalous traffic detection, reducing its impact on network performance during its prevalence. In addition, it also provides a simple yet effective method to automatically detect the end of the anomaly, taking up again the MHRR configuration. A key feature of the RRR approach relies on the fact that the whole routing configuration and reconfiguration algorithm is completely automatic, an interesting property that simplifies network operation.

4.4.1 The Reactive Robust Routing Algorithm

We shall begin by introducing the reactive component of the Reactive Robust Routing approach. The idea of this reactive component is to continuously monitor traffic behavior in search for large volume modifications that may render current routing configuration unsuitable, or even unfeasible. If traffic anomalies can be rapidly detected and accurately isolated, it is possible then to adapt the routing configuration as a countermeasure. Let us explain this idea of *anomaly-adapted* routing reconfiguration. We assume that a Stable Robust Routing configuration $R^{\circ}_{\text{robust}}$ is applied under normal operation conditions, where traffic varies within a normal operation polytope $\mathbb{X}_{\text{o}}$. This polytope is defined in the same way as in section 4.2, based on a certain historical routing configuration R_{o} and the expected links traffic load we shall call Y_{o}.

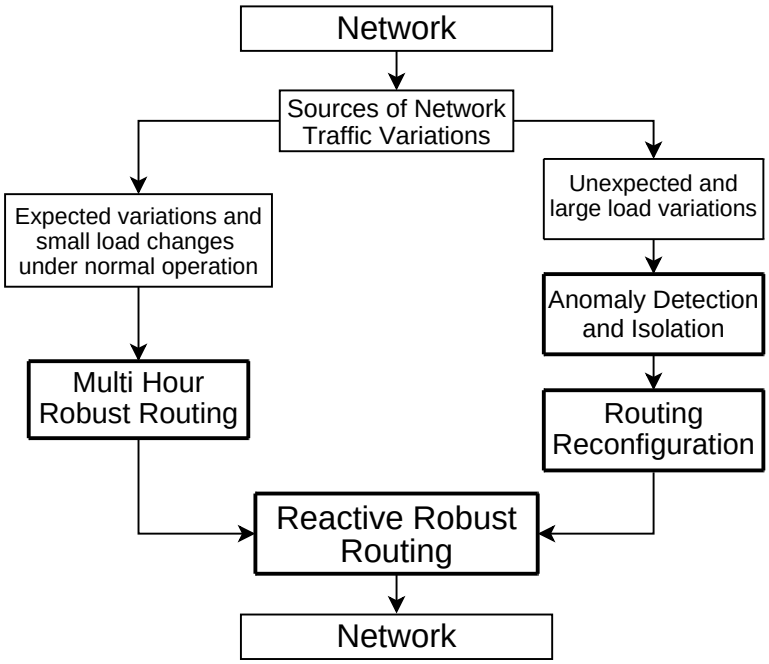


Figure 4.10 — High-level description of the Reactive Robust Routing

Suppose now that an anomalous traffic augmentation of size θ occurs at a certain unknown time t_o. As we did in chapter 3, we will assume that the anomaly occurs in a single OD flow k. This is to ensure that the anomaly can be correctly isolated with the SSB algorithm. The normal operation traffic demand $X \in \mathbb{X}_o$ takes the value $X_{\text{anomaly}} = X + \boldsymbol{\theta}$, where $\boldsymbol{\theta} = \theta.\boldsymbol{\delta}_k$ and $\boldsymbol{\delta}_k = (\delta_{1,k}, \ldots, \delta_{k,k}, \ldots, \delta_{m,k})^T$, $\delta_{i,k} = 0$ if $i \neq k$ and $\delta_{k,k} = 1$. If the size of the anomaly θ is such that $X_{\text{anomaly}} \notin \mathbb{X}_o$, we are in a situation in which traffic has left the uncertainty set and the performance obtained with R^o_{robust} can not be assured. But what if the uncertainty set $\mathbb{X}_o$ were big enough so as to include this unpredictable anomalous traffic variation before it actually happens? This was exactly the case in the scenarios depicted in figures 4.5.(b) and 4.8(c), where traffic anomalies were supposed to be known in advance, and an accurate polytope was built for them.

The solution proposed in RRR is to dynamically adapt the normal operation polytope $\mathbb{X}_o$ so as to completely cover the volume anomaly. This adaptation consists in *expanding* $\mathbb{X}_o$ in the directions of the links crossed by the anomalous OD flow k, obtaining an *anomaly-polytope* $\mathbb{X}_k = \{X \in \mathbb{R}^m, R_o X \leqslant Y_o + R_o\,\boldsymbol{\theta},\ X \geqslant 0\}$. Note that the expansion that we propose is with respect to R_o, which is in fact the routing matrix used in the definition of the normal operation polytope $\mathbb{X}_o$, and not with respect to R^o_{robust}. The reader should bear in mind that the kind of volume anomalies that we deal with originate outside the network, which justifies the relevance of the polytope expansion with respect to R_o and not with respect to the routing configuration that was running when the anomaly arrived. In other words, the obtained polytope $\mathbb{X}_k$ is the smallest polytope that contains X_{anomaly} which could have been built using the normal operation data $\{R_o, Y_o\}$ and the unknown anomaly

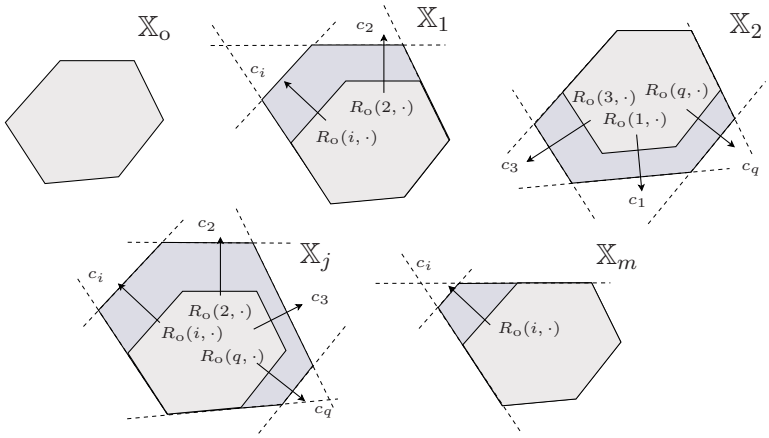


Figure 4.11 — Different anomaly polytopes for preemptive robust routing computation.

size θ. Thus, the corresponding robust routing configuration R^k_{robust} associated with the anomaly-polytope $\mathbb{X}_k$ would certainly provide a better performance than $R^{\text{o}}_{\text{robust}}$ in the event of the anomaly. Once the SSB algorithm has detected and isolated a volume anomaly in OD flow k, the RRR has to deployed the adapted routing configuration R^k_{robust}. Remember that the SSB algorithm has a key property that improves the latency of the routing reconfiguration: it minimizes the number of measurements which are necessary to detect and isolate the anomaly. Thus, RRR can adapt the routing configuration extremely fast when faced to large volume anomalies, contrary to traditional dynamic load balancing schemes. We treat this issue in depth in section 4.6.

The construction of the anomaly-polytope $\mathbb{X}_k$ implies the estimation of the size of the anomalous traffic θ. Moreover, this estimation has to be performed on-line and simultaneously to the detection and isolation of the anomaly, because it can not be forecast. Eventhough this could be partially achieved using similar techniques to those applied in [72, 80], we do not intend to construct $\mathbb{X}_k$ in real-time. In fact, building $\mathbb{X}_k$ and computing the associated SRR configuration R^k_{robust} in real-time has an undeniable cost in terms of computational time that may even render the RRR inapplicable, mainly due to the time-consuming computation of R^k_{robust}. Besides, the estimation of θ has always an associated error. The idea in RRR is then to pre-compute a set of m anomaly-polytopes $\mathbb{X}_k$, $\forall k = 1 \dots m$ in an off-line basis, considering the m single OD flow anomalies that may arise. Given that θ is completely unknown, the primal polytope $\mathbb{X}_o$ is expanded to the limits of link capacities, obtaining the following anomaly-polytope for each single OD flow anomaly:

$$\mathbb{X}_k = \{X \in \mathbb{R}^m, R_o.X \leqslant Y_k,\ X \geqslant 0\}\,,\ \forall k \in N \tag{4.12}$$

In 4.12, the i-th component $Y_k(i)$ takes the value $Y_o(i)$ if $R_o(i,k) = 0$, or the value c_i if $R_o(i,k) > 0$, being $R_o(i,k)$ the element (i,k) of R_o. Such an expansion represents a worst-case dimensioning and may result in routing inefficiency w.r.t. an expansion of size θ, but it certainly provides a more robust routing scheme than simply guessing θ. Figure 4.11 explains the idea of the multiple anomaly-polytope expansion.

4.4.2 Back to Normal Operation

As we said before, the Reactive Robust Routing algorithm consists not only in the robust routing reconfiguration after the isolation of a volume anomaly, but also it provides a way to detect the end of the anomaly so as to regain the normal operation routing. This detection can be easily achieved by using a similar detection algorithm as the one used in the OSBD detection method.

Let us suppose that an anomaly in OD flow k has been detected and isolated at time t_k. At this time, a robust routing reconfiguration is performed and the robust routing matrix R^k_{robust} is deployed. For every time $t > t_k$, the algorithm analyses the distribution of traffic residuals $f_k(U_t)$ and looks for a rupture change that indicates the end of this anomaly. Remember that $f_k(U_t)$ corresponds to the pdf of residuals U_t after a volume anomaly in OD flow k.

As we have only considered anomalies in a single OD flow at a time, the only change that we can expect to find in U_t is a return to normal operation. Then, two simple hypotheses are considered for this detection problem: the null hypothesis $\mathcal{H}_k = \{U_t \sim \mathcal{N}(\theta\,\mathbf{v}_k, \gamma_t^2\, I_{r-q})\}$ where the k-th OD flow presents an anomalous additional amount of traffic, and the alternative hypothesis $\mathcal{H}_0 = \{U_t \sim \mathcal{N}(0, \gamma_t^2\, I_{r-q})\}$ where OD flow k is anomaly-free. A simple Neyman-Pearson test [86] is applied at each time t to decide between both hypotheses. The Neyman-Pearson test represents the most powerful test for two simple hypotheses [86]. The statistics of this test is given by:

$$\Lambda_k(U_t) \;=\; \log \frac{f_0(U_t)}{f_k(U_t)} - h \tag{4.13}$$

where the decision level h is defined according to the tolerated false alarm probability. The reader should remember that f_0 represents the pdf of residuals under anomaly-free behavior, i.e hypothesis $\mathcal{H}_0$, while f_k is the pdf of U_t under hypothesis $\mathcal{H}_k$. If $\Lambda_k(U_t) < 0$, the decision test chooses hypothesis $\mathcal{H}_k$. When $\Lambda_k(U_t) > 0$, the test decides hypothesis $\mathcal{H}_0$, pointing out the end of the anomaly. At this time, the normal operation robust routing configuration is deployed and the SSB algorithm is turned-on once again.

4.4.3 The Complete Reactive Robust Routing Algorithm

Figure 4.12 depicts a diagram of the complete Reactive Robust Routing algorithm. The flow diagram is divided in four major modules that interact together in a non-sequential order, established by the occurrence of volume anomalies. The first module corresponds to the calibration of the anomaly-free traffic model; module two controls the MHRR algorithm; module three represents the SSB anomaly detection/isolation algorithm and it is responsible for locating the anomaly and deploying the corresponding robust routing configuration. Finally, module four is in charge of detecting the end of the volume anomaly, giving back control to module two.

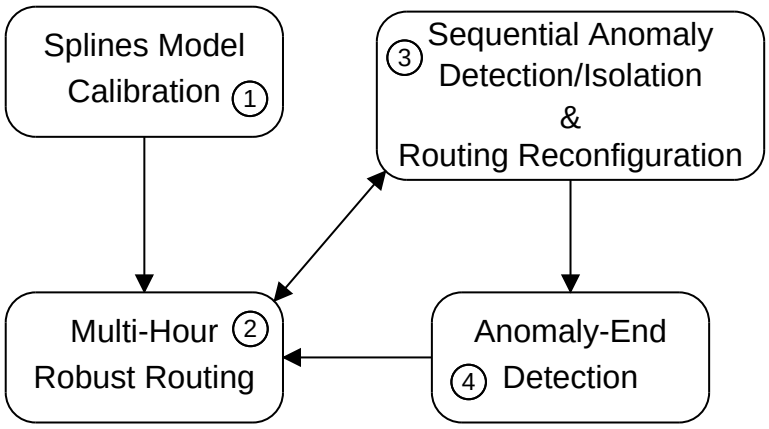


Figure 4.12 — The Complete Reactive Robust Routing Algorithm.

Figure 4.13 presents a detailed flow-diagram of the interaction among the four modules that compose the RRR algorithm. This flow-diagram depicts the daily operation of RRR, assuming that the MHRR algorithm has split the daily routing into two disjointed periods, before and after the partitioning time β^*. Similar to the evaluations presented in figure 4.8, we define a Light Traffic Load (LTL) period before β^* (i.e., for $t < \beta^*$) and a High Traffic Load (HTL) period for $t > \beta^*$. We shall use R^{LTL}_{robust} and R^{HTL}_{robust} as a reference to the corresponding robust routing configurations in both periods.

The algorithm starts in module (1) with the calibration of the splines model, computing the splines basis S and the estimated covariance matrix $\widetilde{\Sigma}$ from the learning dataset. In this first stage we also set two flag variables that are used to coordinate the interaction among the rest of the modules. The `<anomalyFLAG>` variable takes value 1 if a volume anomaly has been detected and 0 otherwise. The `<mhrrFLAG>` variable takes value 0 if the MHRR algorithm has already switched routing configuration from R^{LTL}_{robust} to R^{HTL}_{robust} and 1 otherwise. As we are at the very beginning of the algorithm, we shall set `anomalyFLAG = 0` and `mhrrFLAG = 1`. From now on we assume that the spatial distribution of anomaly-free OD flows remains stable in time, even after a routing modification. This simplifying assumption is quite strong, mainly because we are leaving aside the hot-potato routing effect. Hot-potato routing may induce interdomain routing changes due to intradomain routing modifications, thus modifying the spatial distribution of OD flows. However, given the short calibration period of the splines model, it would be possible to recalibrate the complete model after a routing modification, and thus we believe that this assumption is somewhat reasonable.

The first thing that module (2) verifies is whether R^{LTL}_{robust} or R^{HTL}_{robust} has to be deployed, depending on the time of the day, i.e. before or after β^*. Once the accurate routing configuration has been deployed, the matrices $G = RS$, $\Phi = R\widetilde{\Sigma}R^T$, $H = \Phi^{\frac{1}{2}}G$, and W are (re)computed, and the SSB algorithm is (re)started in module (3).

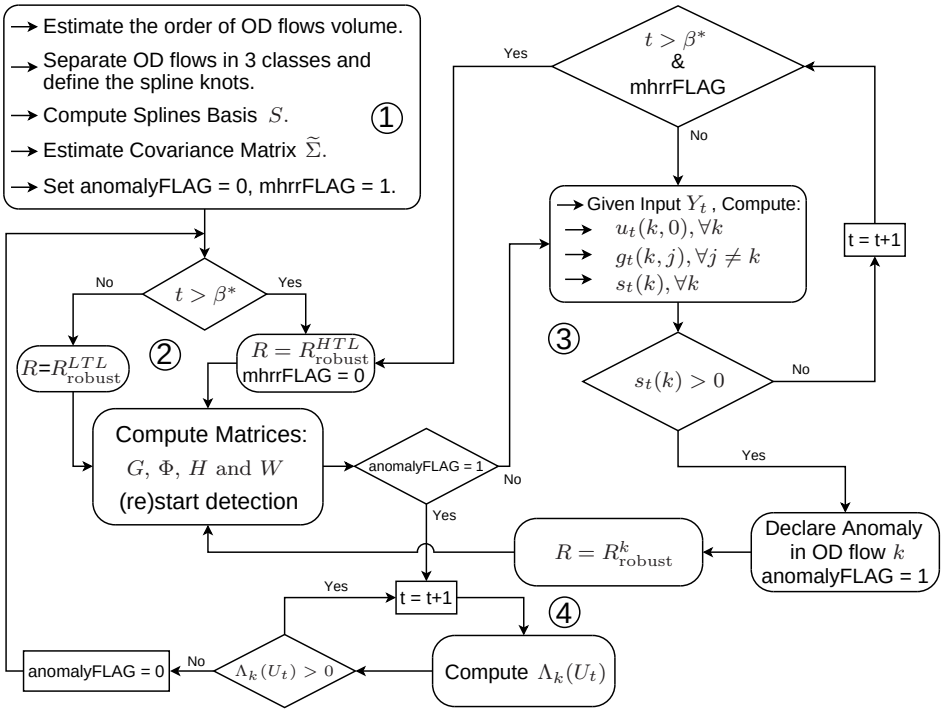


Figure 4.13 — Flow Diagram of the Complete Reactive Robust Routing Algorithm.

With each new SNMP measurement Y_t, module (3) updates the recursive detection and isolation functions. If no anomalies were detected, the algorithm verifies if the MHRR configuration has to be updated or not. If current time $t < \beta^*$, the algorithm keeps running the SSB anomaly detection/isolation method. On the contrary, if $t > \beta^*$, the algorithm goes back to module (2), deploys the corresponding SRR configuration R^{HTL}_{robust}, sets the flag variable `mhrrFLAG = 0`, updates matrices G, Φ, H, and W, and finally restarts the SSB algorithm.

When a volume anomaly is detected and isolated in OD flow k, the flag variable `anomalyFLAG` is set to 1 and the corresponding SRR configuration R^{k}_{robust} is deployed. The matrices G, Φ, H, and W are recomputed, and module (4) comes on stage, using the detection algorithm defined in 4.13 to detect the end of the anomaly. Module (4) keeps running until the volume anomaly ends, in which case `anomalyFLAG` is set to 0 and control goes back to module (2), restarting the complete RRR algorithm once again.

4.4.4 Partial Robust Routing Reconfiguration and Reactive Robust Load Balancing

RRR can handle large and unexpected traffic variations in single OD flows; the case of multiple simultaneous anomalies is beyond the scope of RRR. However, given the difficulty involved in modifying the routing configuration of a large scale network in an on-line fashion, the contributions of RRR are mainly theoretical. This routing reconfiguration problem can be alleviated with two different proposals: partial re-routing and load balancing.

$$
\begin{aligned}
& \text{minimize} && u_{\max} && && (4.14)\\
& \text{subject to:} && && &&\\
\rho(X,l) + \sum_{p \in P_k} \lambda_l^p \,.\, r_p^k \,.\, x(k) & \leqslant && u_{\max} \,.\, c_l && \forall\, l \in L,\ \forall\, X \in \mathbb{X}\\
\rho(X,l) & = && \sum_{\substack{j \in N p \in P_j \\ j \neq k}} \sum \lambda_l^p \,.\, r_p^j \,.\, x(j) &&\\
\sum_{p \in P_j} r_p^j & = && 1 && j = k\\
u_{\max} & \leqslant && 1 &&\\
r_p^k & \geqslant && 0 && \forall\, p \in P_k
\end{aligned}
$$

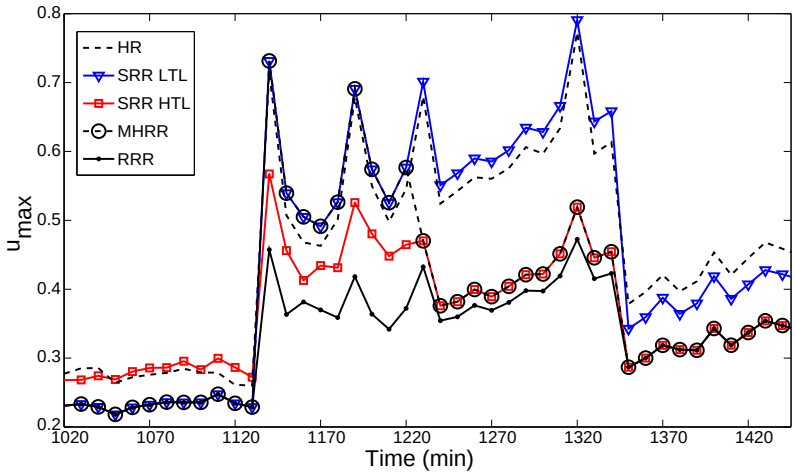
Partial re-routing consists in computing a routing configuration R^k_{robust} which may only add new end-to-end paths for routing the anomalous OD flow k, keeping unchanged the already-established set of paths for the rest of the OD flows. This highly reduces the number of additional paths from one routing configuration to the other, reducing the complexity of the reconfiguration. The Partial Robust Routing Reconfiguration problem is defined in (4.14). The set of paths P_j and the routing fractions r_p^j, $\forall j \neq k$ remain exactly the same in partial re-routing, and the only modifications occur in P_k and r_p^k.

As we have previously explained in section 4.3, load balancing consists in keeping unchanged the set of paths P_j for each OD pair j, only modifying the routing fractions on each path. Load balancing can be easily performed on-line and does not require any additional modifications in current path-based networks such as MPLS, thus we will adopt this solution in the RRR algorithm. We shall refer to the load balancing variant of RRR as the Reactive Robust Load Balancing (RRLB), stressing the difference between routing reconfiguration and load balancing.

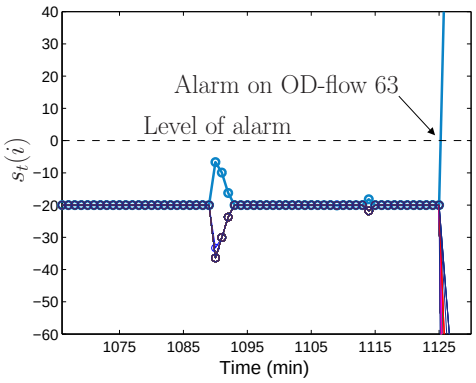
RRLB uses the same set of anomaly-polytopes $\mathbb{X}_j$ defined in RRR, but the computation of the m corresponding SRR configurations R^j_{robust} is slightly modified. The same set of paths P_j obtained during the computation of R^o_{robust} is used in every R^j_{robust}. Each R^j_{robust} is obtained with the simplified version of the former robust routing optimization algorithm presented in section 4.3, where only new traffic demands are progressively added and no extra paths are created.

4.4.5 Reactive Robust Routing Evaluation

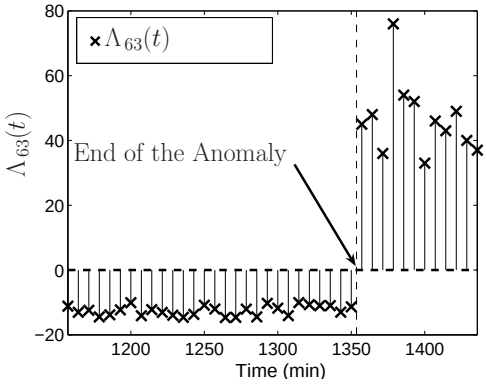
To conclude this with section, we present some evaluations of the RRR and the RRLB algorithms using the Abilene dataset. In these evaluations we shall introduce an artificial abrupt and large volume modification on top of normal operation traffic in one particular OD flow, so as to be sure that it can be correctly isolated by the SSB algorithm.



(a) Reactive Robust Routing.



(b) Anomaly detected in OD-flow 63.

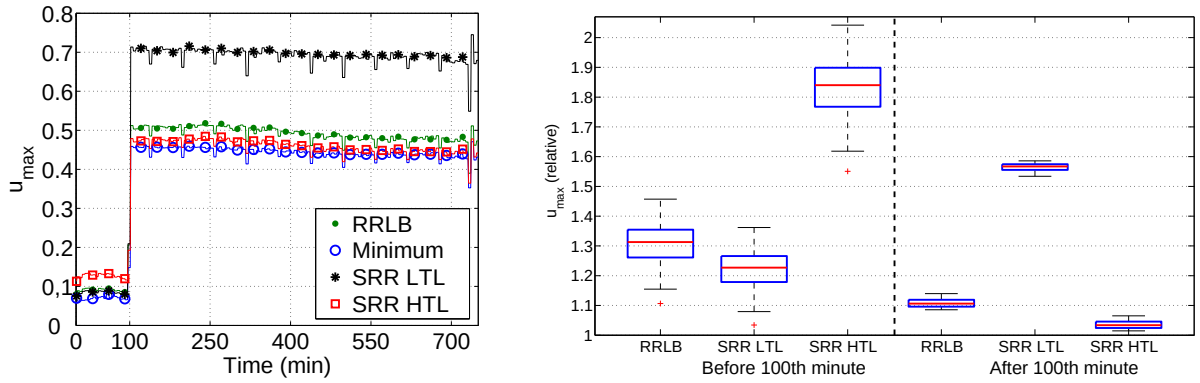


(c) Anomaly-end detected.

Figure 4.14 — Reactive Robust Routing performance under a simulated single-flow volume anomaly.

Figure 4.14 depicts an artificial large volume change in OD-flow 63 of about 4hs long, put on top of the usual daily traffic between times 1125 min and 1350 min. Figure 4.14(a) presents the evaluation of the different routing algorithms so far discussed in this chapter: the dashed curve with label 'HR' corresponds to the Historical Routing configuration R_{o} used in Abilene, which is not necessarily optimized for any particular traffic. The triangle-spotted and square-spotted curves with labels 'SRR LTL' and 'SRR HTL' correspond to the Stable Robust Routing configurations built for light and high load normal operation traffic respectively. The circle-spotted curve corresponds to the MHRR configuration, built from an expected daily uncertainty set under normal operation traffic. The optimal routing division obtained from (4.11) results in $\beta^* = 1230$ min. Finally, the full-line with label 'RRR' depicts the performance of the Reactive Robust Routing algorithm.

The evaluation begins at time 1020, when the MHRR module decides to apply the SRR configuration R^{LTL}_{robust}. The detection/isolation algorithm continuously monitor



(a) RRLB under single-flow volume anomaly.

(b) Comparative evaluation.

Figure 4.15 — Reactive Robust Load Balancing - load balancing after detection and isolation of a large volume anomaly in OD flow 13.

OD flows traffic from SNMP measurements in figure 4.14(b), and at time $t_{63} = 1125$ detects and isolates an anomalous behavior in OD flow 63. After the detection and before the new sampling of SNMP measurements $Y_{t_{63}+1}$, i.e. a 5 min period in Abilene, the new routing configuration R^{63}_{robust} is deployed and the anomaly-end detection phase begins in figure 4.14(c). The decision test $\Lambda_{63}(U_t)$ remains negative for every time $t > t_{63}$, until time $t^{'} = 1350$, when the positive value of $\Lambda_{63}(U_{t^{'}})$ points out the end of the anomalous behavior in OD flow 63. At this time, the MHRR module compares t' with β^* and decides to deploy the SRR configuration R^{HTL}_{robust}. Once the new routing configuration is established, the anomaly detection/isolation algorithm is re-started again.

The performance improvements of RRR w.r.t. the rest of the robust routing algorithms are evident, up to a 40% w.r.t. MHRR and about 50% w.r.t. the traditional SRR approach. The set of paths obtained from the RROP optimization (4.6) for both SRR configurations R^{LTL}_{robust} and R^{HTL}_{robust} turned out to be the same, even though this was not a priori wanted. As regards R^{63}_{robust}, only 3 additional end-to-end paths were established.

In the second evaluation, we study the performance of the load balancing extension of RRR, namely the RRLB algorithm. In this case we introduce a large volume anomaly in OD flow 13 after the 100-th minute. Figure 4.15(a) compares the performance of four different routing approaches: the SRR LTL and SRR HTL routing configurations correspond to a polytope perfectly adapted to traffic flows before and after the 100-th minute; the RRLB corresponds to a normal operation polytope built for normal operation traffic before the 100-th minute, and expanded after the event of the anomaly. Finally, an optimal routing configuration is computed for every TM at every time t, using (4.1). These routing configurations use exactly the same set of paths, and all that varies among them are the routing proportions r^k_p.

Figure 4.15(b) shows a boxplot summary on the performance of these routing configurations w.r.t. the optimal routing, before and after the volume anomaly. The difference between SRR LTL and SRR HTL before and after the 100-th minute is quite impressive, more than 50% of performance loss w.r.t. each other, and between 55% and 85% w.r.t. an ideal routing. This shows once again the major drawback of Stable Robust Routing as regards using a single routing configuration. RRLB performs worse than SRR LTL before the 100-th minute and worse than SRR HTL after the 100-th minute, but its performance is not worse than 30% w.r.t. the ideal routing, even after the occurrence of the volume anomaly. Given that the RRLB algorithm uses the same set of paths before and after the anomaly, it can be easily applied without concerning about the practical implementations issued in RRR. As we will show in section 4.6, RRLB also has a key *stability* advantage w.r.t. traditional Dynamic Load Balancing (DLB) mechanisms when faced to large volume anomalies. Briefly speaking, routing fractions in RRLB are fixed a-priori for every possible single-flow anomalous behavior, whereas in DLB these fractions are dynamically adjusted, inducing undesired fluctuations. We further discuss this issue in section 4.6.

4.5 QoS in Robust Routing: Improving Network-Wide Performance

So far in this chapter we have addressed the cost-efficiency problem of Stable Robust Routing, associated to the use of a single routing configuration and to the definition of the uncertainty set. The second problem that we identify in current Robust Routing techniques is related to the objective function it intends to minimize. Robust optimization is generally more complex than classical optimization, which forces using simpler optimization criteria such as the one we have used so far. Maximum Link Utilization $u_{\max}$ is the most popular Traffic Engineering objective function, but it simply represents a local performance indicator, becoming quite unsuitable for network-wide routing optimization. A routing configuration minimizing $u_{\max}$ may often lead to a worse distribution of traffic, adversely affecting the global performance of the network. Besides, while it is true that overloaded links tend to cause QoS degradation (e.g., larger delays and packet losses, throughput reduction, etc.), $u_{\max}$ does not represent a direct QoS indicator, a desirable property in the context of QoS provisioning.

From all the different QoS indicators generally used in networking, end-to-end delay is probably the most important of them. An end-to-end path with high loss-rates will surely experience large end-to-end delays due to retransmissions, but a zero-loss path may also experience long delays. An obvious extreme case is a path with infinite buffer at the bottleneck link. Conversely, a low latency path will generally present small loss-rates. In this sense, it is better to have a network optimized for low latencies than optimized for low loss-rates.

In order to evaluate Robust Routing from a network-wide QoS perspective, we shall consider the path end-to-end (e2e) delay. The e2e delay on a path is the sum of the delays on each link of the path. The delay on each link consists of two components, namely the queuing delay (i.e., buffer and service delay) and the link propagation delay. The former depends on the link load, while the latter is constant. In this sense and as a simplification to the problem, we shall consider the e2e path queuing delay as a measure of performance. Assume that queuing delay on link l is given by the function $d_l(y(l))$. Given this function, we can compute the e2e queuing delay of path p as $d_p = \sum_{l \in p} d_l(y(l))$. In order to evaluate the network-wide performance of a routing configuration, we shall define the expected e2e path queuing delay d_{mean} as follows:

$$d_{\text{mean}}(X, R) = \sum_{k \in N} \sum_{p \in P_k} \left(r_p^k \,.\, x(k) \right) d_p = \sum_{l \in L} y(l) \,.\, d_l(y(l)) \tag{4.14}$$

That is to say, a weighted mean e2e queuing delay, where the weight for each path is how much traffic is sent through it, i.e., $r_p^k \,.\, x(k)$, or in terms of links, the weight for each link is how much traffic is traversing it, namely $y(l)$. A large mean e2e queuing delay translates into bad performance for all the traffic and not only for the traffic that traverses a particular loaded link. We prefer a weighted mean

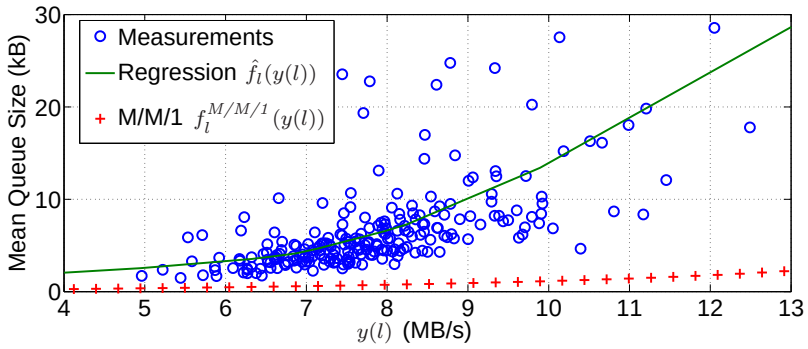


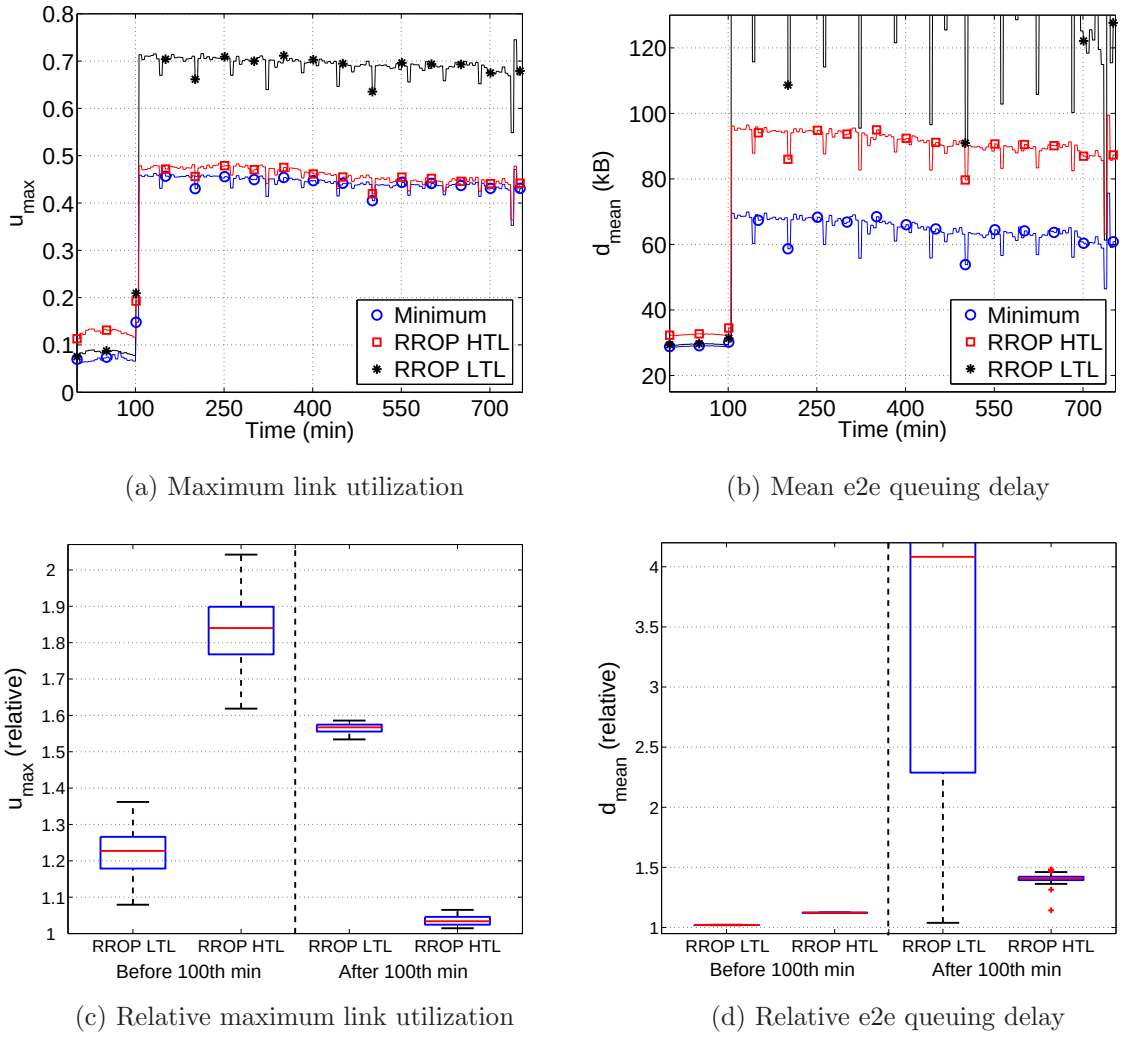
Figure 4.16 — Mean queue size, measurements and approximations

queuing delay to a simple total delay because it reflects more precisely performance as perceived by traffic. Two situations where the total delay is the same, but in one of them most of the traffic is traversing heavily delayed links should not be considered as equivalent. Note that, by Little's law, the value $f_l(y(l)) = y(l) \,.\, d_l(y(l))$ is proportional to the volume of data in the queue of link l. We will then use this last value as the addend in the last sum in (4.14), since it is easier to measure than the queuing delay.

The function $f_l(y(l))$ is unknown and in the literature it is generally estimated using a classical *M/M/1* model, where $f_l^{M/M/1}(y(l)) = y(l)/(c_l - y(l))$ [121]. However, in [114] authors show that a simple *M/M/1* model has little to do with reality, and so they propose to use a non-parametric regression technique to estimate $f_l(y(l))$ from measurements without assuming any given model. Figure 4.16 depicts the real mean queue size of an operational network link at Tokyo obtained from [126], together with the *M/M/1* estimation $f_l^{M/M/1}(y(l))$ and the non-parametric regression $\hat{f}_l(y(l))$. It is clear that $f_l^{M/M/1}(y(l))$ consistently underestimates the real queue size value, while $\hat{f}_l(y(l))$ provides quite accurate results. Thus, we shall use this estimation in the definition of d_{mean}.

In order to appreciate the disadvantage of the RROP optimization problem as regards network-wide performance and QoS, we shall evaluate the performance of SRR regarding both $u_{\max}$ and d_{mean}. From now on we shall use RROP as a reference to SRR, recalling that the robust routing optimization problem is the one described in (4.6). In this evaluation we consider the same traffic scenario depicted in figure 4.15. Based on the historical routing matrix of Abilene R_{o} and assuming that traffic is known in advance, we define two different polytopes, $\mathbb{X}^{\text{LTL}}$ and $\mathbb{X}^{\text{HTL}}$, the former adapted to the LTL period, before the 100th minute, and the latter adapted to the HTL period, after the 100th minute. The corresponding SRR configurations will be now referred as RROP LTL and RROP HTL. In this evaluation, both RROP LTL and RROP HTL use the same set of paths, namely the paths obtained from (4.6) for polytope $\mathbb{X}^{\text{LTL}}$.

Figure 4.17(a) depicts the maximum link utilization $u_{\max}$ and figure 4.17(b) the mean end-to-end queuing delay d_{mean} during the evaluation period. As a reference



(a) Maximum link utilization

(b) Mean e2e queuing delay

(c) Relative maximum link utilization

(d) Relative e2e queuing delay

Figure 4.17 — (a) Maximum link utilization and (b) mean end-to-end queuing delay. Traffic demand volume abruptly increases after the 100th minute. (c) and (d) depict the corresponding boxplot performance summaries, relative to the optimal values.

for comparison, we also compute the optimal routing configurations for every TM at every time t, regarding both $u_{\max}$ and d_{mean}. In the first case, we use (4.1) with a fixed set of paths to compute a minimal $u_{\max}$ routing configuration for every time t. In the second case, we use the algorithms presented in [114] to compute a minimal d_{mean} routing configuration for every time t. These algorithms are explained in section 4.6. Figures 4.17(c) and 4.17(d) present a boxplot summary on the performance of RROP LTL and RROP HTL, relative to the optimal values.

Let us focus the attention on the performance of RROP HTL after the 100th minute. Despite achieving an almost optimal performance as regards $u_{\max}$, with a difference smaller than 4%, RROP HTL obtains a queuing delay that constantly exceeds the optimum by almost 40% under a fairly network load. Such a difference may not be even acceptable from a QoS perspective, where end-to-end delays are even more important

$$
\begin{array}{rcll}
\text{minimize} & u_{\text{mean}} & & (4.16)\\
\text{subject to:} & & & \\
\sum_{l\in L}\sum_{k\in N}\sum_{p\in P_k}\frac{1}{c_l}\lambda_l^p \,.\, r_p^k \,.\, x(k) & \leqslant & u_{\text{mean}}.q & \forall\, X\in\mathbb{X}\\
\sum_{k\in N}\sum_{p\in P_k}\lambda_l^p \,.\, r_p^k \,.\, x(k) & \leqslant & u_{\text{max}}^{\text{thres}}.c_l & \forall\, l\in L,\ \forall\, X\in\mathbb{X}\\
\sum_{p\in P_k} r_p^k & = & 1 & \forall\, k\in N\\
r_p^k & \geqslant & 0 & \forall\, p\in P_k,\ \forall\, k\in N
\end{array}
$$

than network congestion. The situation becomes even more critical for RROP LTL, in which case the mean end-to-end delay grows more than 400% w.r.t. the ideal values, even if $u_{\max}$ remains bounded. As we show next, this loss in performance is a direct consequence of the local criterion used in RROP.

4.5.1 Improving Network-Wide Performance and E2E QoS

As we show in figure 4.17(b), the minimization of $u_{\max}$ leads to a distribution of traffic that results in an excessive end-to-end delay. Using the mean delay $d_{\text{mean}}(X,R)$ as the objective function in (4.6) would be an interesting approach to ease the problem; however, $f_l(y(l))$ is a non-linear function and the optimization problem becomes too difficult to solve. As we previously said, optimization under uncertainty is more complex than classical optimization, and simple optimization criteria should be used. Let us consider a very simple network-wide linear objective function, namely the mean link utilization $u_{\text{mean}}(X,R)$, defined as:

$$u_{\text{mean}}(X,R) = \frac{1}{q}\sum_{l\in L} u_l \tag{4.15}$$

The mean link utilization considers at the same time the load of every link in the network and not only the utilization of the most loaded link; as we will show in the results, such an objective function provides a better global performance as regards end-to-end delay. However, a direct minimization of u_{mean} does not assure a bounded maximum link utilization, which is not practical from an operational point of view. In this sense, we propose to change the objective function in (4.6) by u_{mean}, while bounding the maximum link utilization by a certain threshold $u_{\max}^{\text{thres}}$ a priori defined. The resulting problem, which we shall call the Robust Routing Mean Utilization Optimization Problem (RRMP), is defined in (4.16).

Problem (4.16) is solved in the same way as (4.6), using the same recursive algorithm proposed in [101]. Note that (4.16) adds only a new constraint per each new traffic demand in $\mathbb{X}$ w.r.t. (4.6). To be more precisely, it only adds a new constraint for each extreme point of $\mathbb{X}$, which does not represent a problem from the numerical complexity side of the algorithm. The drawback of (4.16) is its dependence on the

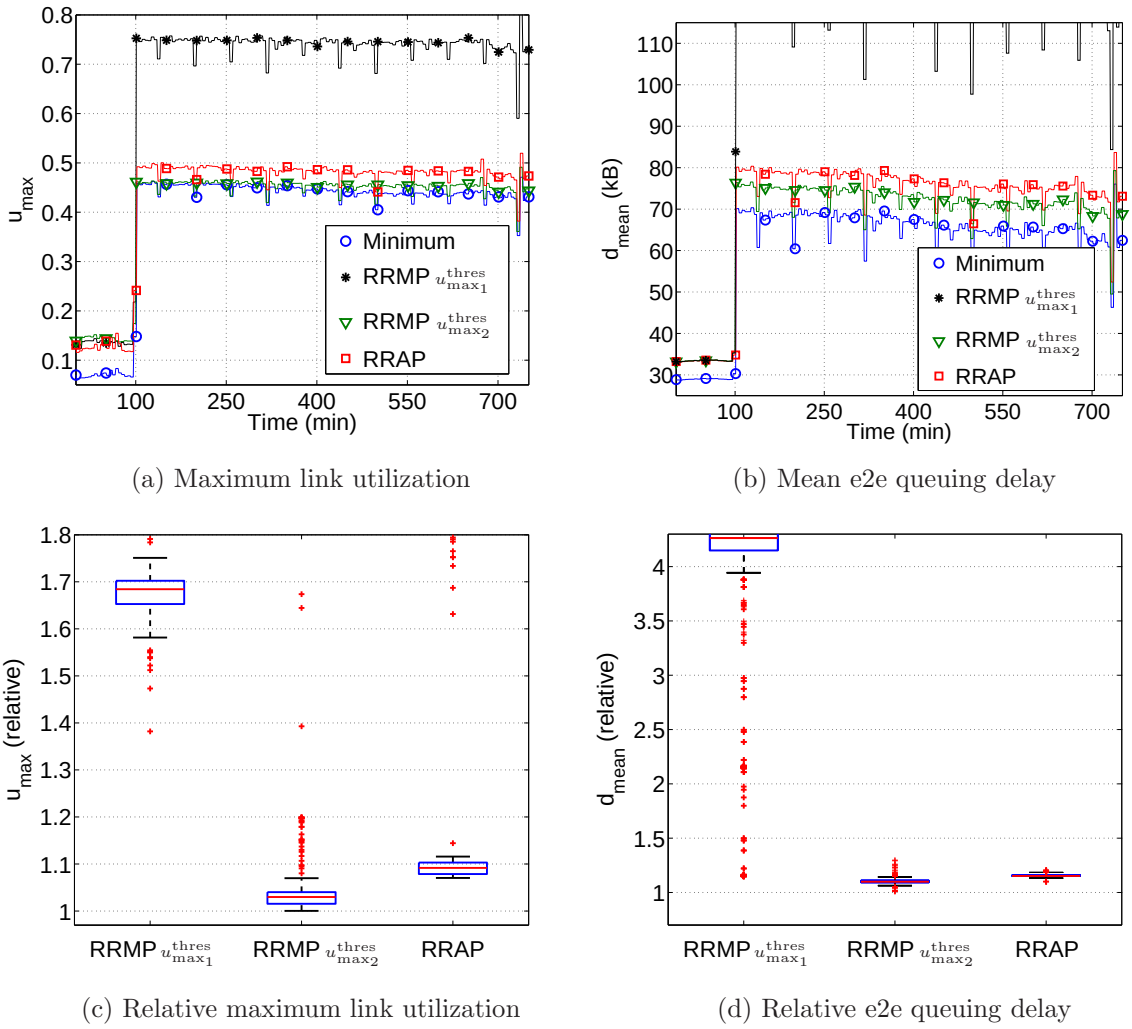
$$\begin{aligned} \text{minimize} \quad & u_{\text{aof}} = \beta \,.\, u_{\max} + (1-\beta) \,.\, u_{\text{mean}} && (4.17) \\ \text{subject to:} & \\ \sum_{l \in L} \sum_{k \in N} \sum_{p \in P_k} \tfrac{1}{c_l} \lambda_l^p \,.\, r_p^k \,.\, x(k) \quad & \leqslant \quad u_{\text{mean}}.q \quad \forall\, X \in \mathbb{X} \\ \sum_{k \in N} \sum_{p \in P_k} \lambda_l^p \,.\, r_p^k \,.\, x(k) \quad & \leqslant \quad u_{\max}.c_l \quad \forall\, l \in L,\ \forall\, X \in \mathbb{X} \\ \sum_{p \in P_k} r_p^k \quad & = \quad 1 \qquad \forall\, k \in N \\ r_p^k \quad & \geqslant \quad 0 \qquad \forall\, p \in P_k,\ \forall\, k \in N \end{aligned}$$

value of $u_{\max}^{\text{thres}}$, which directly influences the routing performance as we will see shortly. An interesting choice for $u_{\max}^{\text{thres}}$ would be to use the output of (4.6), namely $u_{\max}^{\text{robust}}$. To some extent, this would result in a similar routing solution but with better traffic balancing.

A alternative approach would be to minimize both the value of $u_{\max}$ and u_{mean} at the same time, which constitutes a problem of multi-objective optimization (MOO). MOO problems are generally more difficult to solve because traditional single-objective optimization techniques can not be directly applied. Nevertheless, the problem of finding all the Pareto-efficient solutions to a linear MOO problem is well known and different approaches can be used to treat the problem [122, 123]. In this work we consider an intuitive and easy approach to solve a MOO problem with standard single-objective optimization techniques. The approach consists in defining a single aggregated objective function (AOF) that combines both objective functions. We define a weighted linear combination of $u_{\max}$ and u_{mean} as the new objective function $u_{\text{aof}} = \alpha \,.\, u_{\max} + (1-\alpha) \,.\, u_{\text{mean}}$, where $0 \leqslant \alpha \leqslant 1$ is the combination fraction. Despite its simple form, this new objective is very effective and provides accurate results for both performance indicators. We shall call this new optimization problem the Robust Routing AOF Optimization Problem (RRAP), defined in (4.17). As before, problem (4.17) is solved with the same algorithms used in (4.6).

4.5.2 Comparison between RRMP and RRAP

Let us evaluate both the RRMP and the RRAP extensions of SRR in the same traffic scenario previously considered. In order to appreciate the dependence of RRMP on the maximum link utilization threshold $u_{\max}^{\text{thres}}$, two different thresholds are used in the evaluation: $u_{\max_1}^{\text{thres}} = 1$, which corresponds to the constraint $u_{\max} \leqslant 1$ in (4.6), and $u_{\max_2}^{\text{thres}} = u_{\max}^{\text{robust}}$, where $u_{\max}^{\text{robust}}$ is the output of RROP HTL in figure 4.17. In the case of RRAP, the weight α is set to 0.5, namely an even balance between $u_{\max}$ and u_{mean}. This may impress as a somewhat naive approach to the reader, but practice shows that this choice provides in fact very good results.



(a) Maximum link utilization

(b) Mean e2e queuing delay

(c) Relative maximum link utilization

(d) Relative e2e queuing delay

Figure 4.18 — (a,c) Maximum link utilization and (b,d) mean end-to-end queuing delay for RRMP and RRAP. The boxplot performance summaries are relative to the optimal values.

Figure 4.18 depicts the comparison as regards (a) maximum link utilization and (b) mean end-to-end queuing delay. Let us focus our attention on the operation after the 100th minute, as all robust routing configurations use $\mathbb{X}^{\text{HTL}}$ as uncertainty set. To be as fair as possible, both RRMP and RRAP use the same set of paths as those used by RROP in figure 4.17. The figure clearly shows that the performance of RRMP strongly depends on the threshold $u^{\text{thres}}_{\text{max}}$. In the case of $u^{\text{thres}}_{\text{max}_1}$, the attained maximum link utilization is well beyond the optimal values, reaching almost a 70% of relative performance degradation. This overload directly translates into huge mean end-to-end queuing delays. Results are quite impressive when considering the second threshold, both as regards $u_{\text{max}}$and d_{mean}. RRMP with threshold $u^{\text{thres}}_{\text{max}_2}$ provides a highly efficient robust routing configuration, showing that it is possible to improve current implementations of SRR with a slight modification of the objective function. However, this dependence on the threshold $u^{\text{thres}}_{\text{max}}$ introduces a new tunable parameter, something undesirable when looking for solutions that simplify network management.

As regards RRAP, obtained results are slightly worse than those obtained by RRMP with threshold $u_{\max_2}^{\text{thres}}$, but still very close to the optimal performance, with a relative performance degradation of about 10% as regards $u_{\max}$ and d_{mean} w.r.t. an optimal routing configuration. Nevertheless, RRAP has no tunable parameter apart from the combination factor α, which in fact is set to a half independently of the traffic situation. In the following sections, we use the RRAP algorithm combined with the RRLB approach to provide a robust and adaptive routing configuration that can handle both normal operation and anomalous traffic behaviors, while providing accurate network-wide performance levels from a QoS perspective.

4.6 Reactive Robust Load Balancing vs Dynamic Load Balancing

In this section we present a comparative analysis between our Reactive Robust Load Balancing algorithm and traditional Dynamic Load Balancing mechanisms. To begin with, we shall introduce the basic notions used in most DLB algorithms. Using these notions we shall evaluate the performance of some traditional DLB algorithms, particularly regarding their routing stability and dynamic convergence when faced with volume anomalies. To be as fair as possible in our comparative study, we present some variants to traditional DLB schemes that improve their performance in those cases. Finally, we compare RRLB and our improved DLB algorithms using both normal operation and anomalous traffic scenarios.

4.6.1 Dynamic Load Balancing

Given a fixed set of paths P_k for every OD pair $k \in N$, the objective of DLB is to iteratively adapt the routing fractions $\{r_p^k\}_{k \in N, p \in P_k}$ to minimize a certain cost function $g(X, R)$. DLB algorithms generally work in a distributed fashion, without relying on any centralized entity. In this kind of mechanisms, a path cost function ϕ_p is defined, and each OD pair greedily minimizes the cost it obtains from each of its paths. This context constitutes an ideal case study for game theory, and is known as Routing Games in its terminology [113, 117].

Since each OD pair may arbitrarily balance traffic among its paths, we will assume that OD pairs are constituted of infinitely many agents. These agents control an infinitesimal amount of traffic, and decide along which path to send their traffic. In this context, r_p^k represents then the fraction of agents of OD pair k that choose path p. If each of these agents acts selfishly, then the system will be at equilibrium when no agent may decrease its cost by unilaterally changing its path decision. This situation constitutes what is known as a Wardrop Equilibrium (WE) [115]. A WE can be formally defined as follows:

Definition 1 *The paths vector $\{r_p^k\}_{k \in N, p \in P_k}$ is a Wardrop Equilibrium if, for each OD pair $k \in N$ and for each couple of paths $p, q \in P_k$ with $r_p^k > 0$, it holds that $\phi_p \leq \phi_q$.*

Intuitively speaking, a WE is a situation where each OD pair routes its traffic only on those paths with minimum cost for himself.

The path cost ϕ_p is generally defined in terms of a certain nonnegative, nondecreasing and continuous link cost function $\phi_l(y(l))$. There are roughly two kinds of games depending on the definition of ϕ_p. A Congestion Routing Game defines the path cost as $\phi_p = \sum_{l \in p} \phi_l(y(l))$. On the other hand, a Bottleneck Routing Game defines $\phi_p = \max_{l \in p} \phi_l(y(l))$.

Much effort has been devoted to characterize the resulting equilibrium WE of these games. In this sense, a certain social cost function is defined, which measures the dissatisfaction of OD pairs as a whole. A vector of paths $\{r_p^k\}_{k\in N, p\in P_k}$ is said to be the optimum if it minimizes this function. The objective of the characterization is then to quantify the difference between the optimum and the resulting WE. In the case of a congestion game, the typical social cost function is the one defined in (4.14), i.e., $\sum_{l\in L} y(l)d_l(y(l)) := \sum_{l\in L} f_l(y(l))$, whereas for a bottleneck game the social cost is usually the maximum link cost function $\phi_l(y(l))$ over all links, i.e.,$\max_{l\in L} \phi_l(y(l)))$.

It may be proved that the WE of a congestion game coincides with the unique minimum of the so-called potential function $\Phi(R) = \sum_{l\in L} \int_0^{\rho_l} \phi_l(x)dx$ [113]. This means that if $f_l(y(l))$ is continuous, differentiable, non-decreasing, and convex, the WE of a congestion game with $\phi_l(y(l)) = f_l^{'}(y(l))$ is socially optimum. In this sense, to minimize d_{mean} through DLB, we will play a Congestion Routing Game with a link cost equal to the derivative of the link mean queue size. From now on, we shall note this game as MinDG (Minimum Delay Game).

On the other hand, the characterization of the WE of a bottleneck game is somewhat more complicated. In fact, it is relatively easy to see that in this case, the WE is not even unique. Moreover, and rather unfortunately, it has been proved in [116] that even if there always exist at least one WE that is socially optimum, nothing may be guaranteed about the rest of them, if any. However, the same paper proved that every WE that fulfills the so-called *efficiency condition* is optimum, where this condition is defined as follows:

Definition 2 *Let $B(p)$ denote the number of network bottleneck links over p. That is to say, $B(p) = \left|\left\{l \in p : \phi_l(y(l)) = \max_{j\in L, y(j)>0}\{\phi_j(y(j))\}\right\}\right|$, where $|\cdot|$ indicates the cardinality of $B(p)$. Then, a WE is said to satisfy the efficiency condition if all OD pairs route their traffic along paths with a minimum number of network bottlenecks, i.e., for all $k \in N$ and $p, q \in P_k$ with $r_p^k > 0$, it holds that $B(p) \leq B(q)$. The network bottleneck links are simply those with the maximum utilization in the network.*

This result, which is relatively new, was not applied in the design of traditional DLB algorithms such as TeXCP or REPLEX, both of which strive to minimize the maximum link utilization by means of a greedy algorithm in the path utilization, i.e., a bottleneck game with $\phi_p = \max_{l\in p} u_l$). It could then be the case that these algorithms converge to a sub-optimal WE. Possible consequences of ignoring this result will be further discussed below. From now on, we shall note this game as MinUG (Minimum Utilization Game).

Let us now briefly discuss how to attain the WE for both routing games MinDG and MinUG. In a recent article [118], authors proved that if all OD pairs use no-regret algorithms, then the global behavior will converge to the WE. To be more precise, given a certain TM X to route, the instantaneous paths vector $\{r_p^k\}_{k\in N, p\in P_k}$ is very

close to the WE, and this difference vanishes with time. No-regret algorithms are a popular class of machine learning algorithms which always present small regret, no matter what sequence of learning-examples they see. We refer the reader to [119] for an overview of some of these algorithms.

This result is very general as it does not specify any particular algorithm, and all it requires is the use of no-regret algorithms by all OD pairs. In particular, we will consider a standard no-regret algorithm, known as the Weighted Majority Algorithm (WMA) [120]. The pseudo-code of the WMA for OD pair k is described in Algorithm 2.

Algorithm 2 Weighted Majority Algorithm (WMA)

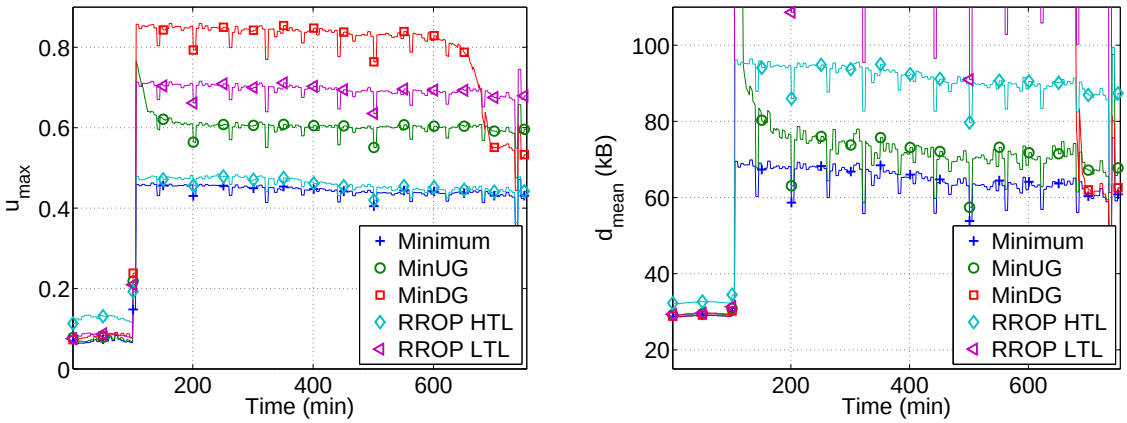
for $t = 1, \ldots, \infty$ **do**
  Obtain path costs $\phi_p \; \forall p \in P_k$
  for every path $p \in P_k$ **do**
    if $\phi_p > \min_{q \in P_k} \phi_q$ **then**
      $r_p^k \leftarrow \beta \times r_p^k$
    end if
  end for
  Normalize the r_p^k
end for

At each iteration t, those paths whose cost is bigger than the minimum are punished by multiplying their respective r_p^k by a certain constant $\beta < 1$ (we use $\beta = 0.95$ in our following evaluations). Actually, and in order to avoid unnecessary changes in the traffic distribution, we shall only update r_p^k when the corresponding path cost is bigger than the minimum cost plus a certain margin. In the case of MinUG we shall fix this margin at 0.005, whereas for MinDG we shall use 5% of the minimum.

4.6.2 A Preliminary Comparison

Let us present some first evaluations that will help to gain insight into both DLB mechanisms MinDG and MinUG, highlighting some of their respective shortcomings. The evaluations are performed using once again the traffic scenario depicted in figures 4.15 and 4.17, where a volume anomaly highly modifies the traffic in one single OD flow from the 100-th minute until the end of the evaluation. As before, we consider both $u_{\max}$ and d_{mean} as the performance indicators.

In order to evaluate the MinDG algorithm, we must define the cost function $f_l(y(l))$ and obtain its derivate $f_l^{'}(y(l))$. As we did in section 4.6, we use the non-parametric regression technique applied in [114] to learn $f_l(y(l))$ and its derivative from measurements. This regression technique additionally assures that the estimated function $\hat{f}_l(y(l))$ is continuous, differentiable, non-decreasing, and convex, a necessary condition for MinDG to converge to the optimum.



(a) Maximum link utilization

(b) Mean e2e queuing delay

Figure 4.19 — Maximum link utilization and mean end-to-end queuing delay. Traffic demand volume abruptly increases after the 100th minute.

As a reference for Robust Routing, we include in the evaluation the SRR configurations RROP LTL and RROP HTL, depicted in figure 4.17. The different TMs used in the evaluation are presented to the DLB mechanisms in consecutive temporal order. Both MinUG and MinDG are initialized at arbitrary values of r_p^k, which are updated as new link load measurements arrive. Although TMs in Abilene are collected every 5 minutes, we assume that each OD pair receives these measurements every minute. This means that five updates of the corresponding r_p^k values are performed for each new TM. Results are shown then for every minute.

Additionally, we also include the optimum values for $u_{\max}$ and d_{mean} for every TM in the evaluation. The optimal values of d_{mean} for each TM are computed off-line with MinDG, letting enough consecutive updates of r_p^k to ensure convergence to the minimum. To be as fair as possible, all mechanisms use the same set of paths.

Results are presented in figure 4.19. Let us focus the attention on the performance obtained by the dynamic schemes after the 100-th minute, when the volume anomaly occurs. A first important observation is that both MinDG and MinUG present an important overshoot, with an absolute difference with the optimum values of $u_{\max}$ of approximately 40%. The convergence of MinDG is particularly slow, taking more than 8 hours to finally converge. However, it should be noted that when it eventually converges, it obtains a value of d_{mean} that is very close to the optimum. In terms of $u_{\max}$, the absolute difference w.r.t. the optimum is approximately 10%.

Special attention deserves the case of MinUG. After a shorter convergence time of approximately 100 minutes, the resulting value of $u_{\max}$ is not the optimum. Let us recall that this kind of game models schemes such as REPLEX of TeXCP, where the idea is to converge to a routing configuration that minimizes $u_{\max}$ [111, 112]. However, in this case, the absolute difference with the optimal values is more than 15%.

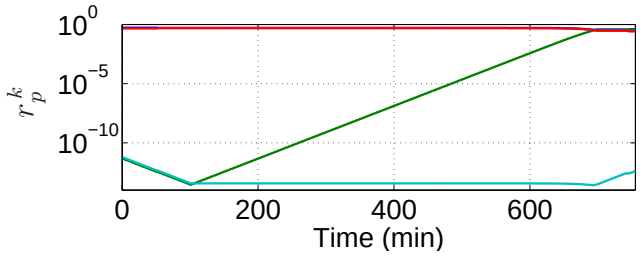


Figure 4.20 — Evolution of r_p^k for the anomalous OD pair (MinDG)

4.6.3 Improving Dynamic Load Balancing

The preliminary evaluation previously done shows some conception drawbacks of the MinUG and MinDG algorithms as presented before. In this section we shall explain the origin of these problems and present some enhanced mechanisms to overcome them.

Convergence Time

Both DLB algorithms present an important overshoot and a significant settling time in the presence of sudden and large traffic variations. If the traffic anomaly is a perfect step, then the overshoot is unavoidable. We will try to address the long settling time instead. The reason behind this problem is relatively simple, as shown in figure 4.20. The graph depicts the evolution over time of the corresponding r_p^k values of the anomalous OD pair for MinDG in the previous evaluation. Although r_p^k values change exponentially fast, the ones that should increase are so small at the moment of the anomaly that it takes them a very long time to converge. A possible solution would be to impose a minimum value to all r_p^k. However, this will affect the precision of the algorithm and will still result in significant settling times.

Actually, r_p^k may be seen as an indicator of the performance of path p in previous iterations of the game. A very small r_p^k means that p performed very badly w.r.t. the rest of the paths in the past. However, when the anomaly occurs, conditions severely change and history is no longer as relevant. If we consider that we are in such situation, we could for instance completely ignore history and restart the game by setting $r_p^k = 1/|P_k|$, $\forall k \in P_k$. Before deciding how to reassign r_p^k, we will discuss how an OD pair may decide if it should restart its game or not.

Consider a situation where most of the traffic for OD pair k is routed along a path that is not the cheapest, and that the r_p^k corresponding to the minimum-cost path is very small. This could mean that although the former performed better in the past, this is no longer true and some traffic should be re-routed to the latter. However, this "suspicious" situation could be due to noisy measurements. To make sure that the game has actually changed and that it should be restarted, we will require such a situation to persists during a certain number of consecutive iterations. Once we detect that the game should be restarted, we re-route some of the traffic that was being routed along the path with the biggest r_p^k to the cheapest one. To

avoid over-reacting, the amount will be proportional to the relative difference in cost. Finally, remember that with WMA fast adaptation is achieved when the r_p^k are not too small. The objective with this "game restart" is simply to move r_p^k from critically small values. The algorithm will then rapidly converge to the optimum. We now present the pseudo-code of the complete algorithm for OD pair k, which we shall call WMA with Restart (WMA-R):

Algorithm 3 WMA with Restart (WMA-R)

for $t = 1, \ldots, \infty$ **do**
  Obtain path costs $\phi_p \; \forall p \in P_k$
  Determine $p_{\min} = \underset{p \in P_k}{\operatorname{argmin}} \, \phi_p$ and $p_{\max} = \underset{p \in P_k}{\operatorname{argmax}} \, r_p^k$
  if $(r_{p_{\min}}^k < 0.1)$ and $(\phi_{p_{\min}} + \phi_{th} < \phi_{p_{\max}})$ **then**
    $n_e^k \leftarrow n_e^k + 1$
  else
    $n_e^k \leftarrow 0$
  end if
  if $n_e^k \leq n_{th}^k$ **then**
    Perform a normal iteration of WMA (cf. Algorithm 2)
  else
    $n_e^k \leftarrow 0$
    $\Delta_r \leftarrow \min\left\{\frac{\phi_{p\max}}{\phi_{p_{\min}}} - 1, 1\right\} \times \frac{r_{p\max}^k - r_{p_{\min}}^k}{2}$
    $r_{p_{\max}}^k \leftarrow r_{p_{\max}}^k - \Delta_r$
    $r_{p_{\min}}^k \leftarrow r_{p_{\min}}^k + \Delta_r$
  end if
end for

The new variable n_e^k counts the number of consecutive occurrences of a "suspicious" situation; we shall use $n_{th}^k = 3$ in the following evaluations. The threshold ϕ_{th} assures that the difference in cost between paths is significant. In particular, we will take $\phi_{th} = 0.005$ for MinUG and $\phi_{th} = 0.2\phi_{p_{\min}}$ for MinDG. Finally, note that when the game is restarted, we re-route a certain amount of traffic from $p_{\max}$ to $p_{\min}$, but at most the amount of traffic routed along each path is equalized.

Converging to the Social Optimum in Bottleneck Games

Figure 4.19 shows an example in which MinUG does not converge to the optimum, and obtains a difference of 15% with respect to the optimum MLU. The reason behind this poor performance is simply that MinUG does not take into account the result regarding the optimality of the WE and the efficiency condition previously discussed. This result states that if all OD pairs send their traffic along paths with a minimum number of network bottleneck links at a given WE, then this WE is the optimal. Note that this condition is only sufficient, meaning that a WE that fulfills the efficiency condition may not exist. A simple example of such case is a single OD pair with two paths of different lengths, where all links have the same capacity.

The problem that we analyze now is how to design a path cost function ϕ_p that takes into account this condition, so that the DLB algorithm that uses it converges to the correct WE, when possible. The two main difficulties in the design of such path cost function are the following. Firstly, the number of bottleneck links in a path is an integer, thus not continuous on r_p^k. Secondly, the probability of two links having exactly the same utilization is zero, and as such we should consider the number of links that have an utilization *similar* to the network bottleneck.

The objective is then to find a cost function that penalizes paths in which several bottleneck links have similar utilizations, and that it does not switch between values to avoid oscillations. A candidate ϕ_p that fulfills these two conditions is the so-called *log-sum-exp* function. Consider a set of arbitrary numbers $A = \{a_i\}$; the log-sum-exp function $g(A)$ is defined as follows:

$$g(A) = \frac{1}{\gamma_A} \log \left(\sum_{i=1,..,|A|} e^{\gamma_A a_i} \right) = a_{i^*} + \frac{1}{\gamma_A} \log \left(1 + \sum_{i=1,..,|A| \wedge i \neq i^*} e^{\gamma_A (a_i - a_{i^*})} \right) \quad (4.18)$$

Consider the special case in which $a_{i^*} = \max A$. It should be clear that if a_{i^*} is significantly bigger than the rest of the elements in A, the above convex and non-decreasing function constitutes an excellent approximation of a_{i^*}. In fact, it easy to prove that $a_{i^*} \leq g(A) \leq a_{i^*} + \log(|A|)/\gamma_A$, meaning that we may control the precision of the approximation through the parameter γ_A: the bigger this parameter, the more precise the resulting approximation. Moreover, as more elements in A are similar to the maximum, $g(A)$ approaches the upper bound, reaching it when all elements are the same.

We shall use the second term of (4.18) as a penalty to those paths with several links whose utilization is similar to $u_{\max}$. More precisely, given a path p, let $U_p = \{u_l\}_{l \in p}$ be the link utilizations in the path, and let $l^* \in p$ be the link with the biggest utilization in p. We will then use the penalty function with the alternative set U_p^*, which has the same elements as U_p, but substitutes u_{l^*} by $u_{\max}$. This results in the following cost function:

$$\phi_p = u_{l^*} + \frac{1}{\gamma_p} \log \left(1 + \sum_{l \in p \wedge l \neq l^*} e^{\gamma_p (u_l - u_{\max})} \right) \quad (4.19)$$

Even if this new cost function penalizes paths with several network bottleneck links, it also penalizes longer paths, which was not our original objective. A good choice of γ_p will alleviate this side-effect. For instance, we shall use $\gamma_p = \log(|p|)/\max\{0.01, u_{l^*}/10\}$. This way, we try to minimize the effect of $\log(|p|)$ and relativize the penalization to u_{l^*}.

4.6.4 Evaluation and Discussion

Let us finally compare the performance of Reactive Robust Routing against the enhanced versions of the MinUG and MinDG DLB algorithms. We shall consider two different flavors of Robust Routing, namely the traditional RROP algorithm that minimizes $u_{\max}$, and the RRAP algorithm that minimizes d_{mean} (using $\alpha = 0.5$). Both algorithms are coupled with the RRLB mechanism presented in section 4.4 to adapt traffic balancing in the event of volume anomalies. We shall use RRLB-OP and RRLB-AP to designate the Reactive Routing Load Balancing variants of RROP and RRAP respectively. Evaluations are performed in two different traffic scenarios, considering both normal operation and anomalous traffic situations.

Normal Operation Traffic Scenario

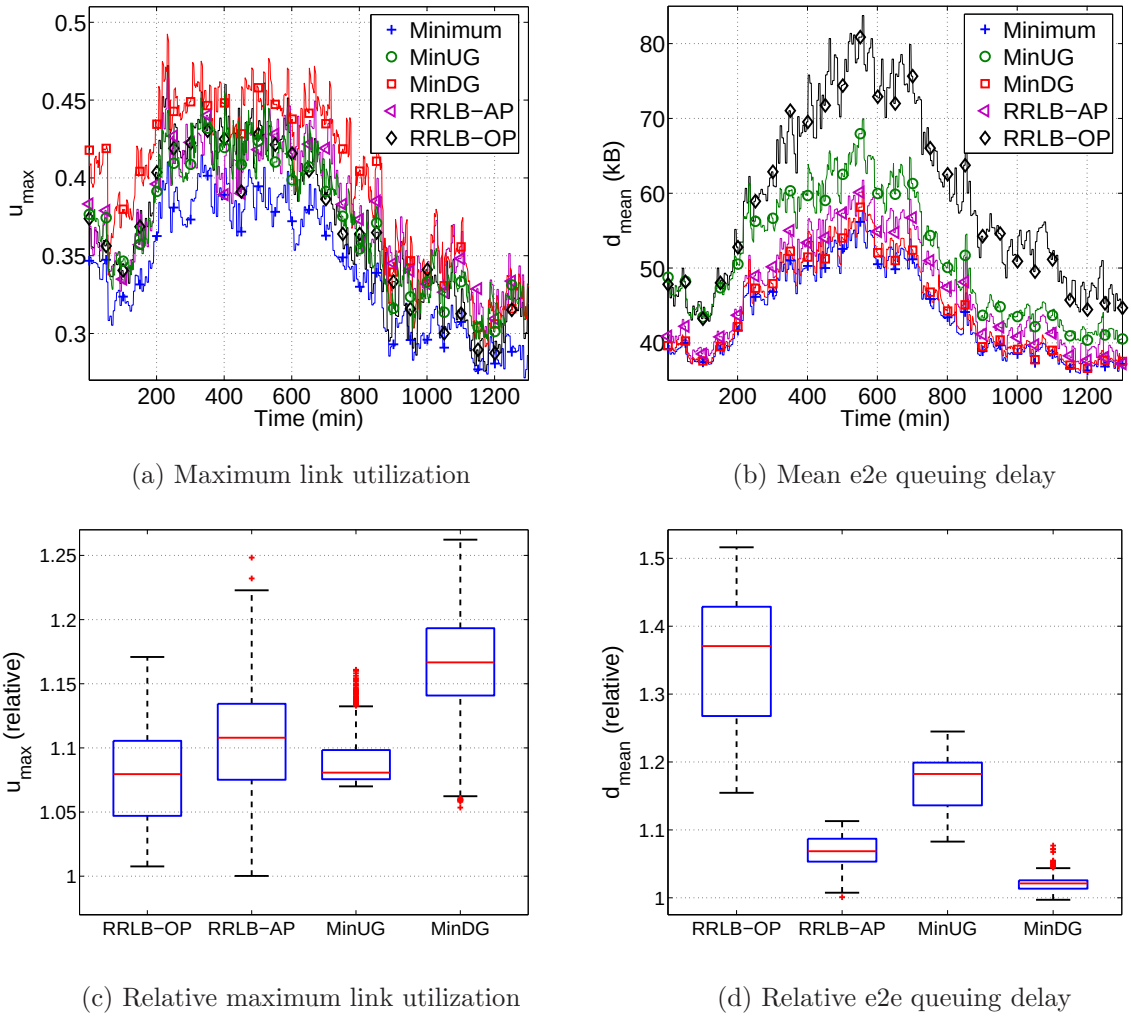
The first case-scenario corresponds to traffic in normal operation. The only variability in traffic is due to typical daily fluctuations. Figure 4.21 presents the evolution of $u_{\max}$ and d_{mean} for the different mechanisms, using a set of 260 TMs from the Abilene dataset [129]. All algorithms perform similarly as regards $u_{\max}$, depicted in figure 4.21(a). This may be further appreciated in the boxplot summary of figure 4.21(c), where we present the relative difference in $u_{\max}$ w.r.t the optimum for all the mechanisms. Naturally, the algorithm that performs best is MinUG, although its improvement over the rest is not significant. Note that the relative performance degradation is around 10% in most cases.

Figures 4.21(b) and 4.21(d) show that results are quite different as regards d_{mean}. The best results are obtained by MinDG, followed closely by RRLB-AP. However, RRLB-OP systematically obtains a significant difference w.r.t. the optimum, generally between 30% and 40%. Results obtained with MinUG are also quite poor for a normal operation scenario, presenting a relative difference of about 20% w.r.t. the optimum. These results further highlight the limitations of RROP and MinUG previously discussed: using $u_{\max}$ as a performance objective results in a relatively low maximum utilization, but neglects the rest of the links, impacting the network-wide performance.

Anomalous Traffic Scenario

The second case-scenario is the one considered in section 4.6.2, where there is a sudden and abrupt increase of the traffic volume carried by one OD flow. As a difference for Robust Routing algorithms w.r.t. the evaluation in figure 4.19, where traffic was assumed known in advance, this case-scenario corresponds to a real situation where traffic anomalies can not be forecast. To be fair with DLB mechanisms, both RRLB-AP and RRLB-OP use the RRLB mechanism previously described to adapt traffic balancing after the detection of the anomaly.

Figure 4.22 shows how the improvements discussed in section 4.6.3 for MinUG and MinDG result in a relatively smaller overshoot than before, but most importantly, the



(a) Maximum link utilization

(b) Mean e2e queuing delay

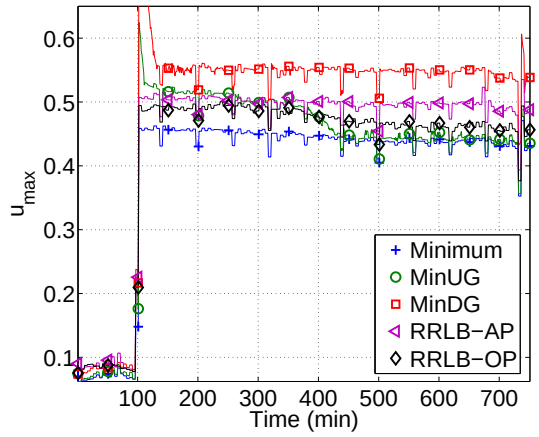
(c) Relative maximum link utilization

(d) Relative e2e queuing delay

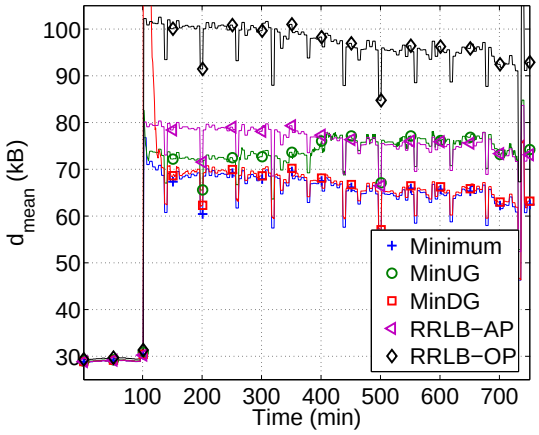
Figure 4.21 — (a,c) Maximum link utilization and (b,d) mean end-to-end queuing delay for normal traffic variation. The boxplot performance summaries are relative to the optimal values.

settling time has been significantly decreased; in the case of MinDG, from 600 minutes to less than 50 minutes. Moreover, note how the modified cost function proposed in section 4.6.3 results in MinUG converging to the socially optimum WE.

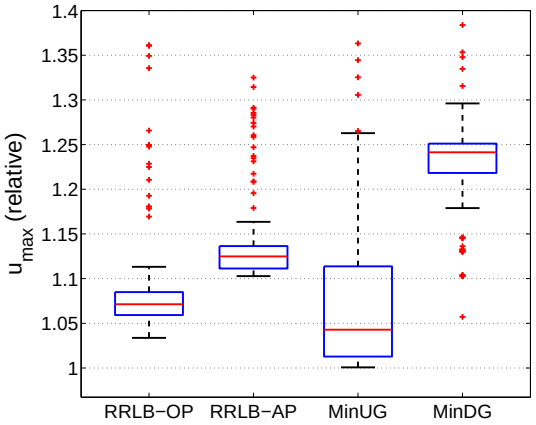
Regarding $u_{\max}$, both RRLB-OP and RRLB-AP obtain similar results, with a relative performance degradation generally smaller than 15%. Note that while relatively important, this performance degradation is surprisingly small if we consider that traffic increases more than 500% in less than 10 minutes. The same may be said about MinDG, which obtains a degradation between 20% and 25%. In terms of d_{mean}, MinUG and RRLB-AP perform similarly. They both clearly outperform RRLB-OP, achieving a relative mean queuing delay almost 30% smaller. These results reinforces once again our observations about the difficulty in RROP to attain global performance, and the advantages of using a simple network-wide objective function in Robust Routing. Moreover, they also illustrate the difference between MinUG and RROP. Even when



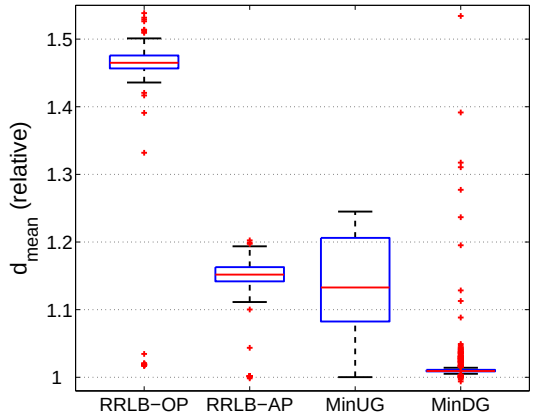
(a) Maximum link utilization



(b) Mean e2e queuing delay



(c) Relative maximum link utilization



(d) Relative e2e queuing delay

Figure 4.22 — (a,c) Maximum link utilization and (b,d) mean end-to-end queuing delay facing a volume anomaly. The boxplot performance summaries are relative to the optimal values.

MinUG was designed with the same objective than RROP, namely to minimize $u_{\max}$, the fact that in MinUG each OD pair greedily minimizes the path utilization results in a different overall behavior.

4.7 Conclusions

In this chapter we have studied the problem of routing optimization under highly variable and difficult to predict traffic demands. We have presented a comprehensive analysis of different plausible solutions to the problem, including traditional Prediction-Based Routing Optimization, Robust Routing Optimization, and Dynamic Load Balancing, evaluating their performance under different traffic scenarios.

From this study we may draw several conclusions. The most important of them is probably that we have shown that using a single routing configuration is not a cost-effective solution when traffic is relatively dynamic. Traditional Prediction-Based Routing Optimization may provide quite inefficient or even unfeasible routing configurations when traffic is uncertain and difficult to forecast. Stable Robust Routing Optimization offers performance guarantees against traffic uncertainty, but the associated trade-off between robustness and routing efficiency may be particularly difficult to manage with a single routing configuration. Besides, it still may present rather poor performance when faced with large volume anomalies. It is clear from our study that some form of dynamism is necessary, either in the form of Reactive Robust Routing and Load Balancing (RRLB) or Dynamic Load Balancing (DLB).

RRLB computes a nominal operation robust routing configuration, and provides an alternative robust routing configuration (using the same paths than in normal operation) for every possible single OD-flow anomalous situation. In order to detect these anomalous situations, link load measurements have to be gathered and processed by a centralized entity. On the other hand, DLB gathers the same measurements but also requires updating load-balancing in a relatively small time-scale. The added complexity is then to distribute these measurements to all ingress routers and to update the load-balancing on-line.

Our results show that the additional complexity involved in DLB is not justified when traffic variability is not very significant. In the case of large volume anomalies, DLB algorithms generally provide better results than RRLB after convergence, but they present an undesirable transient behavior, which is the main cause of why DLB approaches are generally met with reluctance by network operators. On the other hand, RRLB algorithms do not suffer from this problem, basically because the load balancing fractions are computed off-line in a robust basis, taking advantage of the goodness of the Robust Routing paradigm. The use of DLB becomes very appealing when volume anomalies are difficult to locate. Indeed, our RRLB algorithm assumes that volume anomalies occur in single OD-flows, but the case of multiple anomalous OD-flows is beyond the scope of the underlying SSB anomaly detection/localization algorithm.

Regarding RR in particular, we have shown that a local performance criterion such as the maximum link utilization $u_{\max}$ does not represent a suitable objective function as regards global network performance and QoS provisioning. In particular, we showed that an almost optimal robust routing configuration with respect to $u_{\max}$ can experience rather high mean end-to-end queuing delays, a very important performance indicator for all types of traffic. The maximum link utilization is widely used in current network optimization problems, particularly in most Robust Routing proposals, thus we believe that this simple evidence can help and should be considered in enhanced future implementations. In fact, we have shown that objective optimization functions can be kept simple, and yet better network-wide performance can be attained. By using a simple combination of performance indicators such as the maximum and the mean link utilization, we obtained a robust routing configuration that definitely outperforms current implementations from a global end-to-end perspective, while achieving very similar results as regards worst-case link utilization.

As regards DLB algorithms, we have shown that the transient behavior that they present under large traffic modifications can be effectively controlled, or at least alleviated, by simple mechanisms. Concerning the two different games that we have presented, conclusions are similar to those drawn for RR. Striving to minimize d_{mean} instead of $u_{\max}$ results in a somewhat bigger maximum link utilization, but a generally much better global performance.

The application of no-regret algorithms in DLB proved to be quite efficient and obtained results are very promising. Our study also highlighted a problem with previously proposed DLB algorithms, namely the wrong assumption that OD pairs that greedily minimize the path utilization converge to a routing configuration that minimizes $u_{\max}$. Based on recent results [116], we have explored the possibility of modifying the path cost function so that the resulting routing configuration is actually optimum, obtaining very interesting preliminary results.

Conclusions and Perspectives

Whether it be for economical, educational, cultural, or even social reasons, current Internet plays a paramount role in our life. Despite its extraordinary growth during the last decade, the health and proper functioning of the Internet as a whole still depends strongly on the performance of a small group of large-scale networks. The individual performance of these networks is then vital to support the continuous and solid development of the Internet as the main enabler of our information era.

A critical task to ensure the correct performance of the Internet is traffic analysis and monitoring. A global monitoring system for large-scale networks should be lightweight, scalable and fast. This thesis has focused on the statistical analysis of network traffic in large-scale networks, addressing the problem of fast anomaly detection, localization, and rapid routing reconfiguration countermeasures, using easily-available aggregated data to facilitate implementation and scalability issues.

The contributions of this thesis work in the fields of network traffic modeling and estimation, volume anomaly detection and localization, and robust routing optimization are various. The list of associated publications in each relative field shows the wide acceptance of our contributions by the research community. Summarized, the most important contributions of this thesis are the followings:

- A new parametric, linear, and parsimonious traffic model to describe the anomaly-free behavior of a Traffic Matrix in large-scale IP networks.

- Improved methods for efficient on-line Traffic Matrix estimation in large-scale IP networks.

- An optimal method for detecting network-wide volume anomalies in large-scale IP networks, using aggregated measurements. The method presents well established optimality properties in terms of detection and false alarm rates.

- An optimal method for fast network-wide volume anomaly detection and localization in large-scale IP networks, using aggregated measurements. This method

presents well established optimality properties in terms of detection/localization delay and false detection/localization rate.

- A Multi-Hour Robust Routing optimization extension for the Robust Routing paradigm, which better adapts to normal traffic variations.
- An new Robust Routing optimization technique, improved for QoS-provisioning.
- A Reactive Robust Load Balancing optimization approach to rapidly fightback volume anomalies, strengthening the global QoS of the network in the event of strong and abrupt congestion situations.
- A comprehensive comparison between these Robust Routing techniques and different Dynamic Load Balancing approaches.

The first contribution consisted in a novel parsimonious traffic model that correctly captures the anomaly-free behavior of a Traffic Matrix. Despite its strong assumptions, the model verified accuracy and consistency in real data examples taken from different operational networks. The Traffic Matrix Estimation method derived from this model presents estimation results comparable to those provided by the well known and highly accepted Tomo-Gravity Estimation approach, but provides a paramount advantage, that of using a linear parametric Gaussian model to describe OD-flows traffic. This in fact permitted to develop parametric volume anomaly detection algorithms with robust optimality properties, a property which is extremely difficult to achieve with the Tomo-Gravity method, and that it is absent in previous anomaly detection proposals.

The next contribution consisted in deeply evaluating and proposing improved versions for two Traffic Matrix estimation techniques recently proposed. The first of these estimation techniques consists in a recursive estimation of the Traffic Matrix, using the widely known Kalman filtering approach. We showed that the original proposal of this estimation method presents some important drawbacks and omissions that we treated. In particular, the estimation method relies on a difficult to calibrate state-space model, prone to mis-adjustments when the underlying TM properties vary. We proposed a simple state-space model to track the evolution of the TM, allowing in particular explicit variations in the mean values of OD-flows volume, which can further be tracked by the Kalman filter. This improved not only the accuracy but also the stability of the TM estimation method.

The second TM estimation technique consists in using neural networks to unveil the relation between links traffic and OD-flows volume. The former algorithm relies on Artificial Neural Networks (ANNs), which we showed are highly dependent on the number of neurons of the network topology. We proposed to use a new breed of neural networks to alleviate the problem, known as Random Neural Networks (RNNs). Using RNNs instead of ANNs showed to provide much more robust results, improving the application of the estimation technique.

The second group of contributions regards the field of volume anomaly detection and localization in large-scale IP networks. Using the parsimonious traffic model developed in the first part of the thesis, we introduced a method for detecting network-wide volume anomalies in the TM from coarse-grained link traffic measurements. The TM model permits to filter the contribution of anomaly-free OD-flows traffic into SNMP measurements, motivating our original approach of treating the detection of volume anomalies as a statistical change detection problem with a nuisance parameter.

The most interesting contributions of the approach are the well established optimality properties that it presents, maximizing the detection rate for a bounded false alarm rate. Optimality support is fundamental in the conception of general algorithms, not tied to any particular network and more important, independent of individual evaluations in particular network and traffic scenarios. In-house methods may work rather well in certain scenarios, but without a principled and generalizable support they can be easily rebutted. In the evaluations, we showed that our method clearly outperforms the celebrated PCA method for network-wide anomaly detection, which is definitively the most cited work in the literature. The PCA approach is a data-driven approach, and as such it suffers from the particular characteristics of the data, requiring a lot of fine-tunning to provide proper results.

We also developed an optimal sequential method for fast network-wide volume anomaly detection and localization in large-scale IP networks, using as well coarse-grained link traffic measurements as input. The method permits not only to detect a volume anomaly in the TM but also to localize the anomalous OD-flow among all the OD-flows of the TM, improving the post-treatment of the problem. Sequential approaches are used in decision theory to minimize the number of observations needed to decide among the possible hypotheses that better explain the change detected. Our method minimizes the maximum mean detection/localization delay for an upper bounded probability of false localization and a lower bounded mean time between consecutive false alarms, a usual measure of the false alarm rate. The mean detection/localization delay is a crucial design criterion; indeed, the faster the localization, the faster the resolution of the problem. Evaluations with real traffic and large-scale network topologies showed that in practice, the method has similar or even better performance than different representative approaches proposed in the literature, additionally providing the aforementioned optimally properties absent in current approaches.

The complexity evaluation revealed that both network-wide anomaly detection methods have a similar or even smaller numerical complexity than those methods evaluated, showing that we can design accurate and theoretically supported methods without increasing computational complexity. Both methods can be efficiently applied for on-line volume anomaly detection without any kind of modifications to current measurement technology. Even though we have tested a five minutes time scale as the shorter time resolution, the numerical complexity evaluation permits to affirm that a much shorter time scale could be envisioned. In addition, the short calibration

step required by the underlying TM model permits to improve the robustness of the methods, avoiding contamination problems while learning the characteristics of the anomaly-free traffic. This is a paramount advantage w.r.t. current anomaly detection approaches, because collecting long traces of purely anomaly-free traffic in large-scale networks is a challenging task.

The last group of contributions regards robust routing optimization techniques. The objective of our studies in the robust routing field was to conceive an appropriate countermeasure against volume anomalies in large-scale networks, particularity regarding the QoS impairments caused by this kind of traffic anomalies. The robust routing paradigm permits to optimize the routing configuration even when traffic is highly variable and uncertain, a highly appealing feature to address the countermeasures problem. The final solution was developed incrementally, providing slight variations and improvements to the Stable Robust Routing (SRR) approach proposed in the literature.

The first contribution of our study was to show that despite being one of the most important characteristics of SRR, using a single routing configuration is not and efficient and cost-effective solution when traffic is highly dynamic. We proposed a first extension to SRR, building a Multi-Hour Robust Routing configuration to follow traffic variations under normal-operation scenarios. For doing so, we introduced the notion of temporal uncertainty set, and implemented a quasi-optimal partition algorithm, which permits to compute the optimal times to modify the routing configuration.

In our investigations, we discovered that all Robust Routing algorithms proposed in the literature have a serious handicap to provide QoS-based routing optimizations, due to the objective function they optimize. In particular, we showed that a local performance criterion such as the maximum link utilization $u_{\max}$, by far the most popular Traffic Engineering objective function used in the Robust Routing field, does not represent a suitable objective function as regards global network performance and QoS provisioning. We showed that an almost optimal robust routing configuration with respect to $u_{\max}$ can experience rather high mean end-to-end queuing delays, a very important performance indicator for all types of traffic. To alleviate this problem, we proposed to use a combination of simple performance indicators such as the maximum and the mean link utilization, obtaining a robust routing configuration that definitely outperforms current implementations from a global end-to-end perspective, while achieving very similar results as regards worst-case link utilization.

Combining the volume anomaly localization algorithm with our QoS-based extension for Robust Routing, and using the notions of load-balancing, we develop a novel algorithm to adapt routing in the event of volume anomalies, reducing the impacts of such traffic variations in the global performance of the network. The Reactive Robust Load Balancing (RRLB) introduces a dynamic approach to deal with unexpected traffic events, balancing load between prestablished paths as soon as volume anomalies are detected and localized. The notion of uncertainty-set expansion

was introduced and applied to preemptively compute optimal routing configurations, adapted to single OD-flow volume anomalies. By using load-balancing instead of routing reconfiguration, our algorithm ensures routing stability, even in the event of such strong traffic variations.

We also investigated the Dynamic Load-Balancing (DLB) paradigm, particularly analyzing its behavior when faced with volume anomalies. We showed that the different DLB algorithms proposed so far in the literature present important transient performance degradations in the process of adapting traffic balancing fractions, justifying the reluctance of network operators against these techniques. The RRLB algorithm does not suffer from this problem, basically because the load balancing fractions are pre-computed off-line in a robust basis, taking advantage of the goodness of the Robust Routing paradigm. Nevertheless, we developed novel algorithms to alleviate this and some other shortcomings detected in current DLB algorithms, applying no-regret algorithms.

Finally, we performed a comparative analysis between the proposed DLB algorithms and the RRLB approach, both under normal operation traffic and facing volume anomalies. From this study we concluded that the additional complexity involved in DLB is not justified when traffic variability is not very significant. In the case of large volume anomalies, DLB algorithms generally provide better results than RRLB after convergence, but they present an undesirable transient behavior. The use of DLB becomes appealing when volume anomalies are difficult to localize. Indeed, our RRLB algorithm assumes that volume anomalies occur in single OD-flows, but the case of multiple anomalous OD-flows is beyond the scope of the underlying anomaly localization algorithm.

Taking together the ensemble of developed TM models for anomaly-free traffic, the optimal network-wide anomaly detection and localization methods, and the reactive robust load balancing algorithm, this thesis work offers a complete solution for network operators to efficiently monitor large-scale networks and provide accurate QoS-based performance, even in the event of volume anomalies.

Perspectives

Despite being well-known research fields, the different research subjects covered in this work are far from being dead-ends. Below we shall describe different continuations for the research carried-out in this thesis, some of them being part of our ongoing works:

(1) We have analyzed the Traffic Matrix at the intradomain level, which permits to address many different Traffic Engineering (TE) problems and QoS issues at the interior of the network. Different studies have analyzed the Traffic Matrix at the interdomain level [125], evaluating the feasibility of interdomain TE. However, the problem of QoS provisioning at the interdomain level is still under discussion, and no general solutions have been yet provided to established end-to-end services with QoS guarantees. Combining the notions of intradomain and interdomain Traffic Matrix analysis could be highly beneficial to improve this issue. The recently launched FP7 (Seventh Framework Programme) European project entitled "Economics and Technologies for Inter-Carrier Services", in which we participate, will certainly draw on this idea.

(2) We have studied the Robust Routing paradigm at the intradomain level, addressing the problem of routing optimization under intradomain traffic uncertainty. However, the notions of Robust Routing can be used to optimize routing in the case of intradomain topology uncertainty, particularly addressing problems of multiple node and/or multiple link failures.

(3) Robust routing can also be directly applied to interdomain routing optimization, associating the concept of "uncertainty" to different components involved in the optimization. For example, we could assume that the announced BGP routing information is not exact but that presents some level of incertitude, building interdomain routing configurations that are robust against routes oscillation. The uncertainty could also be assigned to describe the way information is shared between different ASes, allowing to optimize routing configurations even if each AS shares partial information about its condition.

(4) The framework of Aggregated Objective Functions (AOF) that was used in our implementations provides interesting results as regards Multi Objective Optimization (MOO) in the context of robust optimization. An AOF approach can be used to construct better objective functions from simple performance indicators, avoiding the need of more complex MOO techniques. An interesting perspective in this direction is to study the trade-off in using a simple AOF approach against a more complex but more complete MOO approach, computing all Pareto-efficient solutions and comparing their performance. This is in fact part of an ongoing work performed by the research group of Assistant Professor Hervé Kerivin (Clemson University), with whom be have developed strong collaboration ties.

(5) The optimal volume anomaly detection algorithms that were developed in this work are threshold-based algorithms, which makes it difficult to simultaneously detect large volume anomalies and small problems. We are currently investigating an extension to our algorithms, developing a two-stages anomaly detection approach to eliminate false negatives while drastically reducing false positives. In the first stage, detection thresholds are set low enough so as to achieve zero false negative rates. In the second stage, traffic is analyzed at a lower level of aggregation, in order to correctly discriminate between real anomalies and false positives.

(6) The techniques that we have developed for anomaly detection and localization can be easily extended to detect and localize more general traffic anomalies, provided that a statistical parametric model for the underlying data is available.

(7) In this work we have focused on supervised anomaly detection, building a model for anomaly-free traffic that must be calibrated with "clean" data. We are currently trying to use our detection techniques for non-supervised anomaly detection, using stream clustering techniques to avoid the need of anomaly-free traffic measurements.

List of Publications

International Journals & Book Chapters:

1. [J_1] **P. Casas**, S. Vaton, L. Fillatre, and I. Nikiforov, "Optimal Volume Anomaly Detection and Isolation in Large-Scale IP Networks using Coarse-Grained Measurements", in *Computer Networks*, vol. 54 (11), pp. 1750-1766, (doi:10.1016/j.comnet.2010.01.013), Elsevier, 2010.

2. [J_2] **P. Casas**, F. Larroca, J. L. Rougier, and S. Vaton, "Taming Traffic Dynamics: Analysis and Improvements", in *Computer Communications*, (doi:10.1016/j.comcom.2010.07.009), Elsevier, 2010.

3. [J_3] **P. Casas**, L. Fillatre, S. Vaton, and I. Nikiforov, "Reactive Robust Routing: Anomaly Localization and Routing Reconfiguration for Dynamic Networks", in *Journal of the Network and Systems Management*, ISSN: 1064-7570, Springer, 2010 (Accepted for Publication).

4. [B_1] **P. Casas**, L. Fillatre, S. Vaton, and I. Nikiforov, "Volume Anomaly Detection in Data Networks: an Optimal Detection Algorithm vs the PCA Approach", in *Traffic Management and Traffic Engineering for the Future Internet*, Lecture Notes in Computer Science Series, vol. 5464, pp. 96-113, ISBN: 978-3-642-04575-2, (doi:10.1007/978-3-642-04576-9_7), Springer, 2009.

International Conferences:

1. [C_1] **P. Casas** and S. Vaton, "On the Use of Random Neural Networks for Traffic Matrix Estimation in Large-Scale IP Networks", in *TRAC'10: 1st International Workshop on Traffic Analysis and Classification*, Caen, France, 2010.

2. [C_2] **P. Casas**, S. Vaton, L. Fillatre, and T. Chonavel, "Efficient Methods for Traffic Matrix Modeling and On-line Estimation in Large-Scale IP Networks", in *ITC'21: 21st conference on International Teletraffic Congress*, Paris, France, 2009.

3. [C_3] **P. Casas**, F. Larroca, J. L. Rougier, and S. Vaton, "Robust Routing vs Dynamic Load Balancing: a Comprehensive Study and New Directions", in *DRCN'09: 7th International Workshop on the Design of Reliable Communication Networks*, Washington D.C., United States, 2009.

4. [C_4] **P. Casas**, F. Larroca, and S. Vaton, "Robust Routing Mechanisms for Intradomain Traffic Engineering in Dynamic Networks", in *LANOMS'09: 6th Latin-American Network Operations and Management Symposium*, Punta Del Este, Uruguay, 2009.

5. [C_5] **P. Casas**, L. Fillatre, S. Vaton, and I. Nikiforov, "Volume Anomaly Detection in Data Networks: an Optimal Detection Algorithm vs the PCA Approach", in *FITraMEn'08: 1st Euro-NF Workshop*, Porto, Portugal, 2008.

6. [C_6] **P. Casas**, L. Fillatre, and S. Vaton, "Multi Hour Robust Routing and Fast Load Change Detection for Traffic Engineering", in *ICC'08: IEEE International Conference on Communications*, Beijing, China, 2008.

7. [C_7] **P. Casas**, L. Fillatre, and S. Vaton, "Robust and Reactive Traffic Engineering for Dynamic Traffic Demands", in *NGI'08: 4th EuroNGI Conference on Next Generation Internet Networks*, Krakow, Poland, 2008.

8. [C_8] **P. Casas**, L. Fillatre, and S. Vaton, "Reliable Routing for the Next Generation Network", in *CQR'08: IEEE Communications Society International Communications Quality and Reliability Workshop*, Arizona, United States, 2008 (Accepted Paper).

9. [C_9] L. Fillatre, I. Nikiforov, **P. Casas**, and S. Vaton, "Optimal Volume Anomaly Detection in Network Traffic Flows", in *EUSIPCO'08: 16th European Signal Processing Conference*, Lausanne, Switzerland, 2008.

10. [C_{10}] L. Fillatre, I. Nikiforov, S. Vaton, and **P. Casas**, "Sequential non Bayesian Network Traffic Flows Anomaly Detection and Isolation", in *IWAP'08: International Workshop on Applied Probability*, Compiegne, France, 2008.

11. [C_{11}] **P. Casas** and S. Vaton, "An Adaptive Multi-Temporal Approach for Robust Routing", in *2nd EuroNGI Workshop on IP QoS and Traffic Control*, Lisbon, Portugal, 2007.

Distinctions of the Thesis

1. **Best Conference Paper Award** for the paper entitled "Robust Routing Mechanisms for Intradomain Traffic Engineering in Dynamic Networks", presented at *LANOMS'09: 6th Latin-American Network Operations and Management Symposium* in Punta Del Este, Uruguay, in October 2009. The authors of the paper are **P. Casas**, F. Larroca, and S. Vaton.

2. **Best Conference Paper Award** for the paper entitled "Robust and Reactive Traffic Engineering for Dynamic Traffic Demands", presented at *NGI'08: 4th EuroNGI Conference on Next Generation Internet Networks* in Krakow, Poland, in April 2008. The authors of the paper are **P. Casas**, L. Fillatre, and S. Vaton.

Bibliography

[1] FTTH Council, "FTTH Fiber Primer", white paper available at `http://www.ftthcouncil.org`, 2009.

[2] KMI Corporation, "Worldwide Markets for Fiberoptics in Broadband Access Networks: Fiberoptic Demand Forecast in Broadband Access Networks", available at `http://cruonline.crugroup.com/WireandCable/MarketForecasts`, 2006.

[3] Cisco Systems, "Global IP Traffic Forecast and Methodology, 2006-2011", white paper available at `http://www.cisco.com`, 2007 - updated 2008.

[4] Cisco Systems, "The Exabyte Era", white paper available at `http://www.cisco.com`, 2007 - updated 2008.

[5] S .Shakkottai, M .Fomenkov, R .Koga, D .Krioukov, and K .Claffy, "Evolution of the Internet AS-Level Ecosystem", in *International Conference on Complex Sciences, Complex*, 2009.

[6] G. Combs, "Wireshark, a Network Protocol Analyzer", `http://www.wireshark.org`.

[7] M. Roesch, "SNORT, Network Intrusion Detection and Prevention", `http://www.snort.org`.

[8] J. Levandoski, E. Sommer, and M. Strait, "L7-Filter, Application Layer Packet Classifier for Linux", `http://l7-filter.sourceforge.net/`.

[9] Cisco Systems, "NetFlow: Cisco Network Traffic Monitoring and Management Protocol", `http://www.cisco.com`, 2000.

[10] Internet Engineering Task Force, "RFC 5101 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", 2008.

[11] Internet Engineering Task Force, "RFC 5102 - Information Model for IP Flow Information Export", 2008.

[12] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a Better NetFlow", in *Proc. ACM SIGCOMM*, 2004.

[13] I. Cunha, F. Silveira, R. Oliveira, R. Teixeira, and C. Diot, "Uncovering Artifacts of Flow Measurement Tools", in *Proc. of Passive and Active Measurement Conference*, 2009.

[14] Cisco Systems, "Virtualization Beyond the Data Center: Increase Network Infrastructure Utilization and Efficiency to Reduce Operational Costs", white paper available at `http://www.cisco.com`, 2009.

[15] D. Medhi and D. Tipper, "Some Approaches to Solving a Multi-Hour Broadband Network Capacity Design Problem with Single-Path Routing", in *Telecommunication Systems*, vol. 13, pp. 269-271, 2000.

[16] D. Medhi, "Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-based Dynamically Reconfigurable Wide-Area ATM Networks", in *IEEE/ACM Trans. on Networking*, vol. 3, pp. 809-818, 1995.

[17] A. Dutta, "Capacity Planning of Private Networks using DCS under Multi-busy-hour Traffic", in *IEEE Trans. on Comm.*, vol 42, pp. 2371-2374, 1994.

[18] M. Eisenberg, "Engineering Traffic Networks for More than One Busy Hour", in *Bell Systems Technical J.*, vol. 56, pp. 1-20, 1977.

[19] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot, "A taxonomy of IP traffic matrices", in *SPIE ITCOM: Scalability and Traffic Control in IP Networks II*, 2002.

[20] Internet Engineering Task Force, "RFC 1930 - Guidelines for creation, selection, and registration of an Autonomous System (AS)", 1996.

[21] Y. Zhang, M. Roughan, N. Duffield and A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Load Measurements", in *Proc. ACM SIGMETRICS*, 2003.

[22] A. Soule, K. Salamatian, A. Nucci, and N. Taft,"Traffic Matrix Tracking using Kalman Filters", in *LSNI*, 2005.

[23] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot,"Traffic Matrices: Balancing Measurements, Inference, and Modeling", in *Proc. ACM SIGMETRICS*, 2005.

[24] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet Tomography", in *IEEE Signal Processing Mag.*, 2002.

[25] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data", in *J. Amer. Statist. Assoc*, 91, pp. 365-377, 1996.

[26] C. Tebaldi et al, "Bayesian Inference on Network Traffic Using Link Count Data", in *J. Amer. Statist. Assoc*, 93, pp. 557-576, 1998.

[27] J. Cao et al, "Time-Varying Network Tomography", in *J. Amer. Statist. Assoc*, 95, pp. 1063-1075, 2000.

[28] S. Vaton and A. Gravey, "Network Tomography : an Iterative Bayesian Analysis", in *Proc. ITC 18*, 2003.

[29] P. Bermolen, S. Vaton, and I. Juva, "Search for Optimality in Traffic Matrix Estimation: A Rational Approach by Cramér-Rao Lower Bounds", in *Proc. NGI'06*, 2006.

[30] I. Juva, S. Vaton, and J. Virtamo, "Quick Traffic Matrix Estimation Based on Link Count Covariances", in *Proc. IEEE-ICC'06*, 2006.

[31] S. Vaton and J. Bedo, "Network Traffic Matrix: How can One Learn the Prior Distributions from the Link Counts Only?", in *Proc. IEEE-ICC'04*, 2004.

[32] S. Vaton, J. Bedo, and A. Gravey, "Advanced Methods for the Estimation of the Origin Destination Traffic Matrix", in *Performance Evaluation and Planning Methods for the Next Generation Internet*, 25th Anniversary of GERARD, Springer, 2005.

[33] A. Medina, K. Salamatian, S. Bhattacharyya and C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions", in *Proc. ACM SIGCOMM*, 2002.

[34] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford and F. True, "Deriving Traffic Demands for Operational IP Networks", in *IEEE/ACM Trans. on Networking*, vol. 9, no. 3, pp. 265-279, 2001.

[35] J. Kowalski and B. Warfield, "Modeling Traffic Demand Between Nodes in a Telecommunications Network", in *ATNAC*, 1995.

[36] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in Measuring Backbone Traffic Variability: Models, Metrics, Measurements and Meaning", in *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[37] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural Analysis of Network Traffic Flows", in *Proc. ACM SIGMETICS*, 2004.

[38] K. Papagiannaki, N. Taft, and A. Lakhina, "A Distributed Approach to Measure Traffic Matrices", in *Proc. ACM IMC*, 2004.

[39] D. Jiang and G. Hu, "Large-Scale IP Traffic Matrix Estimation Based in Backpropagation Neural Network", in *Proc. IEEE ICINIS*, 2008.

[40] C. Cramer, E. Gelenbe, and P. Gelenbe, "Image and Video Compression", in *IEEE Potentials*, 1998.

[41] A. H. Abdelbaki and E. Gelenbe, "Random Neural Network Decoder for Error Correcting Codes", in *Int. Joint Conf. Neural Networks*, vol. 5, 1999.

[42] E. Gelenbe, H. Bakircioglu, and T. Kocak, "Image processing with the random neural network", in *Proc. 13th Int. Conf. Digital Signal Processing*, vol. 1, pp. 243-248, 1997.

[43] H. Bakircioglu and T. Kocak, "Survey of random neural network applications", in *Eur. J. Oper. Res.*, vol. 126, no. 2, pp. 319-330, 2000.

[44] S. Mohamed and G. Rubino, "A Study of Real-Time Packet Video Quality Using Random Neural Networks", in *Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1071-1083, 2000.

[45] C. M. Bishop, "Neural Networks for Pattern Recognition", *Oxford University Press*, 1995.

[46] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution", in *Neural Computation*, vol. 1, pp. 502-511, 1989.

[47] E. Gelenbe, "Stability of the Random Neural Networks Model", in *Neural Computation*, vol. 2, pp. 239-247, 1990.

[48] E. Gelenbe, "Learning in the Recurrent Random Neural Networks", in *Neural Computation*, vol. 5, pp. 154-164, 1993.

[49] S. Timotheou, "A Novel Weight Initialization Method for the Random Neural Network", in *Neurocomputing*, vol. 73, no. 1-3, pp. 160-168, 2009.

[50] R. Duda, P. Hart, and D. Stork, "Pattern Classification (2nd Edition)", Wiley-Interscience, 2000.

[51] M. Hayes, "Statistical Digital Signal Processing and Modeling", J. Wiley & Sons, 1996.

[52] S. Kay, "Modern Spectral Estimation: Theory and Applications", Prentice-Hall, 1988.

[53] C. Hood and C. Ji, "Proactive network fault detection", in *Proc. IEEE INFOCOM*, 1997.

[54] I. Katzela and M. Schwartz, "Schemes for fault identification in communications networks", in *IEEE/ACM Trans. on Networking*, vol. 3, no. 6, pp. 753-764, 1995.

[55] A. Ward, P. Glynn and K. Richardson, "Internet service performance failure detection", in *Performance Evaluation Review*, 1998.

[56] G. Iannaccone, C. Chuah, S. Bhattacharyya, and C. Diot, "Feasibility of IP Restoration in a Tier-1 Backbone", in *IEEE Network Magazine*, vol. 18, no. 2, pp. 13-19, 2004.

[57] A. Fumagalli and L. Valcarenghi, "IP restoration versus WDM protection: Is there an optimal choice?", in *IEEE Network Magazine*, vol. 14, no. 6, pp. 34-41, 2000.

[58] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Fault management in IP-over-WDM networks: WDM protection versus IP restoration", in *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, 2002.

[59] J. Jung, B. Krishnamurthy and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and webs", in *WWW-02*, 2002.

[60] L. Xie et al., "From Detection to Remediation: A Self-Organized System for Addressing Flash Crowd Problems", in *Proc. IEEE ICC-08*, 2008.

[61] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies", in *SIGCOMM Internet Measurement Workshop*, 2002.

[62] J. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring", in *Proc. 14th Systems Administration Conference*, 2000.

[63] C.M. Cheng, H. Kung, K.S. Tan, "Use of Spectral Analysis in Defense Against DoS Attacks", in *Proc. IEEE GLOBECOM*, 2002.

[64] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based Change Detection: Methods, Evaluation, and Applications", in *Proc. USENIX/ACM IMC*, 2003.

[65] C.C. Zou, W. Gong, D. Towsley and L. Gao "The Monitoring and Early Detection of Internet Worms", in *IEEE/ACM Trans. on Networking*, vol. 13, no. 5, pp. 961-974, 2005.

[66] H. Wang, D. Zhang and K.Shin, "Detecting SYN flooding attacks", in *Proc. IEEE INFOCOM*, 2002.

[67] M. Roughan, T. Griffin, M. Mao, A. Greenberg, and B. Freeman, "Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies", in *ACM SIGCOMM NeTs Workshop*, 2004.

[68] R. Dunia and S. J. Qin, "Multi-Dimensional Fault Diagnosis Using a Subspace Approach", in *Proc. American Control Conference*, 1997.

[69] R. Dunia and S. J. Qin, "A Subspace Approach to Multidimensional Fault Identification and Reconstruction", in *American Institute of Chemical Engineers Journal*, pp. 1813-1831, 1998.

[70] J. E. Jackson and G. S. Mudholkar, "Control Procedures for Residuals Associated with Principal Component Analysis", in *Technometrics*, pp. 341-349, 1979.

[71] A. Lakhina, M. Crovella, and C. Diot, "Characterization of Network-Wide Anomalies in Traffic Flows", in *Proc. USENIX/ACM IMC*, 2004.

[72] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies", in *Proc. ACM SIGCOMM*, 2004.

[73] A. Lakhina, M. Crovella and C. Diot, "Mining Anomalies Using Traffic Feature Distributions", in *Proc. ACM SIGCOMM*, 2005.

[74] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone and A. Lakhina, "Detection and Identification of Network Anomalies Using Sketch Subspaces", in *Proc. USENIX/ACM IMC*, 2006.

[75] T. Ahmed, M. Coates, and A. Lakhina, "Multivariate Online Anomaly Detection Using Kernel Recursive Least Squares," in *Proc. IEEE Infocom*, 2007.

[76] H. Ringberg, A. Soule, J. Rexford and C. Diot, "Sensitivity of PCA for Traffic Anomaly Detection", in *Proc. ACM SIGMETRICS*, 2007.

[77] M. Thottan and C. Ji, "Anomaly Detection in IP Networks", in *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2191-2204, 2003.

[78] A. Soule, K. Salamatian and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection", in *Proc. USENIX/ACM IMC*, 2005.

[79] A. Tartakovsky *et al.*, "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods", in *IEEE Trans. on Signal Processing*, vol. 54, no. 9, pp. 3372-3382, 2006.

[80] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network Anomography", in *Proc. USENIX/ACM IMC*, 2005.

[81] G. Nürnberger, "Approximation by spline functions", Springer-Verlag, 1989.

[82] C. Rao, "Linear Statistical Inference and its Applications", John Wiley & Sons, 1973.

[83] E. Lehman, "Testing Statistical Hypotheses, Second Edition", Chapman & Hall, 1986.

[84] A. Wald, "Tests of statistical hypotheses concerning several parameters when the number of observations is large", in *Trans. American Math. Soc.*, vol. 54, pp. 426-482, 1943.

[85] L. Fillatre and I. Nikiforov, "Non-bayesian detection and detectability of anomalies from a few noisy tomographic projections", in *IEEE Trans. on Signal Processing*, vol. 55, no. 2, pp. 401-413, 2007.

[86] M. Basseville and I. Nikiforov, "Detection of abrupt changes: theory and applications", Prentice Hall, 1993.

[87] I. Nikiforov, "A generalized change detection problem", in *IEEE Trans. on IT*, vol. 41, pp. 171-187, 1995.

[88] T. Oskiper and H. Poor, "Online activity detection in a multiuser environment using the matrix CUSUM algorithm", in *IEEE Trans. on IT*, vol. 48, pp. 477-493, 2002.

[89] I. Nikiforov, "A simple recursive algorithm for diagnosis of abrupt changes in random signals", in *IEEE Trans. on IT*, vol. 46, no. 7, pp. 2740-2746, 2000.

[90] I. Nikiforov, "A lower bound for the detection/localization delay in a class of sequential tests", in *IEEE Trans. on IT*, vol. 49, no. 11, pp. 3037-3046, 2003.

[91] M. Fouladirad and I. Nikiforov, "Optimal statistical fault detection with nuisance parameters", in *Automatica*,vol 41, pp. 1157-1171, 2005.

[92] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network Flows: Theory, Algorithms, and Applications", *Prentice Hall*, 1993.

[93] A. Girard, "Routing and Dimensioning in Circuit-Switched Networks", in *Addison-Wesley (Reading, Mass.)*, 1990.

[94] M. Pioro and D. Medhi, "Routing, Flow, and Capacity Design in Communication and Computer Networks", *Elsevier/Morgan Kaufmann*, 2004.

[95] M. Resende and P. Pardalos, "Handbook of Optimization in Telecommunications", *Spinger Science/Business Media*, 2006.

[96] D. Mitra and K. G. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Netorks", in *GLOBECOM '99*, 1999.

[97] M. Roughan, M. Thorup, and Y. Zhang, "Traffic Engineering with Estimated Traffic Matrices", in *IMC '03*, 2003.

[98] C. Zhang, Z. Ge, J. Kurose, Y. Liu and D. Towsley, "Optimal Routing with Multiple Traffic Matrices: Tradeoff between Average case and Worst case Performance", in *Proc. 13th International Conference on Network Protocols (ICNP)*, 2005.

[99] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll and D. Towsley, "On Optimal Routing with Multiple Traffic Matrices", in *Proc. INFOCOM*, 2005.

[100] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. van der Merwe, "A Flexible Model for Resource Management in Virtual Private Networks", in *SIGCOMM '99*, 1999.

[101] W. Ben-Ameur and H. Kerivin, "Routing of Uncertain Traffic Demands", *Optimization and Engineering*, vol. 6, pp. 283-313, 2005.

[102] D. Applegate and E. Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs", in *SIGCOMM '03*, 2003.

[103] M. Johansson and A. Gunnar, "Data-driven Traffic Engineering: techniques, experiences and challenges", in *BROADNETS '06*, 2006.

[104] H. Wang, H. Xie, L. Qiu, Y. Yang, Y. Zhang, and A. Greenberg, "COPE: Traffic Engineering in Dynamic Networks", in *SIGCOMM '06*, 2006.

[105] M. Kodialam, T. Lakshman, and S. Sengupta, "Efficient and Robust Routing of Highly Variable Traffic", in *HOT-NETS-III*, 2004.

[106] A. Gunnar and M. Johansson, "Robust Routing Under BGP Reroutes", in *GLOBECOM '07*, 2007.

[107] M. Kodialam, T. Lakshman, J. Orlin, and S. Sengupta, "Oblivious Routing of Highly Variable Traffic in Service Overlays and IP Backbones", in *IEEE Trans. on Networking*, vol. 17, pp. 459-472, 2009.

[108] I. Juva, "Robust Load Balancing", in *GLOBECOM '07*, 2007.

[109] W. Ben-Ameur, "Between Fully Dynamic Routing and Robust Stable Routing", in *DRCN '07*, 2007.

[110] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", in *INFOCOM '01*, 2001.

[111] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the Tightrope: Responsive yet Stable Traffic Engineering", in *SIGCOMM '05*, 2005.

[112] S. Fischer, N. Kammenhuber, and A. Feldmann, "REPLEX: dynamic traffic engineering based on wardrop routing policies", in *CoNEXT '06*, 2006.

[113] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez and L. Wynter, "A survey on networking games in telecommunications", *Comput. Oper. Res.*, vol. 33, no. 2, pp. 286-311, 2006.

[114] F. Larroca and J.L. Rougier, "Minimum-Delay Load-Balancing Through Non-Parametric Regression", in *Networking '09*, 2009.

[115] J. Wardrop, "Some theoretical aspects of road traffic research", in *Proceedings of the Institution of Civil Engineers, Part II*, vol. 1, no. 36, pp. 352-362, 1952.

[116] R. Banner and A. Orda, "Bottleneck Routing Games in Communication Networks", in *IEEE Journal on Selected Areas in Comm.*, vol. 25, no. 6, pp. 1173-1179, 2007.

[117] F. Larroca and J.L. Rougier, "Routing Games for Traffic Engineering", in *ICC '09*.

[118] A. Blum, E. Even-Dar, and K. Ligett, "Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games", in *PODC '06*.

[119] R. Yaroshinsky, R. El-Yaniv, and S. S. Seiden, "How to Better Use Expert Advice", in *Mach. Learn.*, vol. 55, no. 3, pp. 271-309, 2004.

[120] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm", in *Inf. Comput.*, vol. 108, no. 2, pp. 212-261, 1994.

[121] L. Kleinrock, "Queueing Systems", *Wiley-Interscience*, 1975.

[122] J. Evans and R. Steuer, "A Revised Simplex Method For Linear Multiple Objective Programs", in *Mathematical Programming*, vol. 5, no. 1, pp. 54-72, 1973.

[123] J. Ecker and I. Kouada, "Finding All Efficient Extreme Points for Multiple Objective Linear Programming Programs", in *Mathematical Programming*, vol. 14, pp. 249-261, 1978.

[124] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing Public Intradomain Traffic Matrices to the Research Community", in *ACM Sigcomm Computer Communication Review*, 2006.

[125] S. Uhlig, "Implications of Traffic Characteristics on Interdomain Traffic Engineering", Ph.D. thesis, 2004.

[126] K. Cho, "WIDE-TRANSIT 150 Megabit Ethernet Trace 2008-03-18", `http://mawi.wide.ad.jp/mawi/samplepoint-F/20080318/`

[127] Cooperative Association for Internet Data Analysis, `http://www.caida.org/`

[128] Abilene Observatory, `http://abilene.internet2.edu/observatory/`

[129] Y. Zhang, "Abilene Data Set 2004", `http://www.cs.utexas.edu/~yzhang/`

[130] TOTEM, a Toolbox for Traffic Engineering Methods, `http://totem.run.montefiore.ulg.ac.be/`

Analyse Statistique de Trafic Réseau pour la Détection d'Anomalies et la Qualité de Service

Internet, le propulseur principal de l'actuelle "ère de l'information", est devenu un des acteurs principaux de notre société. Internet est actuellement le composant fondamental dans l'infrastructure de communication globale, en exerçant un rôle crucial dans l'éducation, l'économie, le divertissement, et la vie sociale de nos nations. Sa croissance extraordinaire et incontrôlable partout dans le monde dans le dernière décennie a permis le développement de milliers de compagnies qui produisent et "submergent" ses contenus dans l'Internet. Cette prolifération de contenus a été suivi par une croissance soutenue de consommateurs, en portant à une explosion du trafic présent dans les réseaux des données qui conforment l'Internet. Cette explosion du tafic réseau n'est pas seulement une croissance au niveau de volume de trafic, mais également quant à l'hétérogénéité et complexité de composition du trafic.

Après un brève retombée vers le milieu de cette décennie, les acteurs plus importants de l'Internet pronostiquent que le trafic de réseau multipliera par deux son volume presque chaque deux ans dans le futur proche, propulsé par la pénétration du vidéo de haute définition et de l'accès de très haute vitesse. Il est attendu que le trafic IP total croîtra de 6.6 exabytes par mois en 2007 à presque 29 exabytes par mois en 2011 (1 exabyte = 10^{18} bytes), en quadruplant son volume en moins de 5 années.

Au même temps, le développement des réseaux optiques et l'évolution des technologies d'accès, notamment la technologie FTTH (Fiber To The Home) augmenteront dramatiquement le bande passante pour les utilisateurs finaux, en imposant des problèmes sérieux et imprévus dans les réseaux de backbone, supposés de capacité infinie jusqu'à présent. L'industrie de la technologie FTTH pronostique une demande de bande passante de jusqu'à 30 Gbps par utilisateur dans l'année 2030 [1, 2]. La figure 4.28 présente l'évolution prospective du trafic Internet et de la bande passante de très haute vitesse pour les prochaines 2 ou 3 années.

Ce futur proche impose de nouveaux défis pour les opérateurs de réseaux de grande échelle, ceux qui sont au bout du compte les responsables de soutenir la croissance de l'Internet dans une grande mesure. Les usagers de l'Internet veulent une Internet plus rapide, plus sûre, et avec de meilleures prestations de qualité de service, et l'analyse et la surveillance du trafic que circule le réseau est probablement la solution la plus efficace et à portée de la main pour les opérateurs de réseau. Connaître et comprendre le trafic qui circule le réseau est crucial pour la conception efficace, le fonctionnement correct, et l'ingénierie des services offerts sur l'Inernet.

La surveillance du trafic réseau est sans aucun doute une des tâches critiques pour les opérateurs de réseau qui sera sérieusement affecté par ce fort développement. En effet, capturer et analyser de grands volumes de trafic hétérogène et dans des multiples points du réseau peut-il résulter extrêmement coûteux. Dans les débuts de

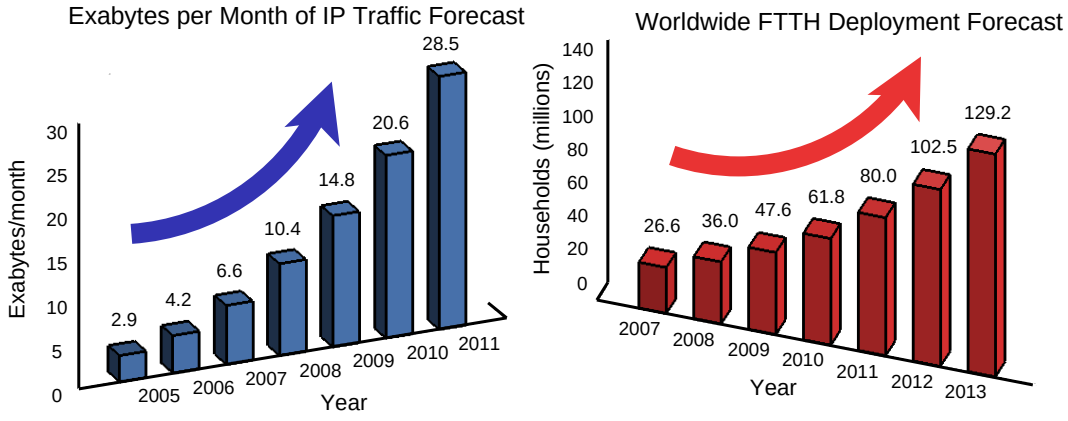


Figure 4.23 — Perspectives de croissance du trafic Internet et du déploiement de la technologie FTTH pour les prochaines années.

l'Internet, la surveillance du réseau était plus un art qui une science, en dépendant de l'expérience et la connaissance de l'opérateur de réseau pour analyser le trafic "à la main". Toutefois, la complexité croissante et la taille des réseaux de fournisseurs de services Internet a motivé le développement de systèmes automatiques de surveillance à grande échelle dans les dernières années.

La surveillance du trafic réseau peut rehausser diverses activités en rapport avec la gestion du réseau, comme la caractérisation et la classification du trafic, l'identification de défaillances ou de problèmes de dégradation de fonctionnement, et la détection de trafic malicieux (i.e., attaques de réseau). Il est justement possible d'apprécier la grande quantité et variété de Systèmes de Détection d'Intrusions (IDS) qui ont donné au réseau des nouvelles capacités pour gérer le trafic malicieux.

Le processus de surveillance du trafic consiste en trois tâches consécutives: la capture des données, l'analyse de ces données, et la décision extraite de cette analyse. Chacune de ces tâches est rendu plus difficile dans la scène actuelle du trafic de réseau. La "récolte" de données est très coûteuse, parce qu'il y a trop de trafic à capturer et dans diverses parties du réseau. L'analyse des données devient plus complexe, parce que le trafic est plus hétérogène et les possibles défaillances qu'on peut trouver sont plus variées. Une décision correcte est beaucoup plus critique qu'avant, parce que les services proportionnés dans l'Internet actuel sont plus vitaux que dans le passé.

Un autre sujet en rapport avec la surveillance du trafic réseau de haute vitesse et à une grande échelle dans un futur proche est celui de la rentabilité des systèmes de surveillance. Les nouveaux modèles d'affaire dans l'Internet des dernières années ont donné lieu à beaucoup de solutions de virtualización de réseau [14], en permettant que de petits fournisseurs de service capturent une partie du marché d'Internet avec des investissements très réduits en infrastructure de réseau. Ceci a conduit aux opérateurs de réseau à réduire radicalement ses investissements en infrastructure, en cherchant des solutions qui permettent de sortir le plus grand bénéfice de la technologie disponible

dans leurs réseaux actuelles. Les systèmes de surveillance du futur doivent alors viser à l'analyse du trafic global du réseau à travers de mesures limitées de trafic, en utilisant de l'information "moins chère" et de basse résolution, et des procédures d'inférence et algorithmes intelligents d'analyse pour réduire des coûts de mesure de trafic, sans réduire la performance du processus de surveillance.

Où surveiller le Trafic du Réseau?

Malgré la croissance massive de l'Internet des dernières années, sa structure globale est encore fortement hiérarchique. L'épine dorsale de l'Internet est composée d'un nombre réduit de grands Systèmes Autonomes (ASes) [5], connus comme réseaux du type Tier-1. En gros, un AS est une collection de préfixes de routage IP qui partagent une même politique de routage vers Internet et qui sont sous le contrôle d'un même opérateur de réseau [20]. Une liste non exhaustive des réseaux Tier-1 actuels inclut AT&T, Global Crossing, Level 3 Communications, NTT Communications, Sprint, Tata Communications, Verizon Business (UUNET), Savvis, TeliaNet, Bell Canada, et XO Communications (XOXO). La figure 4.29 présente une image de la structure actuelle de l'Internet selon CAIDA [127].

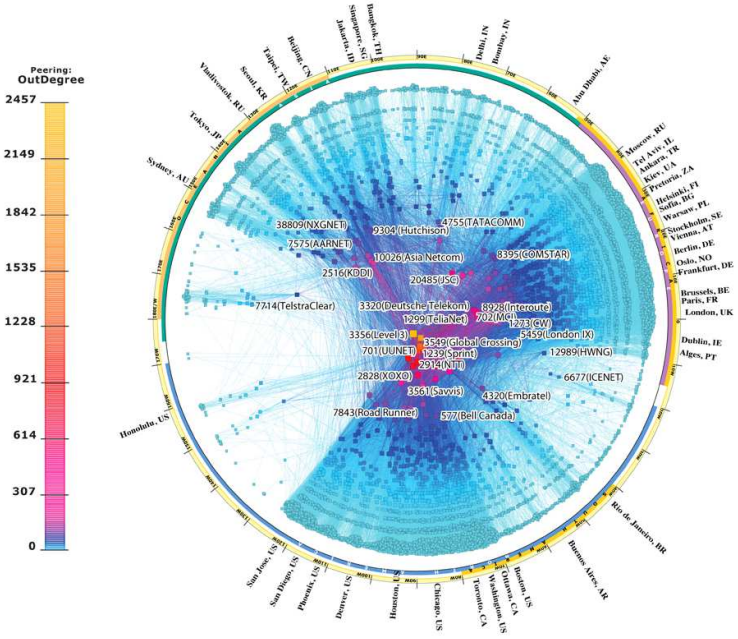


Figure 4.24 — Topologie IPv4 de l'Internet en janvier 2009.

Les réseaux Tier-1 fournissent de la connectivité globale dans Internet et représentent le premier niveau dans la hiérarchie. Les niveaux suivants dans la hiérarchie d'Internet sont composés de ASes plus petits et moins interconnectés, connus comme réseaux du types Tier-2 et Tier-3. Quelques exemples de réseaux Tier-2 sont le réseau de télécommunications de l'entreprise allemande Deutsche Telecom, le réseau de British Telecom, et le réseau de France Telecom entre autres. Finalement,

les marges de l'Internet sont composées par des ASes terminaux, habituellement appelés stub ASes.

Cette division en des Systèmes Autonomes donne deux visions structurelles différentes de l'Internet : l'Internet intra-domaine et l'Internet inter-domaine [125]. L'Internet intra-domaine est composé des routers interconnectés dans chaque AS individuel et qui échangent trafic entre eux en utlisant un protocole de routage intra-domaine. Chaque AS a une topologie de réseau propre et qui est parfaitement connu par l'opérateur de réseau qui l'administre. D'autre part, l'Internet inter-domaine est composé des différents ASes et de ses interconnexions respectives. Les caractéristiques internes de chaque AS sont transparentes d'un point de vue inter-domaine, et le trafic est échangé entre ceux-ci par un protocole de routage inter-domaine.

La surveillance du trafic réseau est normalement effectué à l'échelle intra-domaine dans des réseaux de grande taille (i.e., réseaux Tier-1 et Tier-2), principalement parce que à cette échelle la topologie du réseau est parfaitement connue par l'opérateur de réseau. En même temps, c'est seulement à cette échelle qui le fournisseur a un contrôle complet sur le réseau, et par conséquent il peut la maîtriser sans restriction. L'analyse de trafic à niveau inter-domaine est une tâche longuement plus difficile, parce que l'information disponible est limitée, les différents opérateurs de réseau ne collaborent pas nécessairement entre eux, des questions de caractère privé et d'affaires limitent l'échange d'information entre des opérateurs, et beaucoup d'autres caractéristiques de l'échelle inter-domaine qui rendent très difficile la tâche de la surveillance du trafic.

Dans ce travail de thèse, nous nous avons centrés sur le problème de la surveillance et l'analyse du trafic réseau au niveau intra-domaine, pour deux raisons principalement. D'abord, le contrôle que nous pouvons avoir sur le réseau au niveau intra-domaine permet de proposer des solutions plus complètes, pas seulement en ce qui concerne l'analyse du trafic, mais aussi quant au processus de récupération face à des événements imprévus. Deuxièmement, la structure de l'Internet continue à être fortement concentrée dans un petit groupe des réseaux de grande taille, ce qui conduit au fait que la performance de l'Internet dans son ensemble dépend fortement de la performance individuelle de ces réseaux de grande taille.

Quelle Information À Surveiller?

Les opérateurs de réseau sont confrontés couramment avec un vaste spectre d'événements inhabituels qui attentent contre le fonctionnement correct de leurs réseaux. Un problème important associé à la détection de ces événements anormaux est que ses causes et origines peuvent varier considérablement. Différentes anomalies dans le réseau et/ou dans le trafic qui circule le réseau peuvent se déchaîner pour des causes très variées: défaillances d'équipements et erreurs de configuration, comportements inhabituels de d'un ou de plusieurs utilisateurs (e.g., événements du type flash crowd, des transferts de haut volume de trafic, etc.), modifications du routage

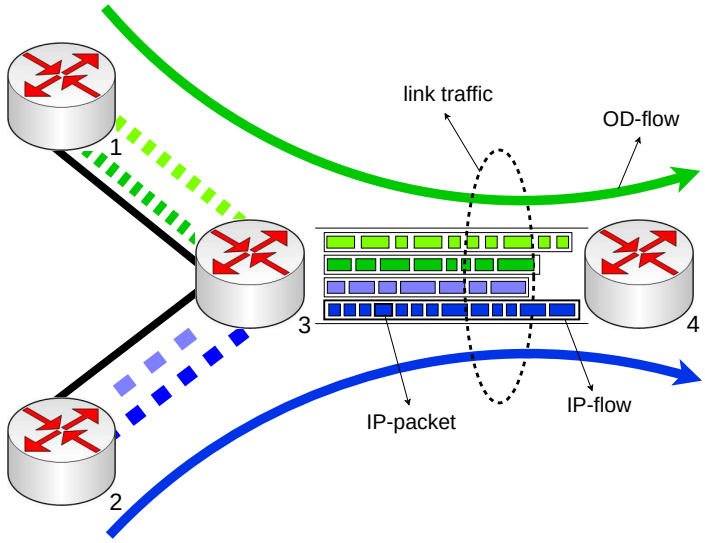


Figure 4.25 — Différents niveaux d'agrégation de trafic réseau.

extérieur à l'AS en question, attaques de réseau (e.g., attaques de type DoS/DDoS, scans de ports et de réseau, propagation de vers et virus, etc.), aussi bien que des nouveaux événements anormaux jamais vu avant.

Un défi important en rapport avec la détection d'anomalies dans le réseau est que celles-ci sont une cible "mobile". Il est inhéremment difficile de définir et de spécifier de façon permanente l'ensemble possible d'anomalies qui peuvent avoir un impact sur réseau, surtout dans le cas de trafic malicieux. De nouvelles anomalies apparaissent constantement, ce pourquoi les systèmes de détection d'anomalies doivent éviter d'être attachés à un ensemble pré-défini d'anomalies à détecter.

Différents types d'anomalies de réseau et de trafic peuvent être détectées, selon le type d'information surveillée et de leur niveau d'agrégation. En général, nous pouvons identifier quatre niveaux de base d'agrégation de trafic, en considérant la "résolution" de l'information que ce niveau d'agrégation apporte: paquet IP, flux IP, flux OD, et trafic de liaison réseau. La figure 4.30 nous aidera à expliquer cette classification. L'analyse de trafic au niveau de paquets IP apporte l'information de surveillance la plus riche et de haute résolution. En travaillant à ce niveau il est possible d'analyser les caractéristiques particulières de chaque paquet IP, en accédant inclusivement à sa charge utile. Plusieurs systèmes de détection d'intrusions et de classification de trafic par application sont développés à ce niveau [7, 8], en utilisant des techniques d'inspection profonde de paquets et outils de capture de paquets [6]. L'analyse de trafic au niveau de paquet est coûteuse et cause la plus grande surcharge de mesure, simplement à cause du fait de devoir analyser chaque paquet qui passe par un interface de réseau. Pour cette raison, ce niveau d'agrégation n'est pas efficace ou même pas réalisable pour la surveillance des réseaux de grande échelle.

Des paquets IP de caractéristiques similaires peuvent être groupés dans un même flux de trafic IP. La définition la plus utilisée de flux IP consiste en un groupe de paquets qui partagent une même 5-tuple, formée par les directions IP d'origine et de destin, les ports d'origine et de destin, et le protocole IP utilisé. La figure 4.30 montre quatre flux IP individuels qui circulent entre les moeuds 3 et 4 du réseau. L'analyse de trafic au niveau de flux IP offre un meilleur bilan entre résolution de l'information de surveillance et consommation de ressources de mesure que l'analyse au niveau de paquet. Plusieurs outils de mesure de trafic basés sur des flux IP ont été développés dans les dernières années, étant NetFlow [9] le plus étendu d'entre eux. Initialement développé par l'entreprise Cisco Systems comme un protocole propriétaire, NetFlow est actuellement un standard émergent de l'IETF (Internet Engineering Task Force), connu comme IPFIX (Internet Protocol Flow Information eXport) [10, 11]. IPFIX est disponible dans les équipements de plusieurs des principaux constructeurs de technologie de réseaux (e.g., Juniper, 3Com, Huawei, Alcatel-Lucent).

L'analyse de trafic au niveau de flux IP présente toutefois certaines restrictions qui peuvent affecter sérieusement le processus de surveillance du trafic réseau. Premièrement, la mesure des flux IP dans des réseaux de grande taille requiert de la technologie spécialisée additionnelle (généralement très coûteuse), en incluant des équipements de mesure des flux, des serveurs pour pré-traiter et centraliser l'information, et des équipements d'analyse si on considère une mise en oeuvre de NetFlow à une grande échelle. Enregistrer des informations des flux IP peut-il être computationnellement coûteux pour les équipements de routage, en pouvant arriver même au point de saturation de mémoire et de capacité de processus si on ne dispose pas de l'équipement précité. Au même temps, exporter des informations des flux IP capturés vers un serveur central peut porter une réduction significative de bande passante dans le réseau, surtout quand il s'agit de surveiller de grands volumes de trafic à très haute vitesse [12]. Ces problèmes sont-ils sérieusement aggravés en considérant le trafic très hétérogène, simplement parce que le nombre des flux IP enregistrés peut rapidement exploser. Cisco Systems propose une variante connue comme "NetFlow échantillonné" pour alléger ces problèmes, où au lieu d'enregistrer chaque paquet d'un même flux, le router prend une de chaque n paquets.

Encore ainsi, NetFlow échantillonné présente certaines déficiences qui rendent difficile la tâche de mesure et l'analyse de trafic [12, 13]. La sélection du taux d'échantillonnage adéquat est un problème intrinsèquement compliqué, puisqu'aucun taux fixe ne fournit un bilan idéal entre consommation de ressources dans le router associé et précision de mesure pour tous les types de trafic. Les volumes de trafic mesurés avec des flux IP échantillonnés résultent une estimation du volume réel, ce qui peut avoir un fort impact sur la qualité du processus de surveillance si on ne prend pas en considération les problèmes avant mentionnés. D'autre part, la reconstruction de flux IP en utilisant échantillonnage des flux est réellement compliquée, puisque les heuristiques de reconstruction utilisées ne sont pas suffisamment robustes [13].

Un groupe de flux IP peut être agrégé dans des flux Origine-Destination (OD). Un flux OD consiste en tous les flux IP qui partagent le même noeud d'origine et le même noeud destination dans le réseau. Dans la figure 4.30, les quatre flux IP décrits avant peuvent être rassemblés dans deux flux OD, le premier avec origine dans le noeud 1 et destination dans le noeud 4, et le deuxième avec origine dans le noeud 2 et destination dans le noeud 4. Le noeud d'entrée et le noeud de sortie de chaque flux IP doivent être identifiés pour pouvoir construire des flux OD. Cette identification est effectuée en général par inspection des tableaux de routage [34, 124]. L'agrégation du trafic au niveau de flux OD pose un problème de surveillance de trafic de dimensions très réduites par rapport au niveau de flux IP. Cependant, la surveillance de trafic basé sur des flux OD présente des problèmes similaires à ceux avant mentionnés, dû principalement au fait que la même technologie de mesure est utilisée pour construire des flux OD (i.e., NetFlow).

Une vision complète des flux OD qui circulent dans un réseau de grande taille est typiquement représentée par une Matrice de Trafic (TM). La TM représente le volume total de trafic transmis entre chaque paire de noeuds entrée et sortie de trafic dans le réseau. Dans la pratique, le terme "volume de trafic" il fait référence aux bytes totaux qui circulent une ou plusieures interfaces de réseau entre deux moments consécutifs de mesure. Pour pouvoir construire une TM, il est nécessaire de disposer d'équipement de mesure de flux IP au moins dans tous les noeuds d'entrée et de sortie du réseau, entraînant les problèmes avant mentionnés.

Puisque la TM est une représentation du volume de trafic, le type d'anomalies qu'ils peuvent être détectées à partir de leur analyse est le composé par des "anomalies de volume". Une anomalie de volume consiste en variations forts et soudaines du volume de trafic sur un ou plusieurs flux OD. Celles-ci peuvent avoir un impact significatif sur la Qualité de Service (QoS) global du réseau, en affectant sérieusement la peeformance des services offerts.

Finalement, le niveau d'agrégation de trafic le plus "basse résolution" est celui-là représenté par le trafic au niveau de liaison de réseau. Dans la figure 4.30, les deux flux OD partagent la même liaison de réseau entre les noeuds 3 et 4. Le trafic au niveau de liaison fait référence au volume total de trafic qui circule entre deux noeuds, physiquement reliés par une liaison de réseau. Le volume du trafic qui circule certain liaison peut être facilement mesuré à partir du protocole SNMP (Simple Network Management Protocol). Ce protocole permet de récolter des informations de tout équipement gérable dans un réseau IP, informations disponibles dans un ensemble de variables connues comme variables MIB (Management Information Base). Tout dispositif d'un réseau IP contient un ensemble de variables MIB qui sont spécifiques à leurs fonctionalités particuliers, comme l'utilisation de la mémoire du dispositif, la charge du processeur, et la bande passante utilisée par une interface de réseau entre autres. Pour pouvoir mesurer la quantité totale de bytes qui circulent à travers une liaison de réseau, il est possible de consulter deux variables MIB spécifiques d'un interface de réseau: la variable `ifInOctets` et la variable `ifOutOctets`. Les deux

variables sont simplement des variables recursives qui accumulent la quantité totale de bytes qui passent à travers l'interface gérée. Le volume de trafic de liaison pourvue par SNMP consiste en la différence entre deux lectures consécutives de ces variables.

SNMP est unique dans le sens qu'il est supporté par pratiquement tout dispositif d'un réseau IP, et il est directement disponible dans les équipements de routage pour pouvoir exécuter des tâches de surveillance de trafic, sans avoir besoin de technologie de mesure additionnelle. En même temps, l'analyse de trafic au niveau de liaison est par loin la technique moins coûteuse, tant dans des questions d'équipement comme de surcharge des équipements de routage. Pour cette raison, la surveillance du trafic de réseau à partir de mesures SNMP de liaison est très attirant pour la surveillance de réseaux de grande échelle. Toutefois, SNMP présente aussi des limitations pratiques. Les lectures SNMP sont envoyées vers un collecteur central par le protocole UDP, ce qui peut résulter en perte d'information de mesure. SNMP souffre aussi des problèmes de synchronisation de mesures dans des réseaux de grande taille, et n'assure pas que les mesures de toutes les interfaces de réseau seront reçues simultáneamnete dans le point d'analyse du trafic.

Dans les travaux développés dans cette thèse, on effectue l'analyse et la surveillance du trafic de réseau au niveau de flux OD. Les trois raisons qui ont motivé cette décision sont les suivantes: tout d'abord, l'agrégation de trafic au niveau de flux OD est suffisamment fine comme pour détecter beaucoup des anomalies qui attentent contre le fonctionnement correct des réseaux de grande taille [71], lesquelles représentent en grande mesure le support de l'Internet. Deuxièmement, la surveillance des flux OD permet d'analyser le trafic dans une échelle de réseau global, considérant l'étude de la Matrice de Trafic. Finalement, il est possible de concevoir des mesures de réponse à ces anomalies avec un impact global dans la performance des services offerts par ces réseaux de grande taille.

Pour éviter les problèmes associés à la mesure directe des flux OD préalablement mentionnés, nous analyserons le comportement de la TM depuis un niveau d'agrégation de trafic de basse résolution, en utilisant des mesures SNMP de trafic de liaisons comme l'information d'entrée pour nos algorithmes. L'utilisation de mesures SNMP permet de concevoir des systèmes de surveillance de grande échelle avec un coût bas et une installation facile, en profitant au maximum de la disponibilité de la technologie SNMP dans tout réseau IP. Toutefois, chaque niveau d'agrégation de trafic a un coût associé à traiter. Dans le cas des mesures SNMP pour l'analyse de flux OD, un clair problème de "observabilité" se présente: le nombre de liaisons dans tout réseau est en général beaucoup plus petit que le nombre des flux OD qui circulent ce réseau, ce pourquoi la TM n'est pas directement observable à partir de mesures de liaison. Ce qui est intéressant dans ce problème d'observabilité c'est que le "coût associé" il peut partiellement "remboursé" par l'utilisation d'outils de modélisation statistique appliqués au trafic d'un réseau IP de grande taille, en utilisant des algorithmes intelligents et efficaces au lieu de mettre en oeuvre une technologie plus coûteuse et complexe.

Les Contre-Mesures: Quelle Décision Prendre?

Le premier pas dans la résolution d'un problème est de connaître son existence. Mais que faire ensuite? Les opérateurs de réseau n'ont seulement besoin de détecter des anomalies dans le trafic de réseau, mais aussi de localiser leurs origines pour prendre des mesures de réponse appropriées. Les mesures de réponse doivent réduire rapidement les impacts négatifs que les anomalies de trafic ont sur le fonctionnement global du réseau, ainsi que maintenir l'intégrité des services et des données délicats en cas d'attaques au réseau. Un système de surveillance de réseau complet doit alors aider à l'opérateur du réseau dans la détection des comportements anormaux, en localisant en même temps ses origines et en proposant des mesures de réponse pertinentes.

L'application de mesures de réponse dans des réseaux de grande taille est un processus d'automatisation difficile, principalement parce que les diverses classes d'anomalies requièrent-elles des réponses différentes. Dans notre contexte de surveillance de la TM, nous sommes particulièrement intéressés aux anomalies de volume dans le trafic des flux OD. Les impacts les plus importants de ce type d'anomalies sont les situations de haute congestion dans les liaisons, qui affectent directement le fonctionnement global du réseau.

Une mesure de réponse possible face aux anomalies de volume est la ré-configuration du routage du réseau. La performance de tout réseau dépend en grande mesure sur l'opération des protocoles de routage sous-jacents. Les réseaux IP de grande taille combinent en général des différents mécanismes de protection et restauration pour réduire la dégradation de fonctionnement en présence d'anomalies [57, 58], en concevant des topologies de réseau redondantes et sur-approvisionnées. Toutefois, les coûts chaque fois plus grands associés à des conceptions robustes du réseau ont joué un rôle important dans la détermination des mécanismes de récupération utilisés actuellement [56]. Comme alternative, plusieurs opérateurs de réseau ont opté pour la restauration de réseau basé en ré-configuration de routage et ré-établissement de chemins [56].

Dans cette thèse nous avons exploré un nouveau paradigme d'optimisation de routage connu comme Routage Robuste (RR) pour établir des configurations de routage robustes et efficaces qui réduisent les impacts des anomalies de volume sur le fonctionnement global du réseau. Différentes variantes de RR ont été proposées et analysées, y compris non seulement des techniques de reconfiguración de ruteo mais aussi d'équilibrage de charge. Ces propositions non seulement réduisent les problèmes de congestion induits par les anomalies de volume, mais aussi fournissent une meilleure utilisation des ressources du réseau dans une perspective de Qualité de Service (QoS), une propriété fondamentale pour maintenir des services de réseau en fonctionnement correcte même en présence d'anomalies de trafic.

La figure 4.31 décrit le contexte adopté pour le problème de surveillance et analyse de trafic du réseau abordé dans la thèse. En bref, nous proposons d'analyser le trafic de réseau dans des réseaux de grande taille, en détectant des anomalies de volume dans

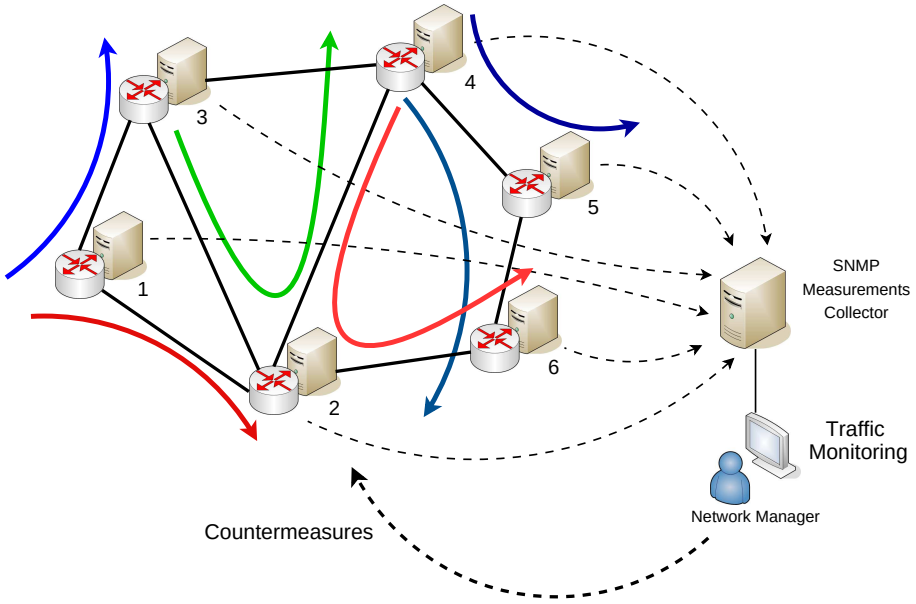


Figure 4.26 — Surveillance de la Matrice de Trafic Intra-domaine.

la Matrice de Trafic à partir de mesures SNMP de trafic de liaison. En outre, nous proposons d'identifier les origines des anomalies détectées, en déployant des contre-mesures appropriées, basées sur des techniques de ré-configuration de routage robuste et d'équilibrage de charge.

Contributions des Travaux de Thèse

Pour les multiples raisons présentées avant, nous croyons que les systèmes de surveillance en grande-échelle doivent construire des vues globaux du trafic de réseau à partir des mesures partielles et de bas coût pour la mise en oeuvre, en les combinant avec des algorithmes d'analyse statistiques intelligentes et efficaces. En même temps, ces systèmes doivent être capables de détecter et de localiser rapidement les différentes anomalies de trafic dans le réseau, en répondant avec des mesures appropriées que permettent de maintenir les fonctions du réseau avec un niveau de performance raisonnable. Une mise en oeuvre confiable d'un tel groupe de techniques serait hautement bénéfique pour les opérateurs de réseau, en fournissant un mécanisme léger et facile à déployer pour monitorer le trafic de réseau dans des réseaux de grande taille.

Les travaux présentés dans cette thèse offrent des contributions importantes dans trois domaines différents liés aux réseaux de données, particulièrement en relation avec la surveillance et l'analyse du trafic de réseau dans des réseaux de grande taille: (i) modélisation et estimation de la Matrice de Trafic, (ii) détection et localisation d'anomalies de volume dans la Matrice de Trafic à partir des mesures SNMP, et (iii) optimisation de routage en présence de trafic de réseau très variable et incertain. Malgré la littérature étendue disponible dans ces trois domaines, laquelle est analysée en profondeur dans les chapitres 2, 3, et 4 de la thèse, nos études montrent que à cette date il n'existe pas une technique complète que permet de détecter et de localiser des anomalies de volume dans la Matrice de Trafic d'un réseau de grande taille de manière optimale et à partir de mesures SNMP, en déployant des contre-mesures de ré-configuration de routage et d'équilibrage de charge basées en Qualité de Service.

La première contribution importante de la thèse est liée à la modélisation statistique de la Matrice de Trafic. Nous avons développé un nouveau modèle paramétrique, linéaire, et de basses dimensions pour décrire le comportement sans anomalies de une TM dans un réseau IP de grande taille. Ce modèle de trafic a plusieurs applications et présente des différents avantages par rapport aux modèles proposés dans la littérature pour la TM: (i) étant un modèle de basses dimensions, il permet de résoudre les problèmes d'analyse de la TM à partir des mesures SNMP, en permettant en particulier de résoudre le problème d'Estimation de la Matrice de Trafic (TME), lequel est introduit dans le chapitre 2. (ii) À la différence de plusieurs modèles basés sur des mesures de trafic, le nôtre est paramétrique et stable tout au long du temps, ce qui permet de concevoir des méthodes de détection d'anomalies confiables. (iii) Nôtre modèle utilise exclusivement des mesures SNMP pour construire une image précise de la TM, en simplifiant des questions pratiques. Finalement et plus important, (iv) ce modèle linéaire et de basses dimensions permet de filtrer le trafic libre d'anomalies du problème de détection, ce qui permet de calculer des résidus sensibles aux différentes anomalies de volume. Cette caractéristique a-priori simple nous a permis concevoir des algorithmes optimaux de détection et localisation d'anomalies de volume dans le trafic de la TM. Ce point est étudié à fond dans le chapitre 3.

Nos études dans le domaine de la modélisation statistique de la TM ont aussi produit des résultats intéressants dans le problème d'estimation de la TM, où nous avons introduit plusieurs améliorations à deux techniques d'estimation préalablement introduites, en améliorant des divers problèmes de base de ces techniques et obtenir des meilleurs résultats. En particulier, nous avons proposé deux techniques améliorées de TME, la première basée des filtres de Kalman et la deuxième basée sur des techniques d'apprentissage statistique.

La deuxième contribution de la thèse concerne la détection et la localisation d'anomalies de volume dans la TM, en utilisant des mesures SNMP comme données d'entrée. En utilisant le modèle paramétrique de basses dimensions décrit préalablement, nous avons proposé deux algorithmes différents pour détection et localisation d'anomalies de volume, avec un avantage prépondérant par rapport aux propositions précédentes, qui consiste en des conditions d'optimalitée bien démontrées. Cette caractéristique généralement absente dans les travaux préalables est fondamentale pour le développement d'algorithmes généraux, déliés d'un réseau ou d'un ensemble des réseaux en particulier et plus important encore, indépendants d'évaluations particulières sur des conditions spéciales de trafic. Les méthodes de détection "fait à la main" ils peuvent fonctionner correctement dans certaines scènes, mais sans une base théorique solide et généralisable, ses résultats sont de validité limitée.

Le premier de ces algorithmes a été conçu pour la détection optimale d'anomalies de volume, en maximisant le taux de détection correcte pour un taux de fausses alarmes bornée. Le deuxième algorithme permet de détecter et de localiser simultanéement un flux OD anormal dans la TM, en diminuant le retard moyenne maximal de détection et de localisation pour un taux de fausse localisation et un taux de fausses alarmes bornées.

La troisième contribution principale de cette thèse est en rapport avec l'optimisation et la ré-configuration du routage et de l'équilibrage de charge intra-domaine dans un réseau de grande taille, dans des conditions de trafic de réseau hautement variable. Motivés par la bonne performance d'un nouveau paradigme d'optimisation de routage sous incertitude appelé Routage Robuste (RR), nous avons étudié en profondeur leur possible application comme mesure de réponse face aux anomalies de volume détectées. Nos études ont révélé des divers défauts présents dans les techniques actuelles de RR pour manier des variations de trafic grandes et abruptes d'une manière efficace, et diverses solutions ont été proposées. D'abord, nous avons développé deux variantes de la méthode de base pour reconfigurer le routage intra-domaine du réseau, la première basée sur une extension multi-temporelle du RR, et la deuxième basée sur une technique pro-active pour calculer a priori des configurations de routage optimales en présence d'anomalies de volume. Deuxièmement, nous avons analysé des nouveaux critères d'optimisation pour calculer configurations de RR avec des prestations de QoS. Finalement, nous avons exploré le paradigme

de Équilibrage Dynamique de Charge (DLB) dans des réseaux intra-domaine, en fournissant une analyse comparative profonde entre les différentes techniques de RR développées et des divers mécanismes de DLB en présence de trafic hautement variable.

Pour vérifier l'applicabilité de nos contributions dans des réseaux opérationnels réels, tous les algorithmes proposés dans la thèse ont été validés en utilisant des données réelles de trafic de divers réseaux IP de grande taille. En même temps, sa performance a été comparé contre des travaux de réputation dans chacun des domaines traités, en obtenant des résultats semblables ou meilleurs dans la majorité des cas. Pour résumer, la liste suivante présente les contributions les plus importants de cette thèse:

- Un nouveau modèle paramétrique, linéaire, et de basses dimensions pour analyser le comportement normal (libre d'anomalies de volume) de la Matrice de Trafic d'un réseau IP de grande taille.
- Des nouvelles méthodes pour l'estimation efficace de la Matrice de Trafic d'un réseau IP de grande taille.
- Une méthode pour détecter des anomalies de volume dans la Matrice de Trafic à partir de mesures de basse résolution. Cette méthode présente des conditions d'optimalitée bien établies en termes de taux de détection correcte et taux de fausses alarmes.
- Une méthode pour détecter et localiser rapidement des anomalies de volume dans la Matrice de Trafic à partir de mesures de basse résolution. Cette méthode présente des conditions d'optimalitée bien établies en termes de délai de détection et de localisation, ainsi qu'en ce qui concerne la localisation erronée et le taux de fausses alarmes.
- Une extension Multi-Temporelle du Routage Robuste, laquelle permet d'adapter la configuration de routage aux variations normales du trafic réseau de manière plus efficace que celle pourvue par l'analyse originale.
- Une nouvelle technique d'optimisation de Routage Robuste, améliorée pour fournir des configurations de RR avec des prestations de QoS.
- Une méthode réactive d'Équilibrage Robuste de Charge, visant à compenser les effets négatifs des anomalies de volume sur la performance globale d'un réseau IP de grande taille.
- Une étude comparative des vertus et défauts de différentes techniques de Routage Robuste et d'Équilibrage Dynamique de Charge.

Pour terminer, je voudrais indiquer que les contributions de cette thèse sont le résultat de divers travaux de collaboration effectués entre les années 2006 et 2009 avec plusieurs professeurs et chercheurs de différentes institutions. En particulier, les contributions en rapport avec la modélisation et l'estimation de la TM, et celles associées à la détection et la localisation d'anomalies du volume dans la TM, sont le résultat des

travaux conjoints avec le professeur adjoint Lionel Fillatre et le professeur Igor Nikiforov (Université de Technologie de Troyes), et avec le professeur Thierry Chonavel (Télécom Bretagne). Les contributions relatives à l'optimisation de routage et l'équilibrage de charge dans des réseaux intra-domaine sont le résultat de travaux conjoints avec le professeur Walid Ben-Ameur (Télécom & Management SudParis), le professeur Hervé Kerivin (Clemson University), le chercheur associé de recherche postdoctorale Federico Larroca et le professeur Jean-Louis Rougier (Télécom ParisTech).

Structure et Distribution de la Thèse

Les travaux développés dans les trois domaines d'étude mentionnés se présentent au long de trois chapitres. La figure 4.32 décrit l'organisation de la thèse et l'interaction entre les différents chapitres.

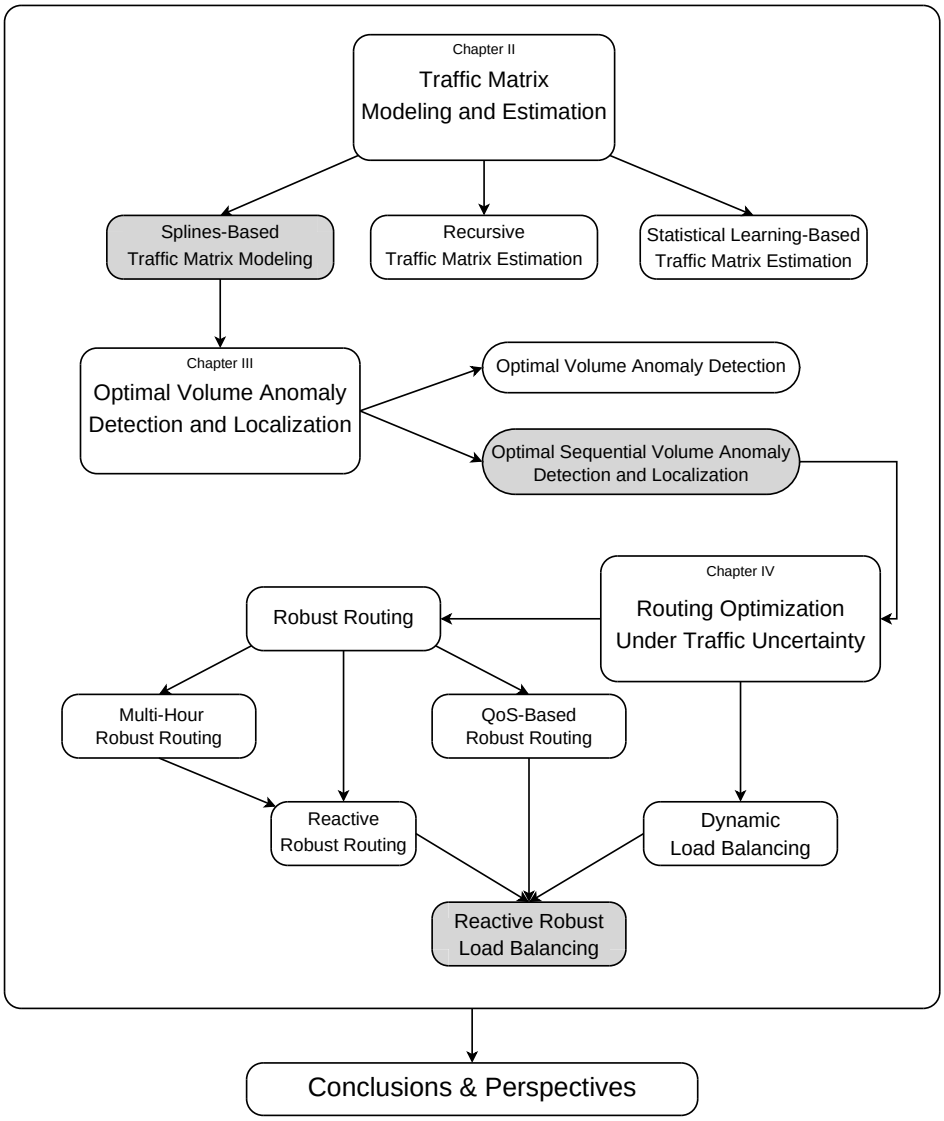


Figure 4.27 — Schéma structurel de la thèse.

Le chapitre 2 présente nos études sur la modélisation et l'estimation de la Matrice de Trafic. Trois modèles pour analyser la TM d'un réseau de grande taille à partir des mesures SNMP sont présentés. Le premier de ces modèles consiste en des techniques de modélisation polinómial en basses dimensions, le deuxième consiste en modélisation de systèmes linéaires et méthodes récursifs d'estimation, et le troisième se base sur des techniques d'apprentissage statistique. Dans ce chapitre on analyse d'autres techniques de modélisation et d'estimation de la TM présents dans la littérature, lesquelles sont utilisées comme référence pour l'évaluation de la performance de nos propositions.

Le chapitre 3 présente la conception et l'évaluation de deux algorithmes optimaux pour détection et localisation d'anomalies de volume dans la TM, en utilisant les principes de la théorie de la décision. Les deux algorithmes utilisent le modèle de trafic de basses dimensions présenté dans le chapitre 2 pour filtrer le trafic d'opération normale hors du problème de détection. Les algorithmes les plus représentatifs dans la littérature pour faire la détection et la localisation d'anomalies dans la TM sont aussi présentés et analysés dans ce chapitre. Finalement, on présente une évaluation comparative entre les algorithmes de référence et nos algorithmes, en considérant pas seulement la performance de détection et localisation, mais aussi la complexité numérique de chaque algorithme et d'autres questions relatives à sa mise en oeuvre.

Le chapitre 4 présente une étude sur les paradigmes de Routage Robuste et Équilibrage Dynamique de Charge dans des réseaux de grande taille. Des diverses variantes et améliorations aux méthodes traditionnels sont proposées et évaluées dans ce chapitre. Finalement, on présente la conception d'un méthode complet d'Équilibrage Robuste de Charge avec des prestations de QoS, en utilisant un des algorithmes de détection et localisation d'anomalies présenté dans le chapitre 3.

Pour finir, les conclusions sur les travaux de thèse développés se présentent, ainsi que différentes perspectives de travail futur et de possibles lignes de recherche à explorer.

Análisis Estadístico del Trafico de Red para la Detección de Anomalías y la Calidad de Servicio

Internet, el propulsor principal de la actual "era de la información", se ha transformado en uno de los actores principales de nuestra sociedad. Internet es hoy en día el componente fundamental en la infraestructura de comunicación global, desempeñando un rol crucial en la educación, la economía, el entretenimiento, y la vida social de nuestras naciones. Su extraordinario e imparable crecimiento en todo el mundo en la última década ha llevado al florecimiento de miles de compañías que generan y "sumergen" sus contenidos en Internet. Esta proliferación de contenidos ha sido acompasada por un crecimiento sostenido de consumidores, llevando a una explosión del tráfico presente en las redes de datos que conforman Internet, no sólo a nivel de volumen de tráfico, si no también en cuanto a heterogeneidad y complejidad de composición.

Luego de una breve recaída a mediados de esta década, los actores más importantes de Internet pronostican que el tráfico de red doblará su volumen casi cada dos años en el futuro cercano, impulsado por la penetración del vídeo de alta definición y del acceso de muy alta velocidad. Se espera que el tráfico IP total crecerá de 6.6 exabytes por mes en 2007 a casi 29 exabytes por mes en 2011 (1 exabyte = 10^{18} bytes), cuadruplicando su volumen en menos de 5 años [3, 4].

Al mismo tiempo, el desarrollo de las redes ópticas y la evolución de las tecnologías de acceso, notablemente la tecnología FTTH (Fiber To The Home) aumentarán dramáticamente el ancho de banda para los usuarios finales, imponiendo problemas serios e imprevistos en las redes de backbone, supuestas hasta ahora de capacidad infinita. La industria de la tecnología FTTH pronostica una demanda de ancho de banda por usuario de hasta 30 Gbps en 2030 [1, 2]. La figura 4.28 presenta la evolución prospectiva del tráfico de Internet y del ancho de banda de muy alta velocidad para los próximos 2 o 3 años.

Este futuro cercano impone nuevos desafíos para los operadores de redes de gran escala, quienes al fin y al cabo son en gran medida los responsables de sustentar el crecimiento de Internet. Los usuarios finales quieren una Internet más rápida, más segura y con mejores prestaciones de calidad de servicio, y el análisis y monitoreo del tráfico que circula la red es probablemente la solución más eficiente y al alcance de la mano para los operadores de red. Conocer y comprender el tráfico que circula la red es crucial para el diseño eficiente, el funcionamiento correcto y la ingeniería de los servicios ofrecidos sobre Inernet.

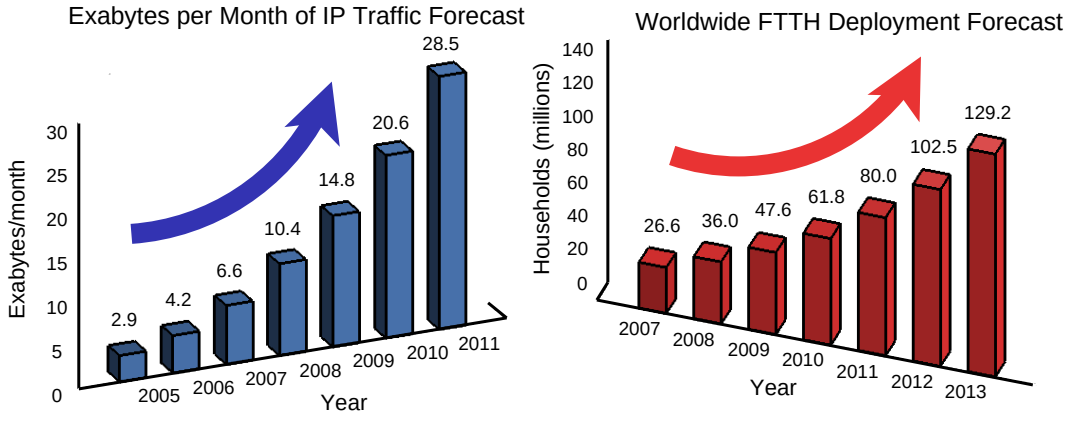


Figure 4.28 — Perspectivas del incremento de tráfico en Internet y del despliegue de la tecnología FTTH en los próximos años.

El monitoreo del tráfico en la red es sin lugar a dudas una de las tareas críticas para los operadores de red que será seriamente afectada por este fuerte desarrollo. En efecto, capturar y analizar grandes volúmenes de tráfico heterogéneo y en multiples puntos de la red puede resultar extremadamente costoso. En los comienzos de Internet, el monitoreo de la red era más un arte que una ciencia, dependiendo en gran medida en la experiencia y el conocimiento del operador de red para analizar el tráfico "manualmente". Sin embargo, la creciente complejidad y el tamaño de las redes de proveedores de servicios en Internet ha motivado el desarrollo de sistemas automáticos de monitoreo a gran escala en los últimos años.

El monitoreo del tráfico de red puede realzar diversas actividades relacionadas con la gestión de la red, tales como el planeamiento y el diseño de la arquitectura de red, la caracterización y clasificación del tráfico, la identificación de fallas o de problemas de degradación de funcionamiento, e incluso la detección de tráfico malicioso (i.e., ataques de red). Justamente es posible apreciar hoy en día una gran cantidad y variedad de Sistemas de Detección de Intrusiones (IDS) que han dotado a la red de nuevas capacidades para gestionar el tráfico malicioso.

El proceso de monitoreo de tráfico consiste en tres tareas consecutivas: la captura de datos, el análisis de estos datos, y la decisión extraída de dicho análisis. Cada una de estas tareas se hace más y más difícil en el escenario actual del tráfico de red. La recolección de datos es muy costosa, porque hay demasiado tráfico a capturar y en diversas partes de la red. El análisis de datos es más complejo, porque el tráfico es más heterogéneo y son más variadas las posibles falencias que este puede presentar. Una decisión correcta es mucho más crítica que antes, porque los servicios proporcionados en la Internet actual son más vitales que en el pasado.

Otro tema relacionado con el monitoreo del tráfico de red de alta velocidad y a gran escala en un futuro cercano es el de la rentabilidad de los sistemas de monitoreo. Los nuevos modelos de negocio en Internet de los últimos años han dado lugar a muchas

soluciones de virtualización de red [14], permitiendo que pequeños proveedores de servicio capturen parte del mercado de Internet con inversiones muy reducidas en lo que infraestructura de red refiere. Esto ha llevado a muchos operadores de red a reducir drásticamente sus inversiones en infraestructura, buscando soluciones que permitan sacar el mayor provecho de la tecnología disponible en sus redes de hoy en día. Los futuros sistemas de monitoreo deben entonces apuntar al análisis del tráfico global de la red mediante mediciones limitadas de tráfico y usando información "más barata" y de baja resolución, utilizando procedimientos de inferencia y algoritmos inteligentes de análisis para reducir costos de medición de tráfico, sin reducir el desempeño del proceso de monitoreo.

¿Dónde monitorear el Tráfico de Red?

A pesar del crecimiento masivo de Internet de los últimos años, su estructura global sigue siendo fuertemente jerárquica. La espina dorsal de Internet está compuesta por un número reducido de grandes Sistemas Autónomos (ASes) [5], conocidos como redes Tier-1. Un AS es básicamente una colección de prefijos de ruteo IP que comparten una misma política de ruteo hacia Internet y que están bajo el dominio de un mismo operador de red [20]. Una lista no exhaustiva de las redes Tier-1 actuales incluye AT&T, Global Crossing, Level 3 Communications, NTT Communications, Sprint, Tata Communications, Verizon Business (UUNET), Savvis, TeliaNet, Bell Canada, y XO Communications (XOXO). La figura 4.29 presenta un mapa de la estructura actual de Internet, provisto por CAIDA [127].

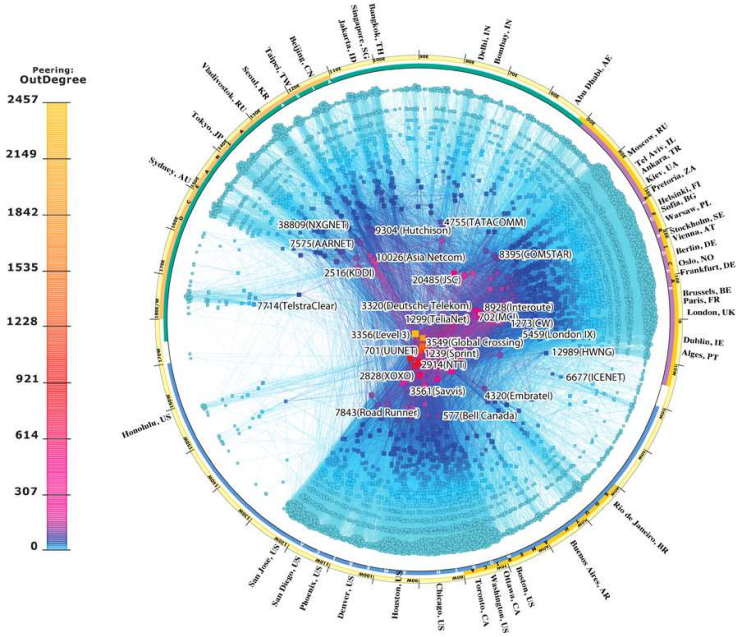


Figure 4.29 — Mapa de la topología IPv4 de Internet en enero del 2009.

Las redes Tier-1 proveen conectividad global dentro de Internet y representan el primer nivel en la jerarquía. Los siguientes niveles en la jerarquía de Internet están compuestos por ASes más pequeños y menos interconectados, conocidos como redes Tier-2 y Tier-3. Algunos ejemplos de redes Tier-2 son la red de telecomunicaciones de la empresa alemana Deutsche Telecom, la red de British Telecom, y la red de France Telecom entre otras. Finalmente, los márgenes de Internet están compuestos por ASes terminales, usualmente llamados stub ASes.

Esta división en Sistemas Autónomos proporciona dos visiones estructurales distintas de Internet: la Internet intra-dominio y la Internet inter-dominio [125]. La Internet intra-dominio está compuesta por los routers interconectados dentro de cada AS individual y que intercambian tráfico entre sí de acuerdo a un protocolo de ruteo intra-dominio. Cada AS tiene una topología de red propia y que es perfectamente conocida por el operador de red que lo administra. Por otra parte, la Internet inter-dominio esta compuesta por los distintos ASes y sus respectivas interconexiones. Las características internas de cada AS son transparentes desde un punto de vista inter-dominio, y el tráfico es intercambiado entre estos mediante un protocolo de ruteo inter-dominio.

El monitoreo de tráfico y de red es normalmente realizado a escala intra-dominio en redes de alto porte (i.e., redes Tier-1 y Tier-2), básicamente porque a dicha escala la topología de red es perfectamente conocida por el operador de red. Al mismo tiempo, es solamente a dicha escala que el proveedor tiene control completo sobre la red, y por lo tanto puede manipularla y configurarla sin restricciones. El análisis de tráfico a nivel inter-dominio es una tarea largamente más desafiante, porque la información disponible es menor, los diferentes operadores de red no necesariamente colaboran entre sí, cuestiones de privacidad y de negocios limitan el intercambio de información entre operadores, y muchas otras características de la escala inter-dominio que dificultan la tarea de monitoreo.

En este trabajo de tesis nos hemos centrado en el monitoreo y análisis del tráfico de red al nivel intra-dominio, básicamente por dos razones. En primer lugar, el control que podemos tener sobre la red a nivel intra-dominio permite proponer soluciones más completas, no solo en lo que refiere al análisis del tráfico, sino también en cuanto al proceso de recuperación frente a eventos imprevistos. En segundo lugar, la estructura de Internet sigue estando fuertemente concentrada en un pequeño grupo de redes de alto porte, lo que conduce al hecho de que el desempeño de Internet en su conjunto depende en gran medida del desempeño individual de estas redes de alto porte.

¿Qué Información Monitorear?

Los operadores de red están confrontados rutinariamente con un amplio espectro de eventos inusuales que atentan contra el funcionamiento correcto de sus redes. Un problema mayor asociado a la detección de estos eventos anómalos es que sus causas

y orígenes pueden variar considerablemente. Distintas anomalías en la red y/o en el tráfico que circula la red pueden desencadenarse por causas muy variadas, desde fallas de equipos y errores de configuración, comportamientos inusuales de uno o varios usuarios finales (e.g., eventos de flash crowd, transferencias de alto volumen de tráfico, etc.), modificaciones del ruteo externo al AS en cuestión, ataques de red (e.g., ataques de tipo DDoS, scans de puertos y de red, propagación de gusanos, etc.) hasta incluso nuevos eventos anómalos nunca antes vistos.

Un desafío importante relacionado con la detección de anomalías en la red es que éstas son un objetivo "móvil". Es inherentemente difícil el definir y precisar permanentemente el conjunto posible de anomalías que pueden impactar la red, sobre todo en el caso de tráfico malicioso. Nuevas anomalías aparecen constantemente, por lo cual los sistemas de detección de anomalías deben evitar el estar ligados a un conjunto predefinido de anomalías a detectar.

Distintos tipos de anomalías de red y de tráfico pueden ser detectadas, dependiendo del tipo de información monitoreada y de su nivel de agregación. En general podemos identificar cuatro niveles básicos de agregación de tráfico, considerando la granularidad de la información que dicho nivel de agregación aporta: paquete IP, flujo IP, flujo OD, y tráfico de enlace. La figura 4.30 nos ayudará a explicar esta clasificación. El análisis de trafico a nivel de paquetes IP proporciona la información de monitoreo más rica y de fina granularidad. Trabajando a esta granularidad es posible analizar las características particulares de cada paquete IP, accediendo inclusive a su carga útil. Muchos sistemas de detección de intrusiones y de clasificación de tráfico por aplicación son desarrollados a este nivel [7, 8], utilizando técnicas de inspección profunda de paquetes y herramientas de captura de paquetes [6]. El análisis de tráfico a nivel de paquete es costoso y causa la mayor sobrecarga de medida, simplemente por el hecho de tener que analizar cada paquete que pasa por una interfaz de red. Por esta razón, este nivel de agregación no es eficiente o incluso implementable para monitoreo de redes de gran escala.

Paquetes IP de características similares pueden ser agrupados en un mismo flujo de tráfico IP. La definición más utilizada de flujo IP consiste en un grupo de paquetes que comparten una misma 5-tupla, formada por las direcciones IP de origen y de destino, los puertos de origen y de destino, y el protocolo IP utilizado. La figura 4.30 muestra cuatro flujos IP individuales que circulan entre los nodos 3 y 4 de la red. El análisis de tráfico a nivel de flujos IP ofrece un mejor balance entre granularidad de la información de monitoreo y consumo de recursos de medición que el análisis a nivel de paquete. Muchas herramientas para medición de tráfico basado en flujos IP han sido desarrolladas en los últimos años, siendo NetFlow [9] la más extendida entre ellas. Inicialmente desarrollado por la empresa Cisco Systems como un protocolo propietario, NetFlow es hoy en día un estándar emergente de la IETF (Internet Engineering Task Force), conocido como IPFIX (Internet Protocol Flow Information eXport) [10, 11]. IPFIX es implementado en los equipos de varios de los principales constructores de tecnología de redes (e.g., Juniper, 3Com, Huawei, Alcatel-Lucent).

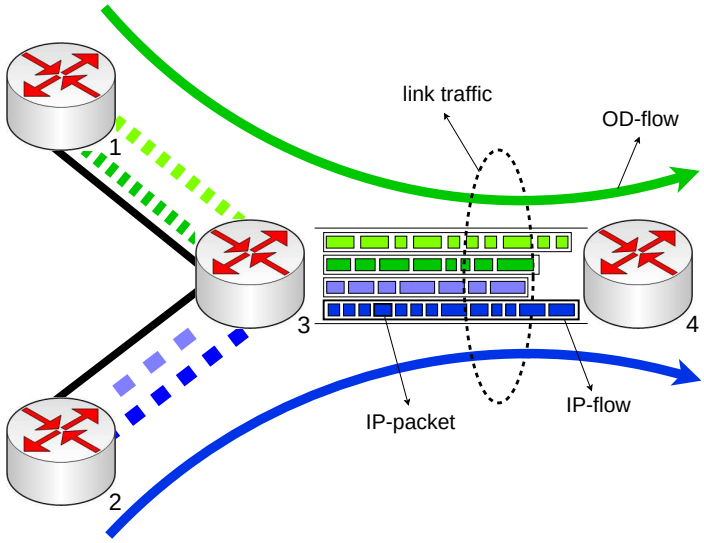


Figure 4.30 — Diferentes niveles de agregación de tráfico.

El análisis de tráfico a nivel de flujos IP presenta sin embargo ciertas restricciones que pueden afectar seriamente el proceso de monitoreo. En primer lugar, la medición de flujos IP en redes de alto porte requiere tecnología especializada adicional (generalmente muy costosa), incluyendo equipos de medición de flujos, equipos servidores para pre-procesar y centralizar la información, y equipos de análisis al considerar una implementación de NetFlow a gran escala. Registrar información de flujos IP puede ser computacionalmente costoso para los equipos de ruteo, pudiendo llegar incluso al punto de saturación de memoria y de capacidad de proceso si no se dispone del equipamiento antes mencionado. Al mismo tiempo, exportar información de los flujos IP capturados hacia un servidor central puede causar una reducción significativa del ancho de banda en la red, sobre todo cuando se trata de monitorear grandes volúmenes de tráfico de muy alta velocidad [12]. Estos problemas se ven seriamente agravados al considerar tráfico muy heterogéneo, simplemente porque el número de flujos IP registrados puede dispararse rápidamente. Cisco Systems propone una variante conocida como "NetFlow muestreado" para aliviar estos problemas, donde en lugar de registrar cada paquete de un mismo flujo, el router mide uno de cada n paquetes.

Aún así, NetFlow muestreado presenta ciertas deficiencias que dificultan la tarea de medición y el análisis de tráfico [12, 13]. La selección de la tasa de muestreo adecuada es un problema intrínsecamente complicado, ya que ninguna tasa fija provee un balance ideal entre consumo de recursos del router asociado y precisión de medida para todos los tipos de tráfico. Los volúmenes de tráfico medidos con flujos IP muestreados resultan una estimación del volumen real, lo que puede impactar fuertemente la calidad del proceso de monitoreo si no se tienen en cuenta los problemas antes mencionados. Por otro lado, la reconstrucción de flujos IP al usar muestreo de flujos es realmente complicada, ya que las heurísticas de reconstrucción utilizadas no son suficientemente robustas [13].

Un grupo de flujos IP puede ser agregado en flujos Origen-Destino (OD). Un flujo OD consiste en todos los flujos IP que comparten el mismo nodo de origen y el mismo nodo destino dentro de la red. En la figura 4.30, los cuatro flujos IP previamente descritos pueden ser agregados en dos flujos OD, el primero con origen en el nodo 1 y destino en el nodo 4, y el segundo con origen en el nodo 2 y destino en el nodo 4. El nodo de ingreso y el nodo de salida de cada flujo IP deben ser identificados para poder construir flujos OD. Esta identificación se realiza en general mediante inspección de las tablas de ruteo [34, 124]. La agregación a nivel de flujos OD propone un problema de monitoreo de tráfico de dimensiones muy reducidas con respecto al nivel de flujos IP. No obstante, el monitoreo de tráfico basado en flujos OD presenta problemas similares a los antes mencionados, debido básicamente a que la misma tecnología de medición es utilizada para construir flujos OD (i.e., NetFlow).

Una visión completa de los flujos OD que circulan en una red de alto porte es típicamente provista por una Matriz de Tráfico (TM). La TM representa el volumen total de tráfico transmitido entre cada par de nodos de ingreso y salida de tráfico en la red. En la práctica, el término "volumen de tráfico" hace referencia al acumulado de bytes que circulan una o varias interfaces de red entre dos instantes consecutivos de medida. Para poder construir un TM es necesario disponer de equipamiento de medición de flujos IP al menos en todos los nodos de entrada y de salida de la red, arrastrando nuevamente a los problemas antes mencionados.

Dado que la TM es una representación del volumen de tráfico, el tipo de anomalías que pueden ser detectadas a partir de su análisis es el compuesto por las "anomalías de volumen". Una anomalía de volumen consiste en fuertes y repentinas variaciones del volumen de tráfico en uno o varios flujos OD. Estas fuertes variaciones pueden impactar significativamente la Calidad de Servicio (QoS) global de la red, afectando seriamente el desempeño de los servicios ofrecidos.

Por último, el nivel de agregación de tráfico más "grueso" es aquel representado por el tráfico a nivel de enlace de red. En la figura 4.30, los dos flujos OD comparten el mismo enlace de red entre los nodos 3 y 4. El tráfico a nivel de enlace hace referencia al volumen total de tráfico que circula entre dos nodos, físicamente conectados por un enlace de red. El volumen del tráfico que circula cierto enlace puede ser medido fácilmente mediante el ampliamente difundido protocolo SNMP (Simple Network Management Protocol). Este protocolo permite recolectar información de cualquier equipo gestionable dentro de una red IP, disponible en un conjunto de variables conocidas como variables MIB (Management Information Base). Todo dispositivo de una red IP contiene un conjunto de variables MIB que son específicas a sus funcionalidades particulares, como ser el uso de la memoria del dispositivo, la carga del procesador, y el ancho de banda utilizado por cierta interfaz de red entre otras. Para poder medir la cantidad total de bytes que circulan a través de un enlace de red es posible consultar dos variables MIB específicas de una interfaz de red: la variable `ifInOctets` y la variable `ifOutOctets`. Ambas variables son simplemente

contadores recursivos que acumulan la cantidad total de bytes que pasan a través de la interfaz gestionada. El volumen de tráfico de enlace provisto por SNMP consiste en la diferencia entre dos lecturas consecutivas de dichos contadores.

SNMP es único en el sentido de que es soportado por prácticamente todo dispositivo de una red IP, y se encuentra directamente disponible en los equipos de ruteo para poder ejecutar tareas de monitoreo de tráfico, sin necesidad de tecnología de medición adicional. Al mismo tiempo, el análisis de tráfico a nivel de enlace es por lejos la técnica menos costosa, tanto en cuestiones de equipamiento como de sobrecarga de los equipos de ruteo. Es por esto último que el monitoreo del tráfico de red a partir de medidas SNMP de enlace resulta muy atractivo para monitoreo de redes de gran escala. Sin embargo, SNMP también presenta limitaciones prácticas. Las lecturas SNMP son enviadas a un recolector central mediante el protocolo UDP, lo que puede resultar en pérdida de información de medida. SNMP también sufre problemas de sincronización de medidas en redes de alto porte, y no asegura que las mediciones de todas las interfaces de red serán recibidas simultáneamnete en el punto de análisis del tráfico.

En los trabajos desarrollados en esta tesis, se realiza el análisis y el monitoreo del trafico de red a nivel de flujos OD. Las tres razones que motivaron esta decisión son las siguientes: en primer lugar, la agregación de tráfico a nivel de flujos OD es suficientemente fina como para detectar muchas de las anomalías que atentan contra el funcionamiento correcto de la redes de alto porte [71], las cuales representan en gran medida el soporte de Internet. En segundo lugar, el monitoreo de flujos OD permite analizar tráfico en una escala de red global, considerando el estudio de la matriz de tráfico. Por último, es posible diseñar medidas de respuesta a dichas anomalías con un impacto global en el desempeño de los servicios brindados por estas redes de gran porte.

Para evitar los problemas asociados a la medida directa de flujos OD previamente mencionados, analizaremos el comportamiento de la TM desde un nivel de agregación de tráfico aún más grueso, utilizando medidas SNMP de tráfico de enlaces como la información de entrada para nuestros algoritmos. El uso de medidas SNMP permite concebir sistemas de monitoreo de gran escala de bajo costo y fácil instalación, aprovechando al máximo la disponibilidad de la tecnología SNMP en toda red IP. Sin embargo, cada nivel de agregación de tráfico tiene un costo asociado a tratar. En el caso de medidas SNMP de tráfico de enlace para el análisis de flujos OD se presenta un claro problema de "observabilidad": el número de enlaces en toda red es en general mucho más pequeño que el número de flujos OD que circulan dicha red, por lo que la TM no es directamente observable a partir de medidas de enlace. Lo interesante de este problema de observabilidad es que el "costo asociado" puede ser parcialmente "reembolsado" mediante el uso de herramientas de modelado estadístico aplicadas al tráfico de una red IP de alto porte, utilizando algoritmos inteligentes y eficientes en lugar de implementar una tecnología más costosa y compleja.

Medidas de Respuesta: Qué Decisión Tomar?

El primer paso en la resolución de un problema es conocer su existencia. ¿Pero qué hacer después? Los operadores de red no sólo necesitan detectar anomalías en el tráfico de red, sino también localizar sus orígenes para tomar medidas de respuesta apropiadas. Las medidas de respuesta deben reducir rápidamente los impactos negativos que las anomalías de tráfico tienen sobre el funcionamiento global de la red, así como mantener la integridad de los servicios y de los datos comprometidos en caso de ataques de a la red. Un sistema de monitoreo de red completo debe entonces ayudar al operador de red en la detección de comportamientos anómalos, localizando al mismo tiempo sus orígenes y proponiendo medidas de respuesta pertinentes.

La aplicación de medidas de respuesta en redes de alto porte es un proceso de difícil automatización, básicamente porque las diversas clases de anomalías requieren diferentes respuestas. En nuestro contexto de monitoreo de la TM estamos particularmente interesados en anomalías de volumen en el tráfico de los flujos OD. Los impactos más importantes de esta clase de anomalías son las situaciones de alta congestión en los enlaces, que afectan directamente al funcionamiento global de la red.

Una medida de respuesta posible frente a las anomalías de volumen es la reconfiguración del ruteo de la red. El desempeño de toda red depende en gran medida en la operación de los protocolos de ruteo subyacentes. Las redes IP de alto porte combinan en general distintos mecanismos de protección y restauración para reducir la degradación de funcionamiento en presencia de anomalías [57, 58], diseñando topologías de red redundantes y sobre-aprovisionadas. Sin embargo, los costos cada vez mayores asociados a diseños robustos de la red han desempeñado un papel importante en la determinación de los mecanismos de recuperación utilizados en la actualidad [56]. Como alternativa, muchos operadores de red han optado por la restauración de red basada en reconfiguración de ruteo y el re-establecimiento de caminos [56].

En esta tesis hemos explorado un nuevo paradigma de optimización de ruteo conocido como Ruteo Robusto (RR) para establecer configuraciones de ruteo robustas y eficientes que reducen los impactos de las anomalías del volumen sobre el funcionamiento global de la red. Distintas variantes de RR han sido propuestas y analizadas, incluyendo no sólo técnicas de reconfiguración de ruteo sino también de balance de carga. Estas propuestas no sólo reducen los problemas de congestión inducidos por las anomalías de volumen, sino que también proporcionan una mejor utilización de los recursos de la red desde una perspectiva de Calidad de Servicio (QoS), una propiedad fundamental para mantener los servicios de red funcionando correctamente incluso en presencia de anomalías de tráfico.

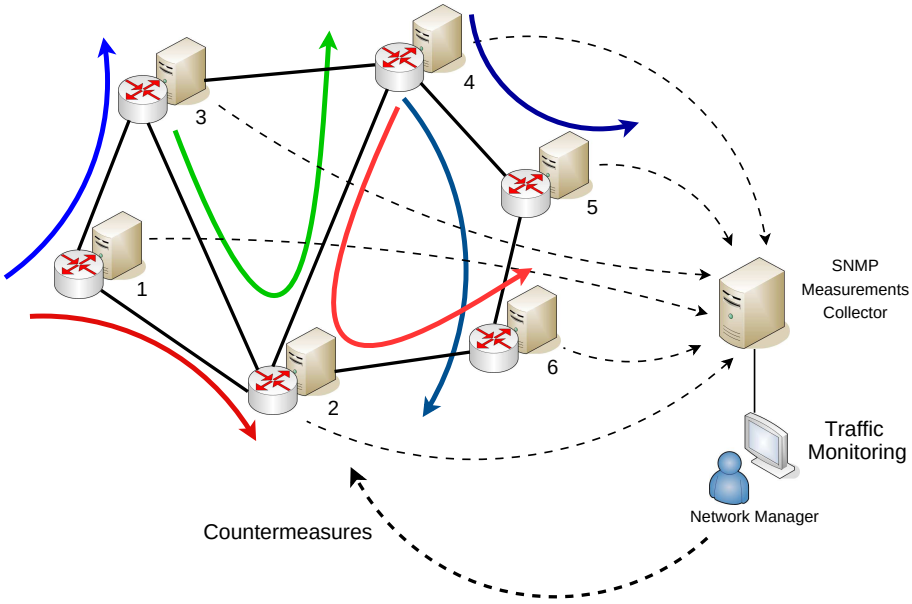


Figure 4.31 — Monitoreo de la Matriz de Tráfico Intra-dominio.

La figura 4.31 describe el contexto adoptado para el problema de monitoreo y análisis de tráfico abordado en la tesis. En resumen, proponemos analizar el tráfico de red en redes de alto porte, detectando anomalías de volumen en la Matriz de Tráfico a partir de medidas SNMP de tráfico de enlace. Además, proponemos identificar los orígenes de las anomalías detectadas, desplegando medidas de respuesta apropiadas basadas en técnicas de reconfiguración de ruteo robusto y balance de carga.

Aportes de los Trabajos de Tesis

Por las múltiples razones presentadas previamente, creemos que los sistemas de monitoreo en gran-escala deben apuntar a construir vistas globales del tráfico de red a partir de medidas limitadas y de bajo costo de implementación, combinándolas con algoritmos de análisis estadístico inteligentes y eficientes. Al mismo tiempo, estos sistemas deben ser capaces de detectar y localizar rápidamente las distintas anomalías de tráfico en la red, respondiendo con medidas apropiadas que permiten mantener las funciones de la red con un nivel de desempeño razonable. Una implementación confiable de tal grupo de técnicas sería altamente beneficiosa para los operadores de red, proporcionando un mecanismo liviano y fácil de desplegar para monitorer el tráfico de red en redes de alto porte.

Los trabajos presentados en esta tesis ofrecen contribuciones importantes en tres dominios distintos del área de redes de datos, relacionados con el monitoreo y el análisis del tráfico de red en redes de alto porte: (i) modelado y estimación de la Matriz de Tráfico, (ii) detección y localización de anomalías de volumen en la Matriz de Tráfico a partir de medidas agregadas SNMP, y (iii) optimización de ruteo en presencia de tráfico de red muy variable e incierto. A pesar de la extensa literatura disponible en estos tres dominios, la cual es analizada en profundidad en los capítulos 2, 3, y 4 de la tesis, nuestros estudios muestran que a la fecha no existe una técnica completa que permita detectar y localizar anomalías de volumen en la Matriz de Tráfico de una red de alto porte de forma óptima y a partir de medidas SNMP, desplegando en respuesta medidas de reconfiguración de ruteo y de balance de carga basadas en Calidad de Servicio.

El primer aporte importante de la tesis está relacionado con el modelado estadístico de la Matriz de Tráfico. Hemos desarrollado un nuevo modelo paramétrico, lineal, y de bajas dimensiones para describir el comportamiento libre de anomalías de una TM en una red IP de alto porte. Este modelo de tráfico tiene varias aplicaciones y presenta distintas ventajas con respecto a los modelos previamente propuestos para la TM: (i) al ser un modelo de bajas dimensiones permite solucionar los problemas de observabilidad de la TM al usar medidas SNMP, permitiendo en particular resolver el conocido problema de Estimación de la Matriz de Tráfico (TME), el cual se introduce en el capítulo 2. (ii) Distinto de muchos modelos basados en medidas de tráfico, el nuestro es paramétrico y estable a lo largo del tiempo, lo cual permite diseñar métodos de detección de anomalías confiables. (iii) El modelo utiliza exclusivamente medidas SNMP para construir una imagen precisa de la TM, simplificando cuestiones prácticas. Finalmente y más importante, (iv) este modelo lineal y de bajas dimensiones permite filtrar el tráfico libre de anomalías del problema de detección, lo cual permite calcular residuos sensibles a las distintas anomalías de volumen. Esta característica a-priori simple nos ha permitido diseñar algoritmos óptimos de detección y localización de anomalías de volumen en el tráfico de la TM. Este punto es estudiado a fondo en el capítulo 3.

Nuestros estudios en el área de modelado estadístico de la TM también han producido resultados interesantes en el problema de estimación de la TM, donde hemos introducido varias mejoras a dos técnicas de estimación previamente introducidas, mejorando diversos problemas de base de dichas técnicas que a la postre ofrecen mejores resultados. En particular, hemos propuesto dos técnicas mejoradas de TME, la primera basada en filtros de Kalman y la segunda basada en técnicas de aprendizaje estadístico.

La segunda contribución de la tesis concierne la detección y la localización de anomalías de volumen en la TM, utilizando medidas SNMP como datos de entrada. Utilizando el modelo paramétrico de bajas dimensiones descrito previamente, hemos propuesto dos algoritmos distintos para detección y localización de anomalías de volumen, con una ventaja mayor respecto de las propuestas anteriores, que consiste en condiciones de optimalidad bien demostradas. Esta característica generalmente ausente en los trabajos previos es fundamental para el desarrollo de algoritmos generales, desligados de una red o de un conjunto de redes en particular y más importante aún, independientes de evaluaciones particulares sobre condiciones especiales de tráfico. Los métodos de detección "caseros" pueden funcionar correctamente en ciertos escenarios, pero sin una base teórica sólida y generalizable, sus resultados son de validez limitada.

El primero de estos algoritmos fue diseñado para la detección óptima de anomalías de volumen, maximizando la tasa de detección correcta para una tasa de falsas alarmas acotada. El segundo algoritmo permite simultáneamente detectar y localizar un flujo OD anómalo dentro de TM, minimizando el máximo retardo promedio de detección y localización para una tasa de falsa localización y una tasa de falsas alarmas acotadas.

El tercer aporte principal de esta tesis está relacionado con la optimización y la reconfiguración del ruteo y del balance de carga intra-dominio en una red de alto porte, en condiciones de tráfico de red altamente variable. Motivados por el buen desempeño de un nuevo paradigma de optimización de ruteo bajo incertidumbre llamado Ruteo Robusto (RR), hemos estudiado en profundidad su posible aplicación como medida de respuesta frente a las anomalías de volumen detectadas. Nuestros estudios revelaron diversos defectos presentes en las técnicas actuales de RR para manejar grandes y abruptas variaciones de tráfico de forma eficiente, y diversas soluciones han sido propuestas. En primer lugar, hemos desarrollado dos variantes del método de base para reconfigurar el ruteo intra-dominio de la red, la primera basada en una extensión multi-hora de RR, y la segunda basada en una técnica proactiva para calcular a priori configuraciones de ruteo óptimas en presencia de anomalías de volumen. En segundo lugar, hemos analizado nuevos criterios de optimización para calcular configuraciones de RR con prestaciones de QoS. Finalmente, hemos explorado el paradigma de Balance Dinámico de Carga (DLB) en redes intra-dominio, proporcionando un profundo análisis comparativo entre las distintas técnicas de RR desarrolladas y diversos mecanismos de DLB en presencia de tráfico altamente variable.

Para verificar la aplicabilidad de nuestras contribuciones en redes operacionales reales, todos los algoritmos propuestos en la tesis fueron validados usando datos verdaderos de tráfico de diversas redes IP de alto porte. Al mismo tiempo, su desempeño ha sido comparado contra trabajos de renombre en cada uno de los dominios tratados, obteniéndose resultados similares o mejores en la mayoría de los casos. A modo de resumen, la siguiente lista presenta las contribuciones más importante de esta tesis:

- Un nuevo modelo paramétrico, lineal, y de bajas dimensiones para analizar el comportamiento normal (libre de anomalías de volumen) de la Matriz de Tráfico de una red IP de alto porte.
- Nuevos métodos para la estimación eficiente de la Matriz de Tráfico de una red IP de alto porte.
- Un método para detectar anomalías de volumen en la Matriz de Tráfico a partir de medidas agregadas. Este método presenta condiciones de optimalidad bien establecidas en términos de tasa de detección correcta y tasa de falsas alarmas.
- Un método para detectar y localizar rápidamente anomalías de volumen en la Matriz de Tráfico a partir de medidas agregadas. Este método presenta condiciones de optimalidad bien establecidas en términos de retardo de detección y localización, así como también respecto de la localización errónea y de la tasa de falsas alarmas.
- Una extensión de Ruteo Robusto Multi-Hora, la cual permite adaptar la configuración de ruteo a las variaciones normales del tráfico en la red de manera más eficiente que la provista por el enfoque original.
- Una nueva técnica de optimización de Ruteo Robusto, mejorada para proveer configuraciones de RR con prestaciones de QoS.
- Un método reactivo de Balance Robusto de Carga, orientado a contrarrestar los efectos negativos de las anomalías de volumen sobre el desempeño global de una red IP de alto porte.
- Un estudio comparativo de las virtudes y defectos de distintas técnicas de Ruteo Robusto y de Balance Dinámico de Carga.

Para concluir, quisiera indicar que las diversas contribuciones de esta tesis son el resultado de diversos trabajos de colaboración conjunta realizados entre los años 2006 y 2009 con varios profesores e investigadores de distintas instituciones. En particular, las contribuciones relacionadas con el modelado y la estimación de la TM, y las asociadas a la detección y localización de anomalías del volumen en la TM, son el resultado de trabajos conjuntos con el profesor adjunto Lionel Fillatre y el profesor Igor Nikiforov (Université de Technologie de Troyes), y con el profesor Thierry Chonavel (Télécom Bretagne). Las contribuciones relativas a la optimización de ruteo y de balance de carga en redes intra-dominio son el resultado de trabajos conjuntos con el profesor Walid Ben-Ameur (Télécom & Management SudParis), el profesor adjunto Hervé Kerivin (Clemson University), el investigador asociado de investigación postdoctoral Federico Larroca y el profesor adjunto Jean-Louis Rougier (Télécom ParisTech).

Estructura y Distribución de la Tesis

Los trabajos desarrollados en los tres dominios de estudio mencionados se presentan a lo largo de tres capitulos. La figura 4.32 muestras la organización de la tesis y la interacción entre los distintos capítulos.

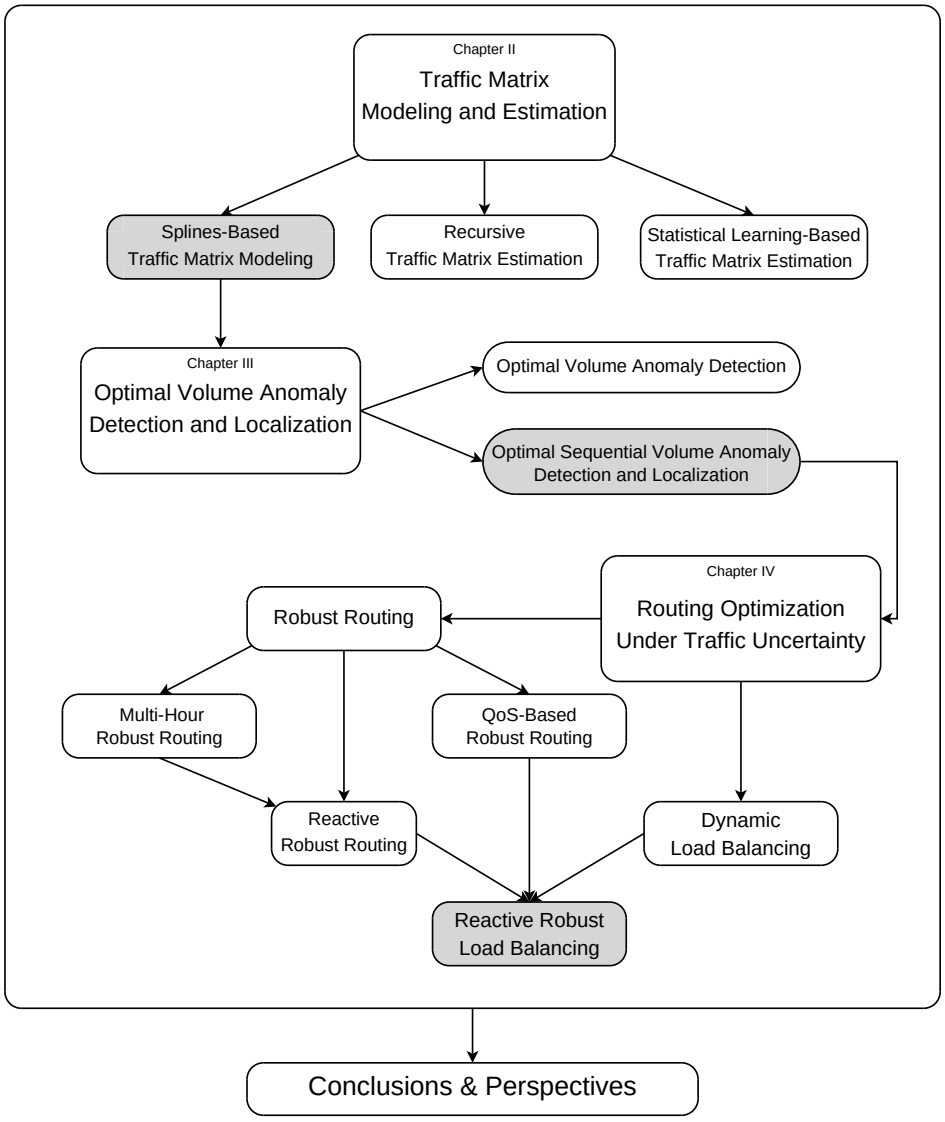


Figure 4.32 — Esquema estructural de la tesis.

El capítulo 2 presenta nuestros estudios en el área de modelado y estimación de la Matriz de Tráfico. Tres modelos para analizar la TM de una red de alto porte a partir de medidas SNMP se presentan y evalúan. El primero de ellos consiste en técnicas de modelado polinómial en bajas dimensiones, el segundo consiste en modelado de sistemas lineales y métodos recursivos de estimación, y el tercero se basa en técnicas de aprendizaje estadístico. En este capítulo se analyzan otras técnicas de modelado y estimación de la TM presentes en la literatura, las cuales son utilizadas como referencia para la evaluación del desempeño de nuestras propuestas.

El capítulo 3 presenta el diseño y la evaluación de dos algoritmos óptimos para detección y localización de anomalías de volumen en la TM, utilizando los principios de la teoría de la decisión. Ambos algoritmos utilizan el modelo de tráfico de bajas dimensiones presentado en el capítulo 2 para filtrar el tráfico de operación normal fuera del problema de detección. Los algoritmos de detección y localización de anomalías en la TM más representativos en la literatura son también presentados y analizados en este capítulo. Finalmente se presenta una evaluación comparativa entre estos algoritmos de referencia y nuestros algoritmos, considerando no sólo el desempeño de detección y localización, sino también la complejidad numérica de cada algoritmo y otras cuestiones relativas a la implementación de los mismos.

El capítulo 4 presenta el estudio de los paradigmas de Ruteo Robusto y Balance Dinámico de Carga en redes de alto porte. Diversas variantes y mejoras a los métodos tradicionales son propuestas y evaluadas en este capítulo. Finalmente, se presenta el diseño de un método completo de Balance Robusto de Carga con prestaciones de QoS, utilizando uno de los algoritmos de detección y localización de anomalías presentado en el capítulo 3.

Por último, se presentan las conclusiones sobre los trabajos de tesis desarrollados, así como también distintas perspectivas de trabajo a futuro y posibles líneas de investigación a explorar.