UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Informática

Intelligent Route Control for Inter-domain Routing

(Controlo Inteligente de Percurso para Encaminhamento Inter-domínio)

Alexandre José Pereira Duro da Fonte

Thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Informatics Engineering

(Tese submetida para obtenção do Grau de Doutor em Engenharia Informática)

Coimbra, September 2011

Intelligent Route Control for Inter-domain Routing

(Controlo Inteligente de Percurso para Encaminhamento Inter-domínio)

Thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Informatics Engineering

(Tese submetida para obtenção do Grau de Doutor em Engenharia Informática)

Alexandre José Pereira Duro da Fonte

Thesis Supervisor: Professor Edmundo Monteiro (University of Coimbra)

Co-supervisor: Professor Jordi Domingo-Pascual (Technical University of Catalonia)

This research was supported by PRODEP III program

União Europeia

Fundo Social Europeu

Coimbra, September 2011

Dedicated to

the memory of my mother

Abstract

The Internet is being increasingly criticized for the poor Quality of Service (QoS) guarantees it provides across the boundaries of Autonomous Systems (AS). The Border Gateway Protocol (BGP) protocol lies at the root of this problem, because its route discovery and path selection mechanisms are agnostic with regard to any performance or QoS metrics.

Underlying the work carried out in this thesis is the identification of the unresolved issues concerning the inter-domain QoS Routing (QoSR), and a survey of the approaches that have been adopted to tackle this problem. Most of the solutions that have been devised for this end, adopt an approach that involves extending BGP with a new QoS route attribute to carry the network state information within update messages. However, the QoS Extensions to BGP (q-BGP) failed to address the most critical issues, such as simplicity and scalability, deployment costs, fast path failover, and the end-user network or application-centered QoS.

A promising traffic control technique that can be able to satisfy these basic requirements, and has been increasingly adopted by multi-homed stub ASs to deal with BGP flaws, is to perform Intelligent Route Control (IRC). More precisely, this means that special network middleboxes, which are called IRC controllers, supplement the functionality of BGP, so that the active path used by the stubs to deliver traffic to remote peers, can be selected in short timescales, while actively monitoring the end-to-end performance or the quality of all the available paths.

The primary objective of this thesis is to study the intelligent route control for inter-domain routing and address some of the key challenges raised by adopting this strategy. The first issue is to be aware that conceiving an IRC system correctly is a challenging task, as currently there are no standard design guidelines. The second is that IRCs may oscillate, especially when a large number of IRCs compete for the same network resources. The third is that selfishness is a common characteristic of the IRCs, as well as of the backbone Traffic Engineering (TE) middleboxes, since the paths are greedily selected by each box, which if there is a misalignment of traffic goals can cause performance degradation of the traffic.

This thesis provides the design of an IRC controller and report results on the development of its mechanisms. The functional architecture that has been devised for the

IRCs, draws on a set of requirements for inter-domain QoSR and aims at addressing the challenges that they raise. Path monitoring, and path switching algorithms and routing decision policies are devised and evaluated through simulation. This includes a formal analysis of the time needed by IRCs to failover, together with an outline of two mechanisms for handling lost probes: a revision of the Jacobson's algorithm for congestion control and an algorithm based on the box-plot descriptive statistical tool. Since IRCs may oscillate persistently, we also investigate a number of ways of dealing with this issue (randomized path switching, randomized path monitoring and history-aware path switching).

The evaluations show to what extent of the proposed approach and mechanisms employed in this thesis are applicable. First, the results show the feasibility of basing end-user network or application-centric routing on the IRC technique, to improve inter-domain QoS. Compared with a BGP-based approach, this technique achieves a better performance with respect to the latency and efficiency of traffic transfers. Second, the results show that when IRCs are blended with the Jacobson or box-plot-based algorithms, they can adapt their timeout timers to network conditions, and above all, protect them against spurious timeouts. Finally, the results show that adding randomness to the route control process is the most effective solution to avoid IRC oscillations, whereas the use of sophisticated IRC algorithms, such as history-aware path switching, is questionable, since they require additional tuning.

This thesis also includes the provision of a Social IRC (SRC) strategy for competitive environments. This is a joint work undertaken with the Technical University of Catalonia (UPC) under the auspices of E-NEXT and CONTENT 6º FP IST NoE projects. The simulations show that a simple enhancement of randomized IRCs, like endowing them with a SRC algorithm supported by adaptive filtering techniques, makes it possible to drastically reduce the number of path switches needed to achieve the desired performance bounds of the traffic.

Finally, this thesis provides a new cooperative framework to balance the divergent interests between stub and transit networks, which is called an Internet Service Provider (ISP)-friendly IRC COOPerative framework (COOP). This is based on a cooperative strategy between direct neighbors and a feedback method, and is, as far as we know, the first successful strategy that has been employed to tackle the problem of the interactions between the IRC and TE boxes. The evaluation results show the feasibility of applying this framework, since it meets the traffic goals of each party, while ensuring the stability of IRC. Furthermore, it has the added advantage of producing synergistic interactions between both traffic control boxes.

We believe that this thesis can help to bring about significant advances in the state-of-the-art of intelligent route control. It is also hoped that the study will provide valuable guidelines for the design of similar systems and further inter-domain QoSR and traffic engineering mechanisms in the future.

Resumo

Um dos principais problemas da Internet é o fato de não oferecer garantias de Qualidade de Serviço (QdS) para além das fronteiras dos Sistemas Autónomos (SA). Na raiz do problema está protocolo BGP (Border Gateway Protocol), dado que a descoberta e seleção de percursos é agnóstica de métricas de desempenho ou de QdS.

Subjacente ao trabalho desenvolvido nesta tese, está a identificação das questões em aberto no encaminhamento inter-domínio com QdS (IDQdS) e o levantamento das abordagens adoptadas à resolução do problema. A maioria das soluções identificadas envolve a adição de extensões ao BGP para distribuição de informação sobre o estado da rede. Contudo, as extensões de QdS ao BGP (q-BGP) falham na resposta a requisitos essenciais, tais como: simplicidade e escalabilidade, contenção de custos, rápida detecção e recuperação de falhas, e suporte da entrega de tráfego centrada nos requisitos de desempenho fim-a-fim ou QdS dos AS nos extremos ou da aplicação.

Uma técnica de controlo de tráfego que promete satisfazer estes requisitos básicos que tem vindo a ser adoptada pelos SAs com múltiplas ligações à Internet para lidar com a ineficiência do BGP, é o Controlo Inteligente de Percurso (CIP). Esta envolve o uso de caixas especiais, designadas por controladores ou sistemas CIP com o objectivo de complementar a funcionalidade do BGP, por forma a que a ligação ativa para transmissão do tráfego seja selecionada enquanto é monitorizado o desempenho fim-a-fim ou QdS oferecidos por cada ligação.

O principal objectivo desta tese é estudar o CIP e endereçar alguns dos desafios chave suscitados pela adopção desta técnica. O primeiro relaciona-se com a inexistência de recomendações padrão para a concepção de um sistema completo de CIP. O segundo é lidar com o problema da oscilação dos percursos, causado pela sincronização dos sistemas CIP. O terceiro deve-se ao facto dos CIP e as ferramentas de engenharia de tráfego inter-domínio procurarem atingir os objetivos de tráfego de forma egoísta, sendo que uma acentuada divergência entre os mesmos pode conduzir à degradação do desempenho do tráfego.

Esta tese fornece a concepção de um sistema CIP e reporta os resultados do desenvolvimento dos seus mecanismos. A arquitetura funcional foi concebida com base num conjunto de requisitos para encaminhamento IDQdS, e visa enfrentar os desafios do CIP. Os mecanismos para monitorização e comutação inteligente de percurso são concebidos e avaliados. É incluída uma análise do tempo necessário a um CIP para contornar a falha de um percurso, conjuntamente com dois algoritmos desenhados para lidar com a perda dos pacotes de prova aos percursos: uma revisão do algoritmo de Jacobson para o controlo de congestão e um algoritmo baseado na ferramenta de análise estatística, a caixa de bigodes. Para se restringir as oscilações nos percursos, são estudadas várias estratégias para se lidar com esta questão (introdução de um factor de acaso na seleção do percurso ou na geração de pacotes de prova, e comutação de percursos com base num histórico).

As avaliações mostram até que ponto o CIP e os mecanismos empregues são válidos. Primeiro, os resultados mostram a viabilidade do CIP na melhoria da QdS inter-domínio, incluindo os objectivos de tráfego dos ASs extremo ou da aplicação. Comparado com o BGP, o CIP atinge melhor desempenho no que diz respeito à latência e eficiência das transferências de tráfego. Em segundo lugar, os resultados mostram a eficácia de se combinar um CIP com um algoritmo do tipo Jacobson ou baseado na caixa de bigodes, sobretudo na redução do número de detecções intempestivas. Finalmente, os resultados mostram a eficácia das várias estratégias CIP para redução das oscilações, sendo que a introdução do factor de acaso na seleção de percurso a mais promissora, sem ser necessário uma calibração adicional.

Esta tese fornece também a concepção de um modelo de controlo de percurso sociável para ambientes competitivos, resultante do trabalho conjunto com a Universidade Politécnica da Catalunha, realizado no âmbito dos projetos E-NEXT e CONTENT do 6º Programa Quadro. Neste modelo, cada CIP implementa um algoritmo de controlo de percurso social que de forma adaptativa restringe o seu egoísmo. Os resultados de simulações, mostram que dotando-se um CIP com um algoritmo sociável suportado por uma técnica de filtragem adaptativa, permite uma drástica redução das oscilações.

Finalmente, nesta tese é concebido o esquema – CIP COOPerativo (COOP) – amigo dos fornecedores de serviço Internet, o qual permite conciliar os interesses divergentes do CIP e da engenharia de tráfego inter-domínio. O COOP propõe um desenho conjunto, com base numa estratégia de cooperação e num método de realimentação. Os resultados das avaliações mostram que o COOP supera o cenário em que os CIP e as caixas de engenharia tráfego inter-domínio operam de forma autônoma e egoísta. Para além das vantagens oferecidas pelos CIP, os resultados revelam uma redução efetiva da penalização sobre desempenho da engenharia de tráfego originada pelos CIP, enquanto é assegurada a estabilidade do CIP. Em suma, o COOP pode produzir uma interação sinérgica entre ambos os mecanismos, dado que são satisfeitos os objectivos de tráfego de cada parte, incluindo a estabilidade do CIP.

Espera-se que esta tese forneça avanços significativos ao estado da arte do controlo inteligente de percurso e valiosas linhas de orientação à concepção de sistemas similares, e demais futuros protocolos de encaminhamento e mecanismos de engenharia de tráfego inter-domínio.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Edmundo Monteiro for his guidance and valuable advice. His wit, intelligence and pragmatic outlook were a constant stimulus and encouragement.

My deepest thanks, too, to a number of UPC (Technical University of Catalonia) researchers, in particular, my co-supervisor Professor Jordi Domingo-Pascual, Dr. Marcelo Yannuzzi, and Dr. Xavi Masip-Bruin for their insightful discussions and for their collaboration on research projects.

I am grateful to Dr. Marilia Curado for her encouraging, advice and help in preparing our joint papers, and for devoting time to proofreading and supplying me with valuable feedback.

Sincere thanks should also be given to the various researchers in the Communications and Telematics Group with whom I had the chance to share space and time at the Department of Informatics Engineering at the University of Coimbra. I would particularly like to thank Manuel Machado, and Thomas Bohnert for joining me in technical discussions.

The author wishes to thank the PRODEP III program for giving financial support to this research study, and his colleagues at the Department of Computer Engineering at the Polytechnic Institute of Castelo Branco. I would particularly like to express my sincere thanks to Dr. J.C. Metrôlho and my office colleagues, the future Drs. Vasco Soares and Paulo Neves, for creating such a good atmosphere and exchanging of views in the course of my work, as well as those who, directly or indirectly made a contribution to this study.

Last but not least, I would like to say thank you to a most special friend for her unconditional support and encouragement on this journey. And, finally my deepest gratitude to my father and brothers.

Foreword

The work described in this thesis was conducted at the Laboratory of Communication and Telematics of the Centre for Informatics and Systems of the University of Coimbra under the auspices of the following projects, and their European doctoral schools:

- E-NEXT, 6º FP IST NoE (Proposal/Contract no.: FP6-506869) – European network of excellence of the 6th framework program on Emerging Networking Experiments and Technologies (from January 2004 to June 2006). In the context of this project and as a result of the work described in this thesis, there was a participation in Working Group 4, concerning routing and traffic engineering.
- CONTENT, 6º FP IST NoE (Proposal/Contract no.: FP6- 0384239) – European network of excellence of the 6th framework program on Content Networks and Services for Home Users (from May 2006 to July 2009). In the context of this project and as a result of the work described in this thesis, there was a participation in the Technical Activity 1 (TA1), concerning routing mechanisms for delivering end-to-end quality of service.
- SATIN-EDRF Award (E-NEXT) – School on Advanced Topics In Networking (SATIN) – European Doctoral Research Foundation (EDRF) of E-NEXT, 6º FP IST NoE. Title of the research: An overlay approach for Inter-domain QoS routing: coordination mechanisms and signalling protocols between Overlay Entities.
- SATIN-EDRF Award (CONTENT) – School on Advanced Topics In Networking (SATIN) – European Doctoral Research Foundation (EDRF) of CONTENT, 6º FP IST NoE. Title of the research: Improving Inter-AS Quality of Service through Multihoming Smart Routing: Tackling the Interaction with Traffic Engineering.

The work done during this thesis resulted in the following publications authored or co-authored by the candidate:

Journals and book chapters

- Fonte, A. and Machado, M. and Curado, M. and Monteiro, E. and Boavida, F. , "Combining Intelligent Route Control with Backbone Traffic Engineering to Deliver Global QoS-Enabled Services," chapter 9 in Recent Advances in Providing

QoS and Reliability in the Future Internet Backbone, Nova Science Publisher, 2011 1st quarter

- Fonte, A. and Curado, M. and Monteiro, E. , "Inter-domain Quality of Service Routing: Setting the Grounds for the Way Ahead," Journal Annals of Telecommunications, Special Issue on Inter-Domain Routing and QoS over Heterogeneous Networks: Utopia or Reality (Fall 2008), October 2008

- Fonte, A. and Monteiro, E. and Yannuzzi, M. and Masip-Bruin, X. and Domingo-Pascual, J. , "A Framework for Cooperative Inter-domain QoS Routing," chapter in EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World, IFIP International Workshop on Networked Applications, Colmenarejo, Madrid/Spain, 6-8 July, 2005, Series: IFIP International Federation for Information Processing, Vol. 196, March 2006

- Yannuzzi, M. and Masip-Bruin, X. and Marin-Tordera, E. and Domingo-Pascual, J. and Fonte, A. and Monteiro, E. , "Improving the Performance of Route Control Middleboxes in a Competitive Environment," IEEE Network, Vol. 22, no. 5, Sep./Oct. 2008

- Masip-Bruin, X. and Yannuzzi, M. and Domingo-Pascual, J. and Fonte, A. and Curado, M. and Monteiro, E. and Kuipers, F. and Mieghem, P. and Avallone, S. and Ventre, G. and Aranda-Gutiérrez, P. and Hollick, M. and Steinmetz, R. and Iannone, L. and Salamatian, K. , "Research challenges in QoS routing," Computer Communications, Vol. 29, pp. 563-581, March 2006

- Yannuzzi, M. and Fonte, A. and Masip-Bruin, X. and Monteiro, E. and Sànchez-López, S. and Curado, M. and Domingo-Pascual, J. ,"Encaminamiento Inter-dominio con Calidad de Servicio basado en Overlay Entities distribuidas y QBGP," Novática, Revista de la Asociación de Técnicos de Informática (ATI) del Concil of European Professional Informatics Societies (CEPIS), 2005

Conferences and Workshops

- Fonte, A. and Curado, M. and Monteiro, E. , "Prediction Error-based Criterion for Selecting Popular Destinations," to appear in the 3rd IEEE International Workshop on Management of Emerging Networks and Services (IEEE MENS 2011) in conjunction with IEEE GLOBECOM 2011, Houston, Texas, USA, December 2011

- Fonte, A. and Curado, M. and Monteiro, E. , "Stabilizing Intelligent Route Control: Randomized Path Monitoring, Randomized Path Switching or History-Aware Path Switching?," in Proceedings of the 11th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services

Management of Converged Multimedia Networks and Services (MMNS 2008), Samos Island, Greece, September 2008

- Fonte, A. and Pedro, M. and Monteiro, E. and Boavida, F. , "Analysis of Inter-domain Smart Routing and Traffic Engineering Interactions," in Proceedings of the IEEE Globecom 2007 Internet Protocol Symposium, Washington, DC, USA, November 2007

- Fonte, A. and Pedro, M. and Monteiro, E. and Boavida, F. , "Improving Inter-AS Quality of Service through Multi-homing Smart Routing: Tackling the Interaction with Traffic Engineering," in Tutorial and Ph.D. Student Workshop of Med Hoc Net 2007, Ionian University, Corfu, June 2007

- Fonte, A. and Monteiro, E. and Yannuzzi, M. and Masip-Bruin, X. and Domingo-Pascual, J. , "A Cooperative Approach for Coordinated Inter-domain QoSR Decisions," in Proceedings of the E-NEXT Doctoral Summer School (EUNICE 2005), University Carlos III of Madrid, Colmenarejo, Spain, July 2005

- Bohnert, T. and Monteiro, E. and Curado, M. and Fonte, A. and Ries, M. and Moltchanov, D. and Koucheryavy, Y. , "Internet Quality of Service: a Bigger Picture," in Proceedings of the 1st OpenNet Workshop - Service Quality and IP Network Business: Filling the Gap, Diegem/Brussels, Belgium, March 2007

- Yannuzzi, M. and Masip-Bruin, X. and Sànchez-López, S. and Domingo-Pascual, J. and Fonte, A. and Curado, M. and Monteiro, E. , "From standalone to collective intelligent route control," in Proceedings of the IEEE INFOCOM 2006 Student Workshop, Barcelona, April 2006

- Yannuzzi, M. and Fonte, A. and Masip-Bruin, X. and Curado, M. and Domingo-Pascual, J. and Monteiro, E. and Sànchez-López, S. , "On the Advantages of Cooperative and Social Smart Route," in Proceedings of the ICCCN2006, Fifteenth International Conference On Computer Communications and Networks, Arlington, Virginia, USA, October 2006

- Yannuzzi, M. and Fonte, A. and Masip-Bruin, X. and Monteiro, E. and Sànchez-López, S. and Domingo-Pascual, J. , "A Self-Adaptive QoS Routing Framework for Multi-homed Stub Autonomous Systems," in Proceedings of the E-NEXT Doctoral Summer School (EUNICE 2005), University Carlos III of Madrid, Colmenarejo, Spain, July 2005

- Yannuzzi, M. and Fonte, A. and Masip, X. and Monteiro, E. and Sànchez-López, S. and Curado, M. and Domingo-Pascual, J. and Sole-Pareta, J. , "Encaminamiento Inter-Dominio con Calidad de Servicio basada en una Arquitectura Overlay y QBGP," in Proceedings of XIV Jornadas TELECOM I+D 2004, Madrid, Spain, November 2004

- Yannuzzi, M. and Fonte, A. and Masip-Bruin, X. and Monteiro, E. and Sànchez-López, S. and Curado, M. and Domingo-Pascual, J. , "A proposal for inter-domain QoS routing based on distributed overlay entities and QBGP," in Proceedings of the First International Workshop on QoS Routing (WQoSR'2004), Barcelona, Springer-Verlag October 2004

At the time of the submission of this thesis, parts of Chapters 3 and 5 are under submission for publication:

Working-papers

- Fonte, A. and Curado, M. and Monteiro, E. , "Protecting Intelligent Route Control against Spurious Timeouts," LCT-Working-paper, March 2011
- Fonte, A. and Curado, M. and Monteiro, E. , "An ISP-friendly Intelligent Route Control Cooperative Framework," LCT-Working-paper, July 2010

Contents

List of Figures

List of Tables

Acronyms

A-RED	Adaptive Random Early Detection
A/D	Analog-to-Digital
ABI	Available Bandwidth Index
AC	Admission Control
AF	Assured Forwarding
AFI	Address Family Identifier
AIMD	Additive Increase Multiplicative Decrease
ARIMA	AutoRegressive Integrated Moving-Average
AS	Autonomous System
ASN	Autonomous System Number
Avail-Bw	Available Bandwidth
BB	Bandwidth Brokers
BE	Best Effort
BGP	Border Gateway Protocol
BRITE	Boston University Representative Internet Topology gEnerator
BRP	Bandwidth Restricted Path
CB	Choose-Best
CCDF	Complementary Cumulative Distribution Functions
CG	Choose-Good
CIDR	Classless Inter-Domain Routing

CLVL	Controlled-Loss Virtual Link
CO	Capacity Over-provisioning
COOP	ISP-friendly IRC COOPerative framework
COOPA	COOP Advertisement
CoS	Class of Service
CP	Controlled Proactivity
CPU	Central Processing Unit
CSP	Content Service Providers
DCLC	Delay-Constrained Least Cost
DCUR	Delay Constrained Unicast Routing
DEBF	Dual Extended Bellman-Ford
DiffServ	Differentiated Services
DLV	Deterministic Last Value
DNS	Domain Name Service
DoP	Degree of Preference
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ERS	Egress Router Selection problem
EWMA	Exponential Weighted Moving Average
FBI	Flow Balance Index
FIB	Forwarding Information Base

FIFO	First In First Out
FP	Fully Proactive
FSM	Finite State Machine
GPT	General Purpose Transport
GRE	Generic Routing Encapsulation
iBGP	internal BGP
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IntServ	Integrated Services
IOS	Internetwork Operating System
IP	Internet Protocol
IQR	Inter Quartile Range
IRC	Intelligent Routing Control/Intelligent Route Controller
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
LARAC	LAgrange Relaxation based Aggregated Cost
LB	Load-Balancing
LpEMA	Low-pass Exponential Moving Average
LSDB	Link State DataBase
LV	Last-Value
MA	Moving Average
MAU	Multi-Attribute Utility
MC	Metrics Combination
MC(O)P	Multi-Constrained (Optimal) Path

MCP	Multi-Constrained Path
MED	Multi-Exit-Discriminator
MEFPA	Multi-constrained Energy Function based Pre-computation Algorithm
MIAD	Multiplicative Increase Additive Decrease
MLU	Maximum Link Utilization
mMLU	minimization of the Maximum Link Utilization
MMM	Monitoring and Measurement Module
MOS	Mean Opinion Score
MPLS	MultiProtocol Label Switching
MRAI	MinRouteAdvertisementIntervalTimer
MTT	Maximum Tolerable Threshold
NID	Network Information Database
NLRI	Network Layer Reachability Information
NN	Neuronal Networks
NP-hard	Non-deterministic Polynomial-time hard
OB	Overlay Broker
OC	Optical Carrier
OE	Overlay Entities
OPCA	Overlay Policy Control Architecture
OPP	Overlay Policy Protocol
OSN	Overlay Service Network
OSPF	Open Shortest Path First
OWD	One-Way Delay
OWL	One-Way-Loss
PA	Policy Agents
PAD	Policies and Algorithms Database

PHB	Per-Hop Behavior
PM	Path Monitor
PoP	Point of Presence
PPA	Probe Packet Acknowledgment
PPR	Probe Packet Request
PQE	Path Quality Estimator
PRC	Proactive Route Control
PS	Path Shifts
q-BGP	qos-enabled BGP
QoS	Quality of Service
QoS_NLRI	Quality of Service Network Layer Reachability Information
QOSPF	QoS routing mechanisms and OSPF extensions
RCE	Route Control Engine
RCM	intelligent Route Control Module
RES	link RESource advertisement message
RIB	Routing Information Base
RIFT	Risk, Interference and Application Fit
RIP	Routing Information Protocol
RMAP	autonomous system Topology and Relationship MAPper
RMON	Remote Network MONitoring
RR	Resource Record
RRA	Resource Reservation Advertisement message
RRC	Reactive Route Control
RSP	Restricted Shortest Path
RSVP	Resource ReSerVation Protocol
RTO	Request Time-Out

RTT	Round-Trip Time
RVM	Reporting and Viewer Module
SAFI	Subsequent Address Family Identifier
SAMCRA	Self-Adaptive Multiple Constraint Routing Algorithm
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SLS	Service Level Specification
SMART	Simple Multi-Attribute Rating Technique
SNMP	Simple Network Management Protocol
SPP	Stable Paths Problem
SPT	Special Purpose Transport
SRC	Sociable Route Control
SWP	Shortest-Widest Path
TAMCRA	Tunable Accuracy Multiple Constraints Routing Algorithm
TCP	Transmission Control Protocol
TD	Traffic Demands
TE	Traffic Engineering
TF	Traffic Forecaster
TRD	Traffic Requirements Database
VoIP	Voice over IP
VPN	Virtual Private Network
WSP	Widest-Shortest Path

Chapter 1

Introduction

This dissertation addresses the problem of the Quality of Service (QoS) routing across Autonomous Systems (ASs) boundaries. In Section 1.1 of this chapter, there is an account of the reasons for undertaking a study of this field. In addition, there is a discussion of the general problem of QoS routing in today's routing system. Section 1.2 defines the scope of this thesis and the target approach adopted for the research. The objectives and contributions of this thesis are outlined in Section 1.3. In the final of this chapter, Section 1.4 provides a brief road map which is designed to help the reader navigate through the thesis.

1.1 The Problem of QoS Routing

The Internet of today has become a global communications system. More and more telecommunication networks and applications are migrating to the Internet Protocol (IP)-based infrastructure [2–4]. The main reasons for employing IP are cost savings, easier network maintenance, and, above all, improvements in business productivity resulting from the convergence of the applications. However, this increasing interest in IP-based applications, such as IP Virtual Private Networks (VPN) or Voice over IP (VoIP), has also led to greater demands for IP services that are able to support stricter Service-Level Specifications (SLS) in terms of end-to-end Quality of Service (QoS) guarantees.

Unfortunately, IP was originally conceived with little support to achieve QoS. IP basically supports QoS marking via a header field. To address the problem of this design flaw, in the 1990s, considerable efforts were made by the Internet Engineering Task Force (IETF) to replace the native best-effort service with the emerging reservation-based services (i.e., IntServ) and reservation-less services (i.e., DiffServ) [5,6]. In normal network conditions, such as when the paths and links are stable and the link congestion is not excessive, IntServ and DiffServ have proved to be effective architectures for stringent QoS. However, it cannot be assumed that will be the case when they are applied to real networks. In fact, several studies have shown that instability and failures are common events in the Internet, which prevent the wide-scale effectiveness of IntServ and DiffServ because these QoS architectures do not support any path discovery feature [7–9].

Extending the QoS concept to Internet routing has thus become the next target goal. The IETF QoS routing Working Group (WG) was created to define a framework and techniques for QoS routing, and identify the main underlying research issues [10]. The WG recognized that QoS routing is an essential missing piece of Internet functionality that is required to support reliable, assured and high performance IP traffic transferences, as well as the prospect of combining QoS routing with IntServ and DiffServ (if the compatibility between both models can be guaranteed) [11]. However, the WG was prematurely shut down in the late 1990s due to the lack of a deep understanding of the problem, an issue that has still not been resolved.

The main purpose of QoS routing is to select the best path that is able to improve the traffic performance [12]. To achieve this goal, a QoS-aware router must be able to compute optimal or feasible paths on the basis of the network resources information, and traffic requirements. A QoS-aware router must also employ a state-dependent routing protocol that gathers this network state information and keeps it up-to-date.

In the next two subsections, the QoS routing problem is explored in more detail, and the issues regarding with both components of the problem are discussed – the QoS routing algorithms and the protocols. This provides the background for the next sections where we outline the scope and objectives of this thesis. It should be stressed that apart from the QoS routing tasks, there are many other tasks that must also be performed by QoS-aware routers, such as admission control, resource reservation, scheduling of IP packets and so on, that are beyond the scope of this work.

1.1.1 QoS Routing Algorithms

The QoS path selection feature of QoS-aware routers seeks to find the best paths that can satisfy the QoS traffic requirements or constraints (e.g., latency, jitter and bandwidth). The path selection problem is known as the Multi-Constrained (Optimal) Path (MC(O)P) routing problem, and can be briefly stated as follows [13, 14]:

Definition 5.1: *Consider a network $G(N, E)$, where N represents the set of nodes and E the set of edges. Each link (u, v) is characterized by k (≥ 2) non-negative weight functions $w_1(u, v) \in R^+, ..., w_k(u, v) \in R^+$, corresponding to k QoS measures. Given k constants represented by a vector $c = (c_1, ..., c_k)$, the Multi-Constrained Path (MCP) problem is to find a path p from a source node src to a destination node dst such $w_1(p) \leq c_1, ..., w_k(p) \leq c_k$, where $w_i(p) = \sum_{(u,v)\in E} w_i(u, v) \leq c_i$ for $i = 1, ..., m$. The MC(O)P is a problem that additionally finds a path $p = src \rightarrow v_1 \rightarrow v_2, ..., dst$, i.e., the*

Table 1.1: Worst-case time complexity of the considered QoS path selection algorithms.

Algorithm	**Metrics**	**Time**
WSP	Hop count or delay and bandwidth	$O(NE)$
SWP	Bandwidth and Hop count or delay	$O(NE)$
DCUR	Delay and Cost	$O(N^3)$
DEBF	Delay and Cost	$O(2NE)$
MEFPA	Multiple additive metrics	$O(b(E + ElogE + E))$
TAMCRA, SAMPRA	Multiple additive metrics	$O(pNlog(pN) + p^2kE)$
LARAC	Delay and Cost	$O(E^2log^4E)$

optimal path, so that there is no other path q where $w_1(q) < w_1(p), ..., w_k(q) < w_k(p)$.

Unfortunately, when the QoS measures are additive and/or multiplicative, independent, and use real numbers or unbounded integers as values, the MC(O)P problem is interpreted as NP-hard (Non-deterministic Polynomial time-hard), and hence is computationally intractable for large networks [15].

Several heuristics or approximation algorithms have been proposed to solve the MC(O)P in polynomial-time [12]. Table 1.1 compares the performance of all the path selection algorithms considered below, where k is the number of constraints, p is the number of shortest paths and b is the number of coefficient vectors.

Two instances of the MC(O)P are the focus of the heuristics or approximation algorithms, depending on the constraints to be meet. The first instance of the MC(O)P is when the MCP problem is defined as a Bandwidth Restricted Path (BRP) problem [16]. Examples of algorithm families that solve the BRP problem are the Widest-Shortest Path (WSP) and Shortest-Widest Path (SWP) algorithms [17–19].

The second instance of MC(O)P, called the Restricted Shortest Path (RSP), is when two additive metrics are used to characterize a link [15]. A widely-used case is the Delay-Constrained Least Cost problem (DCLC), in which cost and delay are used as link metrics [20]. Examples of algorithms that can be used for solving the DCLC problem are the Delay Constrained Unicast Routing (DCUR) and Dual Extended Bellman-Ford (DEBF) [21].

An alternative straightforward heuristic to solve the MC(O)P problems is by means of the link Metrics Combination (MC) into a single composite metric [22]. The difficulty in using this heuristic is how to find a proper means of weighting the QoS metric items together. As a result, linear, non-linear and Lagrange relaxation compositions have been widely used in several proposals, such as the Multi-constrained Energy Function based Pre-computation Algorithm (MEFPA), the Tunable Accuracy Multiple Constraints Routing Algorithm (TAMCRA) (and its successor SAMCRA (Self-Adaptive Multiple Constraint Routing Algorithm)) and the LAgrange Relaxation based Aggregated Cost (LARAC) algorithm, respectively [14, 23–26].

1.1.2 QoS Routing Protocols

The QoS routing protocol or signaling feature of QoS-aware routers is responsible for exchanging the network resource information between routers, which then is supplied to the QoS routing algorithm so that it can compute the best path. We will now turn to the issues regarding the routing protocols.

The Internet routing is handled by two kinds of protocols each of which has a distinct objective. Inside the boundary of an AS (Autonomous System), the routing of IP packets is handled by an Interior Gateway Protocol (IGP). There are two basic types of IGPs at present based on two paradigms: a) distance-vector routing and b) link-state routing. In distance-vector routing, a router distributes to its peers the vectors of cost/distance (arrays) to the other nodes in the network, and the selection of the best path is made on the basis of the Bellman–Ford routing algorithm [27, 28]. In turn, a link-state protocol distributes the entire network topology to all the routers by flooding the network with LSAs (Link State Advertisements). After this, when each router is aware of the complete topology of network (that is stored in a Link State DataBase (LSDB)), it computes the best path toward all the others with the aid of the Dijkstra's shortest path algorithm [29]. The Routing Information Protocol (RIP) is an example of a standard distance-vector protocol, while the Open Shortest Path First (OSPF) protocol is an example of a link-state protocol [30, 31]. On the other hand, to provide connectivity across ASs boundaries the de facto Exterior Gateway Protocol (EGP) is the BGP (Border Gateway Protocol), which is used to exchange and maintain the reachability information to blocks of IP prefixes [32].

The main difficulty resides in the fact that most of the original IP routing protocols have been designed without (or almost without) the QoS (or efficiency) requirement in mind. The problem of extending routing protocols to support QoS is challenging because of the changes that are required in the routers; and issues such as processing and communication overhead, convergence time, instability and QoS metric inaccuracy must be addressed [33]. Latter issues are hard to deal with because they are dependent on each other, and trade-offs are required. For instance, to address the communication overhead imposed by the load of the flooding process used in exchange network resource information (e.g., the available link bandwidth or link delay), the number of messages can be restricted by the limiting their frequency, or size by means of the aggregation of routing information [34–37]. However, low frequency and high aggregation are factors that raise new problems, namely information inaccuracy which can severely impair the performance of the QoS routing protocol because the routing decisions are made on the basis of obsolete information.

Nevertheless, at the intra-AS level, there has been significant progress in dealing with some of the issues mentioned above, and this has resulted in standards, such as QoS routing Mechanisms and OSPF Extensions (QOSPF) [18]. In addition, studies have shown QOSPF to be an effective protocol because the link state approach provides a complete view of the network; so that it is possible to employ fine path selection algorithms based on this approach [38]. The QoS extensions to OSPF enable the LSAs to include network resource information, and assume that a path selection algorithm – a QOSPF Dijkstra version – will take this information into account. QOSPF employs two types of LSAs to flood the network with the QoS link states – RES (link RESource advertisement message) and RRA (Resource Reservation Advertisement message). The RES enables routers to advertise incremental QoS link states in the network. To be more precise, it notifies routers with the largest amount of resources available for reservation, and the link delay. In turn, the RRA message enables routers to notify the resources that are used by a traffic flow. The key difference between QOSPF and the normal OSPF is that the links will be ignored if they do not have sufficient resources – the resources available plus those already reserved – for the flow.

1.2 The Scope of this Thesis and Approach Adopted

While there has been significant progress to enable the Internet to support Quality of Service (QoS) routing, so far most of the research has been focused on routing within a single Autonomous System (AS), as described earlier. When the traffic leaves the end-user network, there is no guarantee that the quality of service that is offered on the crossed domains will meet the traffic requirements, or have enough flexibility to control/select the paths that are able to circumvent the congested or unwanted ASs. Clearly, the main problem is thus how to expand QoS routing across multiple ASs boundaries, which is the concern of this thesis.

The Border Gateway Protocol (BGP) protocol lies at the root of the problem of a lack of QoS across the ASs boundaries, because its route discovery and path selection mechanisms are agnostic with regard to any performance or QoS metrics. The primary design goal of BGP was to enable the border routers to exchange Network Layer Reachability Information (NLRI), while enforcing local AS administrative routing policies. NLRI is included in BGP update messages and is composed of a length and a prefix (e.g., /25, 204.149.16.128), which with the list of ASs that the reachability information transverses (as carried out within the AS path attribute), ensure the end-to-end connectivity. However, this is not enough for the QoS objectives of the traffic to be attained.

To illustrate the lack of QoS support by BGP, Figure 1.1 displays the scenario where the stub `AS1` wants to send traffic to the sub-net 12.0.0.0/8 belonging to the remote stub `AS5`. On the basis of the lowest AS hop count criterion of the BGP decision process, the border routers of `AS1` would select the path `AS4-AS5` (learnt from `AS4`), as the default active path, rather than the alternate path `AS2-AS3-AS5` (learnt from

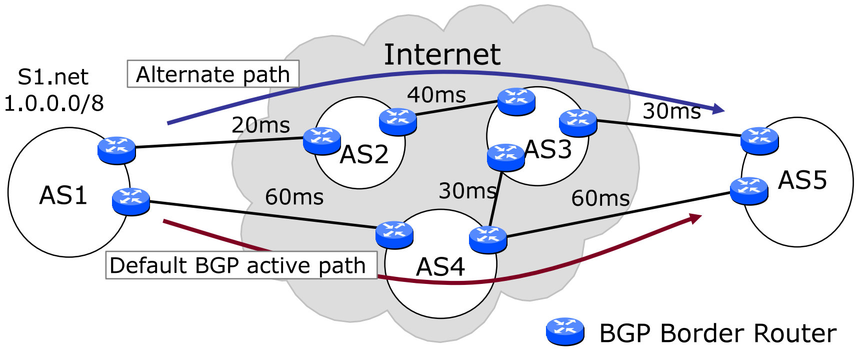


Figure 1.1: Illustration of BGP sub-optimality.

AS2). However, the problem of this choice is that the packets forwarded through the transit network AS4 would experience more delay – more 30ms – than if they were sent through the transit network AS2.

In a similar way to intra-domain routing, a QoS-aware path control is a key function for inter-domain QoS routing. However, before QoS can be deployed across the ASs boundaries, the role of the routing protocol itself has to be strengthened with other functions. It must be able to support the process of establishing QoS peering connections among ASs/providers. Ideally, it should provide an AS with the ability to advertise, discover and select services with specific QoS capabilities, such as a Class or meta-class of Service (CoS) and also to negotiate the terms of a service as Service Level Specifications (SLSs) [39–41]. To be more precise, a SLS contains in an abstract manner all the QoS-related parameters of the expected service, such as flow identification, traffic conformance, performance guarantees, service schedule, reliability among others.

Most of the solutions devised to implement inter-domain QoS [42–44] adopt an approach that involves extending BGP with a new capability [45]. New BGP attributes (e.g., QoS_NLRI) to carry the QoS information within BGP messages, and modifications in the BGP decision process have been suggested, so that they can enable the ASs to express higher QoS levels or to be discouraged from routing toward them [46–48]. Unfortunately, the QoS Extensions to BGP failed to address the most critical issues, which are as follows:

(i) **Simplicity and Scalability.** QoS Extensions to BGP (q-BGP) are well-suited to convey QoS information, but add too much complexity to the BGP software and data distribution mechanisms [49]. These solutions require that BGP routers must support stricter memory and CPU requirements to store the extra amounts of routing state information and to handle the processing overhead caused by the greater complexity of handling the update messages, and select and install the best routes. Moreover, the BGP software must support a multi-session capability so that it can provide routing for multiple CoSs. However, these solutions also run the risk of increasing the difficulty of managing the configuration of routers;

(ii) **Fast Path Failure Reaction and Recovery.** BGP reacts slowly to changes [50–52]. Neither BGP, nor its q-BGP versions, are thus prepared to react to failures/congestion in short timescales; this is due to the slow convergence problem of its routes and the high degree of computational complexity involved in the processing of route attributes and in the path selection. As a result, the end-to-end QoS of the traffic can be decreased significantly in the advent of such events.

(iii) **End-user or Application-centered QoS.** QoS Extensions to BGP only focus on performance objectives as viewed from the network's perspective of quality. To be more precise, although the transit ASs employing Traffic Engineering (TE) with q-BGP may be optimized to meet the local network objectives [42, 53], they provide suboptimal QoS levels (higher/lower) with respect to the real needs of end-user networks or applications. They lack the means to support the delivery of the end-user network or application-layer centered QoS. An additional drawback is that they do no respond to the needs of the users by giving them more route control. In effect, the BGP routing paradigm prevents any user from controlling the transit of its packets in the Internet.

Even after the difficulties of enabling BGP to support QoS routing on a large scale have been recognized, determining whether the inter-domain QoS routing requirements are best met by introducing changes into BGP or by rebuilding the entire Internet routing from scratch, remains an open question.

Underlying the work carried out in this thesis is the identification of the unresolved issues concerning inter-domain QoS routing and a survey of the main approaches that have been adopted to tackle this problem. As a result of this initial study, it can be concluded that neither of the approaches of extending and replacing BGP justifies changing the present routing infrastructure because there is a potential risk that they will increase the degree of complexity and deployment costs.

Before these difficulties can be overcome, there is a need to develop some cost-effective lightweight routing schemes aimed at providing better end-user network or application-layer QoS across ASs boundaries. The fact that they are lightweight means that the overhead of the proposed protocol or mechanisms (in terms of factors such as implementation and message complexity), should be no higher than that of the current BGP version protocol.

A promising approach that can be employed for attaining previous goal, and that has been increasingly adopted by multi-homed stub ASs to deal with BGP flaws, is to perform Intelligent Route Control (IRC) [54]. More precisely, this means that special network middleboxes, which are called IRC controllers, supplement the functionality of BGP, so that the active transit AS/path used by the stubs to deliver traffic to remote peers, can be selected on behalf of the BGP routers in short timescales, while actively monitoring the end-to-end performance or the quality of all the available paths.

Intelligent route control is not intended to replace BGP, but is well-suited to improve the performance of inter-domain routing, since it provides a holistic way of meeting end-user network/application layer traffic challenges. Moreover, it follows a philosophy in which a part of the routing control or intelligence needed to meet the inter-domain QoS challenges must be decoupled from the BGP infrastructure, as well as it can be used irrespective of an explicit or implicit QoS model. Latter means that the IRCs can be used indiscriminately when stubs explicitly requires or negotiates a particular service level with providers, or have no need to request for QoS, because it is assumed that QoS is already embedded in the connection.

1.3 Objectives and Contributions

The primary objective of this thesis is to contribute to the development of lightweight routing schemes for improving inter-domain routing, which entails, studying the Intelligent Route Control (IRC) technique for end-user network or application-layer-based routing and addressing some of the key challenges raised by this strategy. The first issue is to be aware that conceiving an IRC controller correctly is a challenging task, as currently there are no standard design guidelines or recommendations. The second is that IRC middleboxes may oscillate, especially when a large number of IRCs compete for the same network resources. The third is that selfishness is a common characteristic of the IRCs, as well as of the backbone Traffic Engineering (TE) middleboxes, since the paths (or transit providers) are greedily selected by each box, which if there is a misalignment of traffic goals can cause performance degradation of the traffic.

The contributions of this thesis are summarized below.

Analysis of the Inter-domain Quality of Service Routing Problem

The first contribution is the identification and analysis of the main approaches to inter-domain QoS routing. This study also includes an identification of the main challenges surrounding the issue of whether to extend or replace BGP as means of supporting QoS from both technical and non-technical perspectives. After analyzing the potential difficulties and impracticality of inter-domain QoS routing deployment based on BGP, motivation is given for adopting an end-point approach, which is based on the employing of the IRC technique.

Benefits and Feasibility of Intelligent Routing Control

The second contribution is the investigation and demonstration of the feasibility of basing the user-network/application-layer-centric routing on the IRC-based technique, to improve inter-domain QoS. Compared with a BGP-based approach, the IRC-based technique achieves a better performance with respect to traffic latency and the efficiency of traffic transfers. During this study, it was observed that IRC controllers may oscillate. In view of this, we investigate a number of approaches to address this issue. Extensive simulations show that adding randomness to the route control process is the most effective solution. However, when used in very short timescales, it can impair its effectiveness, and this gives rise to the need for alternative strategies, such as social route control.

Design of Intelligent Route Controllers

The third contribution is the provision of a detailed step-by-step design of an IRC controller. The functional architecture that has been devised for the IRCs, draws on a set of principles that were identified for inter-domain QoS routing, and aims at addressing several challenges raised by the IRCs. Path monitoring, and path switching algorithms and routing decision policies are devised and then evaluated through extensive simulations. This includes a formal analysis of the time needed by IRCs to failover, together with an outline of two mechanisms for handling lost probes. These are devised on the basis of the familiar Jacobson-based algorithm for TCP (Transmission Control Protocol) and the Box-plot descriptive statistical tool. The results of the evaluation show to what extent of the proposed mechanisms are applicable.

This contribution includes the provision of an IRC algorithm endowed with social behavior. This is a joint work undertaken together with the Technical University of Catalonia under the auspices of E-NEXT and CONTENT 6º FP IST NoE projects. This work draws on the finding that adding randomness to the route control process may not be enough to avoid spurious path changes, especially in competitive environments. Extensive simulations show that with a sociable IRC strategy, it is possible to obtain similar improvements in terms of the end-to-end performance, while drastically reducing the path switches needed to achieve the desired performance bounds.

Proposal of an ISP-friendly IRC COOPerative Framework

The fourth contribution is the proposal of a novel cooperative framework to address the problem of the interactions between the inter-domain IRC and TE (Traffic Engineering) middleboxes, which is called an Internet Service Provider (ISP)-friendly IRC COOPerative framework (COOP). First, this study provides an empirical analysis of the interactions between these traffic control mechanisms, and shows that IRCs can have a negative impact on backbone traffic engineering. Second, the feedback-based architecture for COOP is designed and evaluated. This is based on a cooperative strategy between direct neighbors or nearest multi-hop neighbors, and is, as far as we know, the first strategy to tackle the problem of the interactions between both middleboxes. The results show that this framework is feasible, since it meets the traffic goals of each party, while ensuring the stability of IRC. Furthermore, the results show that this framework can produce synergistic interactions between both traffic control boxes.

Simulation Models

The proposed IRC strategies and mechanisms are evaluated by using a complete and large scale simulation model that was built on an event-driven simulation tool. Simulations help us debug subtle design flaws. A realistic simulation model was also developed to reproduce the main functionality and behavior of IRC and TE boxes and also to assess the feasibility of using the COOP framework in a competitive environment. We sought to establish a realistic simulation environment based on real traffic demands matrices and on a realistic Internet topology, that was constructed from a characterization of traffic demands and real BGP table dumps.

This contribution includes a prediction criterion for selecting popular destinations, where the traffic is the target of the inter-domain TE optimizations. This criterion is based on an analysis of the errors of common predictors for tracking traffic, and the results show the number of target prefixes can be reduced to a small fraction of the total number, while ensuring stability in the traffic engineering due to an effective way of predicting the volume of the traffic toward these prefixes.

1.4 Structure of this Thesis

This thesis is divided into six chapters and two appendixes, and the structure is outlined below.

Chapter 1 introduces the problem of Quality of Service (QoS) routing and describes the focus, objectives and main contributions of this thesis.

Chapter 2 gives a brief description of BGP (Border Gateway Protocol), including its main drawbacks; and examines the main research efforts of inter-domain QoS routing. This chapter also discusses the controversial issue about whether to extend or replace BGP as a means for providing QoS routing across Autonomous Systems (AS) boundaries. Finally, after investigating all the approaches and challenges, this chapter seeks to encourage the adoption of the Intelligent Route Control (IRC) approach.

Chapter 3 studies the benefits and feasibility of Intelligent Route Control (IRC). This chapter also provides a detailed step-by-step design of an IRC controller in which the path monitoring and path switching mechanisms are studied in some detail. In addition, the strengths of the IRC approach and proposed algorithms are evaluated by means of extensive simulations.

Chapter 4 sets out to move from the conventional standalone and selfish IRC model to a standalone and social route control model. This is carried out by introducing a Social Route Control (SRC) algorithm which is supported by an adaptive cost metric and a two-stage filtering process. The strengths of this approach are evaluated by means of extensive simulations. This chapter is a joint work that was undertaken together with the Technical University of Catalonia.

Chapter 5 analyses the problem of the interactions between IRC and backbone Traffic Engineering (TE), and establishes an ISP(Internet Service Provider)-friendly IRC Cooperative Framework (COOP) to tackle this problem. COOP aims at combining both IRC and TE middleboxes, while synergistic interactions are produced by both mechanisms. The COOP architecture and algorithms are also studied. The strengths of COOP are evaluated by means of extensive simulations with the aid of a realistic simulation model.

Chapter 6 summarizes the main conclusions of this thesis, draws attention to the limitations of this work, and points out avenues for future research.

Appendix A supplements the descriptions about the realistic simulation model employed in Chapter 5, and provides a practical prediction criterion, which can be used to categorize the popularity of the traffic destinations.

Appendix B supplements the results given in Chapter 3. These are preliminary results that compare the performance of the IRC model against BGP, and study of the performance of IRC when governed by different path switching policies.

Chapter 2

Inter-domain QoS Routing: Setting the Grounds for the Way Ahead

A common criticism of the current Internet concerns the poor Quality of Service (QoS) guarantees it provides across the boundaries of Autonomous Systems (AS). The Border Gateway Protocol (BGP) is central to this problem, as it plays a critical role in communications on the Internet, where it facilitates the exchange of reachability information between ASs. However, the actual BGP best-path selection process is agnostic of any performance or QoS metrics. For this reason, there is still an open debate about the requirements for the future inter-domain routing architecture and about whether these requirements are best met by adopting an approach that involves introducing changes into BGP or replacing BGP.

This chapter seeks to throw light on the inter-domain QoS routing problem. First, Section 2.1 provides a brief outline of the BGP protocol. The main drawbacks of current inter-domain routing, regarding the provisions of QoS, are then identified in Section 2.2. Section 2.3 gives a survey of the most important approaches to inter-domain QoS routing. Following this, Section 2.4 offers a broad perspective of the challenges surrounding the issue of whether to extend or replace BGP to support QoS, with an emphasis being laid on the technical challenges. However, we also point out some non-technical issues that have still not been resolved. After all the approaches and challenges have been described, Section 2.5 outlines the target research of this thesis. Finally, Section 2.6 summarizes this chapter.

Bibliographical Note. Part of this chapter has been published in [49].

2.1 Inter-domain Routing: A Brief Overview

This section provides an introduction to the Border Gateway Protocol (BGP), followed by a description of the main traffic control approaches that are based on BGP.

2.1.1 Border Gateway Protocol

BGP, version 4, is the current standard inter-domain routing protocol [32]. Two key functions of BGP are the distribution of Network Layer Reachability Information (NLRI) and control of traffic exchanges among ASs, based on business or Traffic Engineering (TE) policies. However, new capabilities can be easily introduced to BGP due to the high degree of flexibility of its architecture. The support of extra features is captured by the optional capabilities parameter carried within the initial `OPEN` messages used to open sessions between BGP speakers [45]. For instance, BGP speakers supporting new extensions to BGP, such as multi-protocol extensions, should negotiate these capabilities with their peers at the start-up of the BGP sessions [55, 56].

BGP uses a fairly simple path-vector algorithm. The advertisement of reachable destinations includes the Internet Protocol (IP) prefixes and information that describes the properties of the paths to these destinations. This specific information of each path is expressed in terms of path attributes, such as the complete `AS-PATH` sequence (e.g., `AS20:AS21:AS22:AS23`). By default, a BGP speaker selects the route with the shortest `AS-PATH` sequence. However, other path attributes, such as the local preference (`LOCAL-PREF`) and Multi-Exit-Discriminator (`MED`) attributes, can be used to influence the decision process of the BGP routers, since this set out an ordered list of rules or criteria to enable paths to be compared and determine the best path in accordance with the attached attributes.

`UPDATE` messages are exchanged between BGP peers to announce (or withdraw) lists of reachable destinations that share common path attributes. Additionally, to control the size of routing tables maintained in downstream ASs, BGP also supports route aggregation based on Classless Inter-Domain Routing (CIDR), where blocks of IP prefixes can be summarized into a single IP prefix [57].

2.1.2 Path Attributes

BGP-4 specification defines three well-known mandatory attributes, that is attributes that all BGP implementations must be able to recognize and process, and that should appear in `UPDATE` messages: `ORIGIN`, `AS-PATH` and `NEXT-HOP`. Two additional attributes that are part of the BGP decision process, are the `MED` and `LOCAL-PREF` attributes (see Subsection 2.1.4). These are not mandatory attributes, but are often used in traffic control with BGP (see Subsection 2.1.5).

Figure 2.1 illustrates the use of the BGP path attributes. Next, we briefly describe the main BGP path attributes, following the order in which they are used within the BGP decision process (see Subsection 2.1.4).

LOCAL-PREF: This well-known discretionary attribute is a *metric* used internally within an AS to express the degree of preference for each route toward a given

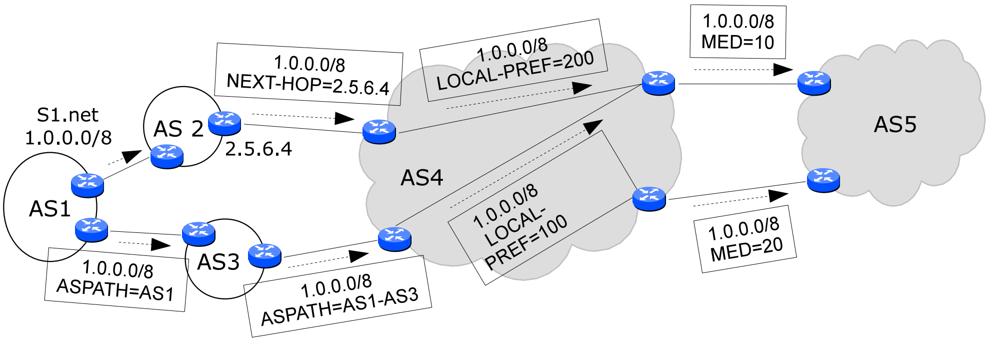


Figure 2.1: Examples of BGP path attributes.

IP prefix (in a range from 0 through $2^{32} - 1$). The default value is 100. Larger LOCAL-PREF values should be attached to the most preferred routes. LOCAL-PREF must not be re-distributed between external BGP (eBGP) speakers, but only between internal BGP (iBGP) speakers. LOCAL-PREF must be recognized by all the BGP implementations;

AS-PATH: This well-known mandatory attribute records the sequence of AS numbers (ASN) that have composed the route so far. It allows BGP speakers to detect loops in the routing. An ASN is important to provide a unique identification of each AS/network in the Internet. ASNs are two-octet numbers, but since 2007, BGP has been able to handle ASNs encoded in four-octet numbers to anticipate the exhaustion of two-octet ASNs [58];

ORIGIN: This well-known mandatory attribute identifies the mechanism that originated the reachability information (0-IGP, 1-EGP, 2-INCOMPLETE). The value assigned by the originating BGP speaker must not be changed by other speakers;

MED: This is an optional non-transitive attribute that might be used when an AS has multiple peering links to the same neighboring AS. The MED attribute has a value that is referred to as a *metric* (in a range from 0 through $2^{32} - 1$). The peering link (i.e., the exit point) with the lowest MED metric is what is the preferred by the neighboring AS. The MED attribute must only be distributed between the iBGP speakers. Thus, eBGP speakers must delete any MED assigned to routes before passing the UPDATE messages to their neighboring AS;

NEXT-HOP: This well-known mandatory attribute contains the IP address of the next-hop router for the IP prefix announced in the UPDATE message.

2.1.3 Messages and Timers

The distribution of the reachability information is achieved over reliable Transmission Control Protocol (TCP) sessions that are set up between BGP speakers using the destination port number 179 [59]. BGP uses only four mandatory messages: `OPEN`, `UPDATE`, `KEEPALIVE`, `NOTIFICATION`, as described below.

OPEN: Initial message sent to open a BGP session. The destination then sends back a `KEEPALIVE` message, which acknowledges the `OPEN` message;

UPDATE: Message used to withdraw or announce a list of reachable IP prefixes that share common path attributes. Every `UPDATE` message might include a variable length sequence of fields to convey routing information, namely withdrawn routes, path attributes, and Network Layer Reachability Information (NLRI);

KEEPALIVE: Periodic message, which is part of the BGP keep-alive mechanism to determine if BGP peers are up. One third of the *Hold Time* interval is usually regarded as a reasonable time between two consecutive `KEEPALIVE` messages;

NOTIFICATION: Message sent to report an error condition detected by the Finite State Machine (FSM), when processing BGP messages (e.g., `OPEN` Message Error or `UPDATE` Message Error) or when the *Hold Timer* expires. The actual BGP session is then closed and the associated RIB – Adj-RIB-In – is flushed. BGP Routing Information Bases (RIBs) are described in the next subsection;

ROUTE-REFRESH: Optional message sent to request the re-advertisement of the Adj-RIB-Out from a peer, for instance, after an incoming policy change. To receive the `ROUTE-REFRESH` message from its peers, a BGP speaker has to negotiate the Route Refresh Capability at the BGP session start-up.

BGP employs timers to control certain periodic activities, such as the *Keepalive Timer* and the *Hold Timer*. The first controls the frequency of `KEEPALIVE` messages and the second indicates the time after which a BGP speaker declares that a peering link is down, which occurs if a `KEEPALIVE` or `UPDATE` message is not received during a given *Hold Timer* interval of time. The default values for *Keepalive Timer* and *Hold Timer* are 30 and 60 seconds, respectively. Nevertheless, manufacturers such as Cisco Systems recommend the use of larger values, such as 60 and 180 seconds, respectively [60].

Another important timer is the Minimum Route Advertisement Interval (MRAI), which its adjustment has a direct impact on the convergence speed of BGP and on ensuring the consistency of the routing information [61]. MRAI refers to the minimum time that must expire before a BGP speaker announces or withdraws routes. The recommended default values for this timer are 30 seconds on external BGP (eBGP) connections, and 5 seconds on internal BGP (iBGP) connections.

2.1.4 Routing Information Bases and Path Selection

Figure 2.2 displays a conceptual model of a BGP speaker. Each BGP speaker maintains three Routing Information Bases (RIBs):

Adj-RIBs-In: the Adjacent RIBs-In store routes learned from peers, before filtering routes or making any modifications to the attributes. These sets of routes are the input of the BGP decision process, described below;

Loc-RIB: the Local RIB stores the best routes selected by the BGP decision process, which are then used to feed the IP forwarding table (i.e., the Forwarding Information Base (FIB));

Adj-RIBs-Out: the Adjacent RIBs-Out store routes to be advertised to peers, as set out in the configured export policy.

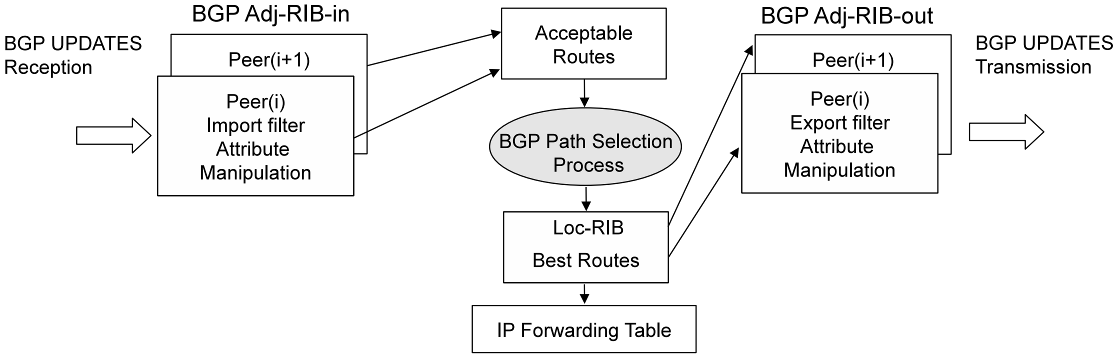


Figure 2.2: Conceptual model of a BGP speaker.

BGP is a single-path vector protocol. It selects and advertises only the best path to each IP prefix. To decide which is the best path, a BGP speaker runs a best-path algorithm on updated routes received from peers. The updated routes are the routes stored in the Adj-RIBs-In after applying incoming filtering and removing all stale routes. To select the best route, a BGP speaker identifies the one that has the highest `LOCAL-PREF`. When there are more than one equally good routes, it proceeds by invoking an extensive sequence of tie-breaking steps that seek to break ties between routes on the basis of their attribute values. The list of rules or criteria adopted by the algorithm are outlined in Figure 2.3, and are applied in the order that is specified. The algorithm stops as soon as only one route can be considered.

It is worth mentioning that some manufactures, including Cisco Systems and Juniper, implement an extra preference highest `WEIGHT` rule that is *local* to the router. This non-transitive absolute attribute overrides the standard `LOCAL-PREF`, and does not conform with the BGP specification, but allows operators, for instance, to assign a preference/weight to a neighbor connection [60, 62]. The `WEIGHT` number must be an

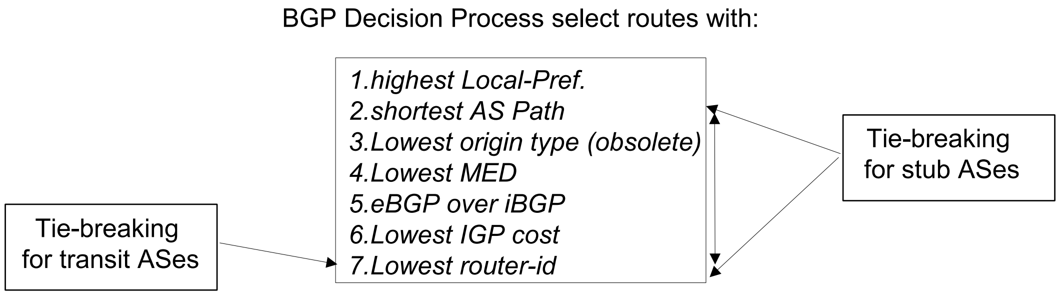

BGP Decision Process select routes with:
1.highest Local-Pref.
2.shortest AS Path
3.Lowest origin type (obsolete)
4.Lowest MED
5.eBGP over iBGP
6.Lowest IGP cost
7.Lowest router-id
Tie-breaking for stub ASes
Tie-breaking for transit ASes

Figure 2.3: Criteria of the BGP decision process.

integer between 0 – 65535. By default, all incoming routes will have a `WEIGHT` of 0 and those routes generated by the local router have a `WEIGHT` of 32768.

2.1.5 Traffic Control with BGP

BGP routing depends on the interaction that exists between the various ASs/routers. For each BGP session, routers employ local policies to select incoming routes or decide whether these routes are advertised to the neighborhood. This distributed policy scheme provides a more flexible system of routing in contrast with other existing routing protocols (that focus on performance), but is a factor of increasing complexity for BGP, especially given the lack of understanding of its effect on routing. However, this routing paradigm has become popular in the ASs of commercial organizations. Much of this popularity can be explained by the fact that BGP enables ASs to control the traffic exchanges that reflect the nature of business relationships and the objectives of inter-domain Traffic Engineering (TE) [63].

This subsection outlines the main techniques used to control traffic exchanges between neighboring ASs/routers.

2.1.5.1 Business Relationships

BGP enables ASs to control the next-hop selection, so as to reflect SLA (Service Level Agreements) or relationships with neighbors. Essentially, there are two major types of relationships between ASs [64–66]:

Provider-Customer: Customers are usually small ASs that pay a transit AS – the provider –, to gain access to the whole Internet (See Figure 2.4). Subsequently, the provider may also contract a transit service from a larger AS. Generally the large providers have a higher degree of market-power than the customers or small providers.

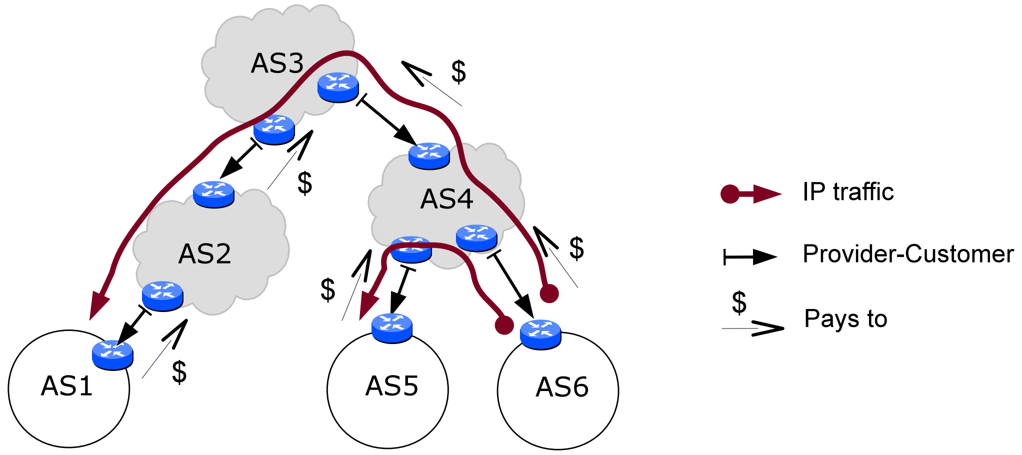


Figure 2.4: Provider-customer relationships.

Peer-to-Peer: Peers are usually large ASs of a comparable size or ASs with significant market-power that agree to implement a cost-sharing policy to exchange the traffic between their direct customers (See Figure 2.5). The establishment of a peer-to-peer relationship may be technically a complex process due to the size of ASs, but it offers some valuable benefits such as traffic exchanges without mutual payments (since the amounts of traffic exchanged are roughly the same), better control over traffic flows, and lower end-to-end traffic latency between peer customers due to the potentially lower number of hops travelled by the packets.

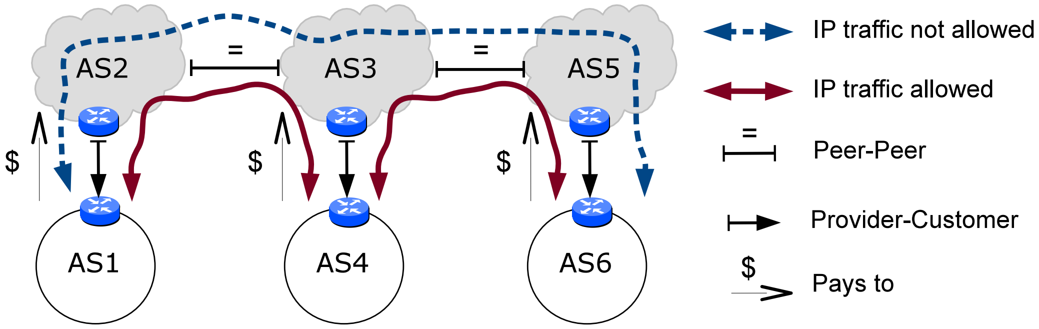


Figure 2.5: Peer-to-peer relationship.

As means of enforcing the business relationships between the ASs, and preventing an AS from being unnecessarily charged, the network operators usually define the following configurations for the export filters of border routers:

- A customer exports its routes to a provider and those it has learned from its customers, but does not export routes it has learned from other providers or peers;
- A provider exports its routes to a customer and any routes that it has learned from other customers, its providers, and peers;

- A peer exports its routes to a peer and those it has learned from its customers, but does not export the routes that it has learned from its providers and other peers.

2.1.5.2 Inter-domain Traffic Engineering

Inter-domain Traffic Engineering (TE) tools are indispensable to engineer the traffic entering and/or leaving ASs, so that a given set of traffic goals (e.g., performance or/and transit costs) are attained. These goals can be achieved by using several techniques as described in the following paragraphs. The interested reader can find more aspects of inter-domain traffic engineering in [63, 67, 68] and the references included.

Egress Traffic Control. The control of outgoing traffic is often carried out by ranking the equal-good routes by means of three techniques to influence the BGP decision process. The first technique consists of modifying the `LOCAL-PREF` of routes and relies, for instance, on active and/or passive measurements or the transit costs of the traffic. For instance, consider a dual-homed stub AS which aims at improving latency. In this case, BGP routers should be configured to insert higher `LOCAL-PREFs` to paths that have been provided by the ISP with lower latency, and lower `LOCAL-PREFs` to paths through the ISP with higher latency.

In view of the fact that the BGP decision process enables the selection of routes based on the Interior Gateway Protocol (IGP) costs of intra-domain paths between pairs of ingress-egress points of a network, the second technique is to reconfigure the IGP costs of the links (e.g., the OSPF (Open Shortest Path First) link weights [30,69]). For instance, an ISP may rely on an integrated TE system to automate the entire process of detecting intra-domain congestion or traffic imbalances over multiple egress points, by selecting suitable IGP costs, and changing the router configurations. A variant of this technique is hot-potato routing which is commonly used by large ISPs. Hot potato routing is accomplished by not passing the eBGP-learned MED into iBGP, so that the routers can select the closest exit points to reduce the transit of packets within the ISP network, and thus the bandwidth consumed by the traffic.

The third, but uncommonly used, technique (since it is only supported by a few BGP routers), is to insert a low `MED` in the routes learned from a more preferred neighbor (e.g., the one that provides lower latency).

Ingress Traffic Control. The control of incoming traffic relies on a careful tuning of BGP advertisements, with the aim of persuading the BGP peers to prefer certain routes to others. There is a plethora of procedures to perform this kind of tuning of BGP, such as `MED` assignments, AS prepending/padding, and selective announcements. One important drawback in the adoption of these procedures, is that they require support from upstream ASs, and thus unfortunately their effectiveness might be limited or even overlooked owing to the traffic goals of the upstream ISP.

MED assignments enables an AS to influence others ASs to decide which link they should use to send traffic. The links/ingress points whose MEDs have lower values are preferred. This technique is generally confined to pairs of multi-connected ASs to avoid route oscillations. This is because MEDs from different ASs cannot be compared on account of their potentially different semantics, and thus the MEDs cannot not form monotonic rankings (i.e., pairs of MEDs do not form a total ordering).

AS number prepending is another technique that can be used by ASs to influence how traffic enters their networks. In this case, during the BGP advertisements, the AS configures the BGP speakers to inflate the AS path length, by inserting multiple copies of its own AS number (e.g., AS1:AS1:AS1:AS1 has a prepending length of 3). A sufficient number of copies could influence other ASs to change traffic to other paths, and thus to deter some traffic from entering through that ingress point.

There are also two additional techniques that are not supported by BGP route attributes. The first technique is to rely on selective advertisements. This includes announcing different routes or IP prefix lengths on different links to balance traffic, for instance. Virtual peering is a more recent idea, which relies on the establishment of unidirectional IP tunnels (e.g., Virtual Private Network (VPN) or Generic Routing Encapsulation (GRE)) between border routers of remote ASs [70]. Through virtual peering, the AS destination of traffic can request the AS source to send traffic via a given tunnel to a preferred ingress point, in order, for instance, to balance the traffic over its ingress points. This technique, however, needs some modifications of the BGP UPDATEs for the establishment of virtual peering, but has the advantage of enabling an AS to select the preferred ingress points for incoming traffic in a deterministic way, which can not be ensured by the pure BGP-based techniques.

2.2 Open Issues in Inter-domain Routing

Inter-domain routing in the Internet is based on BGP (Border Gateway Protocol). Unfortunately, BGP provides end-users with sub-optimal routing in terms of performance and reliability. At the root of the problem of inter-domain routing inefficiency, there are some structural shortcomings that affect BGP. The goal of this section is thus to diagnose the actual role of BGP, and identify the main drawbacks and limitations which can allow it to create or exacerbate the problem of inter-domain routing inefficiency.

2.2.1 QoS Support

The BGP standard only specifies the means that enable BGP speakers to exchange reachability information [32]. At present, there is no standard BGP route attribute to convey QoS or congestion information within the BGP advertisements, and any standard modification to the BGP decision process to enable it to handle these data. Hence, unless attribute manipulation is exercised, the BGP speakers are currently constrained to adapt route selection to the least sequence of ASs that the routes transverse, as announced within the `AS-PATH` route attribute.

The problem with the approach of selecting paths on the basis of the lowest `AS-PATH` length, is that this metric cannot reflect the real end-to-end packet latency. The correlation between path length and Round-Trip Time (RTT) has been shown to be rather poor, which suggests that the traffic performance of the selection of the BGP path may be similar to the one obtained through a purely random choice. In effect, experimental results, obtained using a real Internet topology and based on RTT data, showed that the `AS-PATH` length metric achieved only a 50% success rate in the tests conducted to identify the destinations with a smaller RTT [71].

2.2.2 Route Convergence

Route convergence time, or simply convergence time, is a metric commonly used to measure the *speed* of a routing protocol. It refers to the time that a given routing protocol requires to adapt routing to topology or policy changes (e.g., a fail-down, a new route or a cost change), so that all the nodes in a network have a consistent view of the routing. Otherwise, an incorrect routing table will result in wrong routing decisions, and thus there is a risk of packet loss and degraded quality for applications.

In the case of BGP, the convergence time depends both on topological factors in the Internet and routing policies. In particular, these factors affect the length of the available backup paths for a certain prefix and, thus, the convergence time. As a result, at the Internet scale, the BGP failover process may take several minutes [50,51]. More specifically, the BGP failover may take up to $MINROUTEADVER.max_{p \in P}|p|$ seconds, where $MINROUTEADVER$ (currently 30 seconds) is a timer that states the minimum time that must expire between two consecutive advertisements, P is the set of all the available paths between two remote ASs, and $|p|$ is the length of a path $p \in P$. Unfortunately, this magnitude of route convergence can prevent BGP from support QoS or would put a heavy burden on routers due to the need to achieve shorter response times, while choosing paths that are able to fulfill the QoS traffic requirements. It should be noted, however, that the degree of this burden depends on the relative difference between the timescales of the traffic and the BGP convergence.

2.2.3 Protocol Configuration and Path Control

BGP enables ASs to control how traffic enters or leaves their networks, while reflecting TE goals and business relationships. However, it is notoriously difficult to find the proper configurations of BGP routers in advance (i.e., the right import and export route filters). Moreover, after applying them, the outcome might not be that which was expected in terms of traffic performance, convergence or stability [52, 72].

In addition, even if each AS is able to configure the BGP routers properly, BGP provides little control over the end-to-end path selection, and thus how the traffic generated by each AS is routed to the target destinations. This is because the existing BGP techniques for controlling outbound traffic, such as the `LOCAL-PREF` attribute, only enable control over the first AS hop. As well as this, the ASs usually choose the most preferred neighbor to forward traffic, depending on the economic transit cost of using this neighbor. As a result, it becomes difficult to guarantee end-to-end QoS or to select paths that can circumvent a congested or unwanted ASs.

Furthermore, the existing techniques for inbound traffic control, such as `MED` tweaking, tend, in general, to be very ineffective, as it was mentioned earlier. Basically, this is because these techniques need support from other ASs; and above all, they only enable the ASs to achieve coarse-grained control of incoming traffic [68].

2.2.4 Path Diversity

The path switching technique enables multi-homing ASs to protect traffic from service outages, QoS degradation or even path oscillations, by searching for alternative paths able to bypass the failure or congested ASs. However, the effectiveness of path switching depends on the diversity of the paths available. Path diversity is commonly defined as a *metric* that indicates the number of paths between two remote ASs [73, 74]. Thus, a high degree of path diversity of an AS may indicate a great ability of path switching to failover or to manage congestion. Otherwise, it will be difficult to deal with these problems.

Unfortunately, BGP prevents the path switching from exploiting the full potential redundancy of the Internet. In fact, there are two features of the BGP routing model that are at the root of a modest degree of path diversity. First, BGP is a single-path routing protocol, and therefore a BGP speaker only advertises the best paths to upstream peers. As a result, the RIBs of BGP speakers do not contain all of the available paths for a destination, which can lead to suboptimal path choices being made. Second, the BGP decision process is deterministic. Essentially, the best path for a destination is selected on the basis of its attributes and by applying the criteria outlined in Figure 2.3. Even through operators modify the values of these attributes so that they can influence the path selection, the final outcome can be predictable [52]. This characteristic is particularly important to operators during the traffic control, since it enables them to know in advance how changes in the router configuration affect the path changes. Yet, it can also reduce the path diversity if the tie-breaking rule - *preferring the routes learned from the lowest router ID* - is often used. Unfortunately, observations have shown that between 40% and 50% of route selections are made by following this rule [74]. As a result, many paths with a similar quality are not propagated by the ASs because this criterion always tends to encourage the same next-hop to be preferred.

2.2.5 Stability

One major cause of BGP instability is the presence of route oscillations caused by policy disputes. Policy disputes among ASs can occur when the AS paths are selected by complying with the `LOCAL-PREF` rule. This practice is relatively common because there is a need to reflect business relationships among the ASs or a preference to use transit services from certain ASs. The reason why BGP might fail to converge is that, after any AS selects and advertises its best paths, another AS in the system might switch to a better path. These interactions among policies may lead to persistent route oscillations [75].

A formal BGP model called the Stable Paths Problem (SPP) can provide a sufficient condition for convergence [76]. Unfortunately, the SPP problem has been shown to be NP-complete [77]. An heuristic based on the notion of a "dispute wheel" (i.e., a circular set of conflicting policies) has been set out and it has been shown that "if no dispute wheel can be constructed, then there exists a unique solution for the SPP". In practice, this result implies that a given set of policies in the system have the potential to prevent the routing from oscillating. However, even if this is the case, the routing eventually will be unable to find this solution.

2.2.6 Divergent Interests

BGP has no response to the divergent interests of ASs. This constitutes a major issue. In fact, most of the ISP routing decisions are driven by static local policies that are subject to peering agreements (e.g., cost sharing agreements) or by local traffic objectives. In other words, ASs are concerned about their own efficiency rather than global efficiency. Thus, BGP should enable the ISPs to compute mutual routing, where each AS is suited to a traffic objective in a satisfactory way.

Figure 2.6 illustrates a common example of this problem, where ISPs prefer to select local optimal paths on the basis of a *hot-potato* policy. When using this policy, the purpose of the ISPs is to get rid of the traffic as soon as possible to reduce the transit of packets within their networks by selecting routes with the nearest egress point. The ISPs carry out hot-potato routing by following the *lowest IGP cost* criterion to select the best routes (when there is more than one route to a destination).

Accordingly, `ISP1` will route traffic from `AS1` toward `AS2` selecting the path with egress point 1 (top peering link). In turn, `ISP2` will route traffic from `AS2` toward `AS1` selecting the path with egress point 3 (bottom peering link). As a result, rather than reducing the global length of the paths, the selfishness of the ISPs potentially increase their length instead (to a total IGP distance of 50), and thus the bandwidth resources consumed by the traffic. The best mutually beneficial routing pattern would be to use the routes with egress point 2 (middle peering link) with a total IGP distance of 40. Moreover, in this example it would ensure symmetric routing, which would be welcome for elastic traffic.

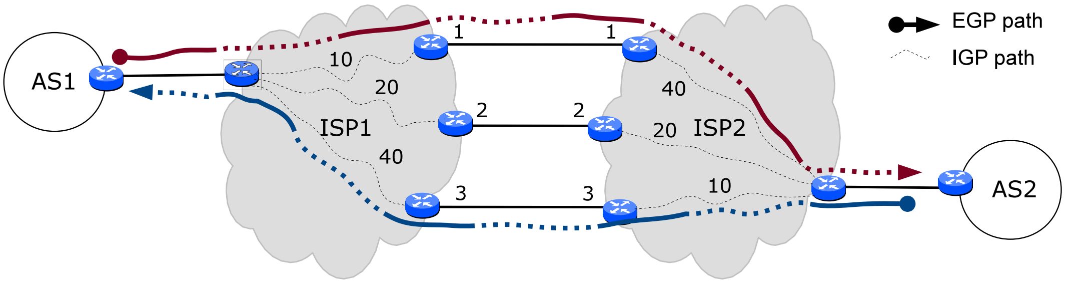


Figure 2.6: Hot-potato routing effect.

2.3 Research Efforts in Inter-domain QoS Routing

This section outlines and discusses relevant studies that aim at addressing the problem of improving QoS support of current inter-domain routing. The work that is analyzed includes existing approaches to the problem, which can be classified into two major classes (and thus two main research topics), – QoS extensions to BGP and traffic control schemes –, as below.

2.3.1 QoS Extensions to BGP

Most of the solutions devised to implement inter-domain QoS [42–44] adopt an approach that involves extending BGP with a new capability [45]. New BGP attributes to convey the QoS information within the `UPDATE` messages and modifications in the BGP decision process have been suggested, so that they can enable the ASs to express higher QoS levels or to be discouraged from routing toward them. By adopting, this approach, two pioneer BGP QoS attributes have been proposed in [46–48]. The first proposal defines an optional and transitive BGP QoS attribute, called QoS_NLRI (Quality of Service_Network Layer Reachability Information). This attribute allows three pieces of QoS information to be carried: the type of QoS information (e.g., packet rate, delay, PHB), the sub-type of QoS information (reserved rate, available rate, loss rate, min/max/average delay) and the value of the QoS information identified in the previous fields (see Table 2.1). The authors argue that the chosen type and format of the QoS_NLRI attribute meets the scalability and smooth transition requirements of inter-domain QoS Routing.

Table 2.1: BGP QoS_NLRI attribute.

QoS Information Code (1 octet)
QoS Information Sub-Code (1 octet)
QoS Information Value (2 octets)
QoS Information Origin (1 octet)
Address Family Identifier (2 octets)
Address Family Subsequent Identifier (1 octet)
Network Address of Next-Hop (4 octets)
Network Layer Reachability Information (Variable)

Figure 2.7 illustrates a scenario of QoS_NLRI-capable BGP speakers. The BGP speaker from AS1 takes 10 ms to reach the *S*1*.net* with IP prefix 1.0.0.0/8. After this, it updates the information in the optional attribute: the delay on the path toward 1.0.0.0/8 is 10 ms; installs this QoS path in its Local RIB and propagates the corresponding `UPDATE` message to its BGP speaker from AS3. In turn, the BGP speaker from AS3 takes 20 ms to reach AS1. Thus, it updates the information in the optional attribute: the delay on the path toward 1.0.0.0/8 is 30 ms; installs this QoS path in its Local RIB and propagates the `UPDATE` message to its BGP speaker from AS4, and so on until the delay information arrives at AS5. It should be stressed that the best QoS path selected by the BGP QoS route decision process of BGP speakers is the path with the smallest delay. The path via `AS5-AS4-AS3-AS1` has a delay of 80 ms, while the best QoS path would be via `AS5-AS4-AS2-AS1` and has a delay of 60 ms.

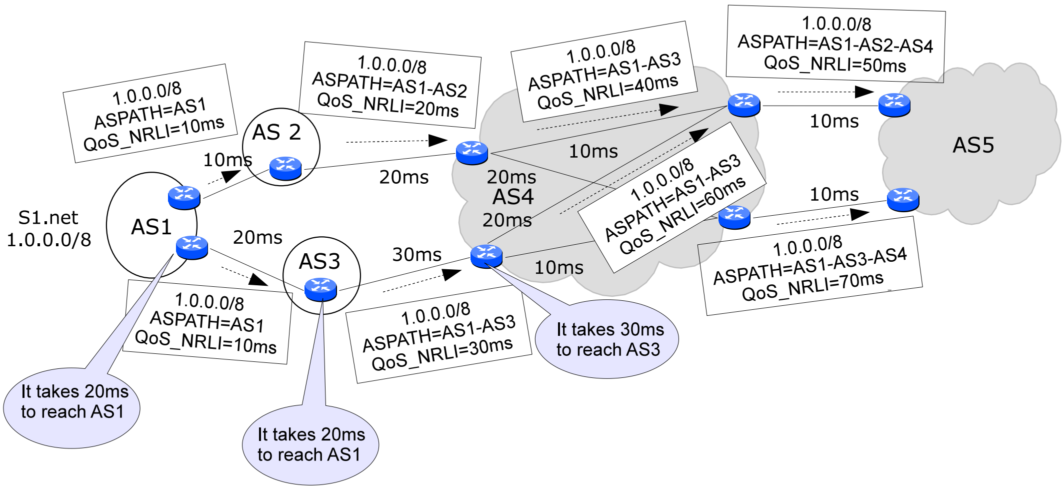


Figure 2.7: QoS_NLRI capable BGP speakers with path delay information.

The second proposal defines a variable length, optional and non-transitive BGP QoS attribute, that allows a domain to decide which type of QoS information a BGP border router redistributes. The non-transitivity property implies that unless the attribute is supported by a BGP speaker, it must not be forwarded to its peers. This attribute associates with the announced prefix, the Diffserv Per-Hop Behavior (PHB) (i.e., Best Effort (BE), Assured Forwarding (AF) or Expedited Forwarding (EF)), supported by the BGP QoS-aware routers, and the type and the value of QoS parameters (e.g. maximum bandwidth associated with a PHB, or available bandwidth, or even maximum or minimum delay), or the required QoS signaling (e.g., an indication that the border router supports resource ReSerVation Protocol (RSVP)) (see Table 2.2).

Table 2.2: Flexible BGP QoS attribute.

PHB identification (2 octets)
QoS Type (1 octet)
QoS value (4 octets)

Both of the proposals involving QoS extensions to BGP have drawbacks. First, the use of in-band signaling results in low convergence and instability problems. Second, neither of these proposals offers a means of representing dynamic changes in the network state. Another important issue that is not addressed is the potential problems raised by the non-uniform semantics of QoS information, i.e., the ASs might use different types of QoS information, with different meanings, and degrees of precision. One interesting possible of overcoming this problem is to use a common external representation to keep a uniform semantic for distributed QoS information, such as the IST Mescal Meta-Class concept, to characterize a domain transfer capability [44].

A new inter-domain QoS metric, called index Available Bandwidth Index (ABI) has been designed to address two main challenges that arise from extending BGP – scalability and the heterogeneity of inter-domain links [78, 79]. As a result, the ABI index is conceived as a semi-dynamic metric, and can be defined as the probability that the available bandwidth belongs to a given interval. For path ABI propagation between BGP peers, it is adopted a similar approach to the one proposed in [48], previously described. At the decision process level the decision criterion employed is the maximum weight, W, of a QoS route. The idea of using weights representing ABI indexes is to make it easier to draw a comparison between the ABIs of the available path options. Finally, two thresholds are designed as stability mechanisms: a) the Route Update Threshold – a new route is installed into the local routing table only if its W is significantly better than the W of the actual route; and b) the Link State Threshold – this is to control `UPDATE` generation to ensure that only important variations in the available bandwidth of a link are propagated.

The ABI scheme tackles the problem of the fine-grained notification of dynamic changes in the network state. It also makes the precision of QoS information uniform. However, the proposal is based on the controversial assumption that reasonable Central Processing Unit (CPU) resources can be allocated for the execution of mathematical operations during the computation of ABI indexes of links and paths. In the Internet scale, this feature has a potential to affect the scalability of the solution.

Finally, most of these proposals assumes that, before, the QoS information can be processed, both the BGP decision process and the `UPDATE` message handling process should be slightly modified, while keeping the original state machine of the BGP software intact, as outlined in [32]. The kind of modifications to be added to the classical BGP decision process depend on the type of QoS information, but basically include the addition of some extra steps as shown in the general Algorithm outlined in 2.1 [44, 80]. These extra steps involve seeking to identify the paths that serve the same destination in the same class of service or quality of service level, and to select a path that optimizes the QoS performance characteristics (e.g., delay or bandwidth). Path comparison is a key operation of the algorithm, because it enables a QoS-aware BGP router to achieve this goal, by means of comparing the QoS metrics of paths, and returning the best quality path.

Methods such as lexicographic ordering (where in selecting best path, the QoS metrics are ordered in terms of their degree of preference, and compared from the highest to the lowest preferred metric) or simultaneous comparison (where in selecting the best path, all the QoS metrics are compared, and is returned the path in which all the QoS metrics are the best), can support the path comparison logic, when the quality of paths is derived from multiple QoS metrics [80]. However, these methods are not recommended because either they bias the path selections toward the most important criterion at the expense of less important criteria, or they output a undefined result in the case when none of the paths have the best of all the QoS metrics.

The recommended path comparison methods found in the literature are based on weighted methods, through which the multiple QoS metrics are properly weighted, and thus fairly compared. Two of the most promising instances of this, because of their simplicity, have been suggested within the scopes of the IST Mescal and EuQoS projects, as follows respectively [43, 80]. The first instance is called the weighted ordering method; in this, the multiple m QoS metrics $a_{ik}, i = 1, .., m$ of a path $k \in P$ are normalized to create dimensionless values, and linearly combined, as shown in Equation (2.1) for additive QoS metrics. The optional weights $\alpha_i \geq 0, i = 0, ..., m$ can be chosen to allow one metric to be preferred to the others. The lowest-cost path is selected as the best path. The second instance relies on a concept of Degree of Preference (DoP) as shown in Equation (2.2), where f_i is a QoS preference factor chosen by the network provider to control routing, and T_i the maximum/target value for QoS metric i such that the path k non-conforming with a QoS objective is excluded from the decision process. The path with the highest DoP among all the QoS conforming paths is selected as the best path.

$$M_k = \sum_{i=1}^{m} \alpha_i \frac{max\,[a_{i1}, ..., a_{ip}]}{a_{ik}},\ with \sum_{i=1}^{m} \alpha_i = 1,\ p = |P| \tag{2.1}$$

$$DoP_k = \sum_{i=1}^{m} \frac{f_i}{max\,[0, T_i - a_{ik}]} \tag{2.2}$$

It is worth to notice, that when transitive QoS attributes are adopted, non-QoS-aware BGP routers should set the partial bit of the attribute to 1 (to mark that the QoS information is incomplete) and passes the network state information unchanged to peers [47, 48]. In this case, the path comparison logic of a QoS-aware BGP router should prefer paths with complete QoS information, and compare the paths only when both have complete (both with partial bit=0) or both have incomplete (both with partial bit=1) QoS information.

Algoritmo 2.1 Basic steps of the QoS path selection algorithm [80].

```
Identify routes that serve the same destination;
Consider routes that have the same QoS class identifier;
Compare the QoS performance characteristics associated with resulting routes with
respect to a given comparison logic;
Return the route that optimizes the QoS performance characteristic;
If more than one route has been returned, apply the classical BGP route selection
process.
```

2.3.2 Traffic Control Schemes

A set of detour solutions are also emerging in the literature to addressed the inter-domain routing problems and these are based on traffic control schemes. There are two main approaches to the issue of traffic control at the inter-domain level: a) Internet-wide approaches that resort to overlay network-based mechanisms; and b) end-point approaches using multi-homing and smart routing-based mechanisms (a.k.a. intelligent route controllers). This section outlines the most relevant solutions that have been underpinned by both paradigms.

2.3.2.1 Internet-wide approach: Overlay network-based mechanisms

Overlay network-based solutions have been developed to overcome the disadvantages of plain BGP extensions. In this approach, a number of overlay nodes, called Overlay Entities (OE), is strategically placed across several ASs to build an additional layer of indirection/virtualization on top of existing network infrastructure. The role of OEs is to provide users/ASs with a fine-grained control of path selection, so that whenever an Internet path fails or does not perform as expected, an OE locally determines an alternate path for outgoing traffic or remotely controls a proper path change, within a timescale of several seconds rather than several minutes.

There are essentially two groups of overlay network-based solutions depending on the level of interaction with the underlying routing layer. In the first group of solutions, collectively described as pure-overlays, an extra routing layer is used that is completely independent from the underlying routing. The major advantage of this approach is that it is easier to implement, since it enables QoS support and resilience capability, without requiring the employment of QoS mechanisms (e.g., scheduling or buffer management) along underlying IP paths or having to modify existing protocols.

Figure 2.8 illustrates a pure-overlay network composed of five OEs. A virtual link is a unidirectional link that connects two OEs, and corresponds to an underlying IP path between them. Each OE sends probes to others OEs via virtual links. These probes are used to capture virtual link performance characteristics (in general these are capacity, loss and delay) and availability. This information is then shared with the other OEs. By means of this knowledge, when an OE receives data packets toward another OE, it can dynamically route packets throughout best paths or select an alternate path in the case of path or quality outage. Thus, in the illustration, OE1 sends probes to OE2 and OE4; OE2 sends probes to OE3 and OE5; OE3 sends probes to OE5; and OE4 sends probes to OE2 and OE5. Thus, after sharing this knowledge, OE1 can route data toward OE5 via OE2 or OE4 depending on the performance of the available paths (in this case `OE1-OE2-OE5`, `OE1-OE2-OE3-OE5`, `OE1-OE4-OE5`, `OE1-OE4-OE2-OE3-OE5` and `OE1-OE4-OE2-OE5`), instead of sending the traffic directly to the Internet via the single path available at AS1 (in this case `AS1-AS2-AS3-AS6-AS9`).

Two examples of pure-overlay networks are shown in [65, 81]. The first scheme uses a mechanism called Controlled-Loss Virtual Link (CLVL) to provide per-flow bandwidth differentiation, rate assurance and congestion control. The second scheme offers a complete set of mechanisms for QoS Routing, including hierarchical aggregation for overlay networks. Overlay Brokers (OB), are strategically placed across domains, to form an Overlay Service Network (OSN), which provides QoS applications with a unified access to routing and resource allocation.

The second group of solutions decouples a part of the policy control portion of the routing process from the BGP devices. In this way, the Overlay Policy Control Architecture (OPCA) has been proposed to enhance the BGP's fail-over and deal with the problem of its limitations with regard to the control of incoming traffic [82]. It adopts an overlay combined with BGP. Like OSN, overlay entities, called Policy Agents (PA), that communicate via a new protocol Overlay Policy Protocol (OPP), process incoming policies or the route changes that are constrained by local AS policies (e.g. pricing constraints or SLA). The key requirement of OPCA is to have a knowledge of AS

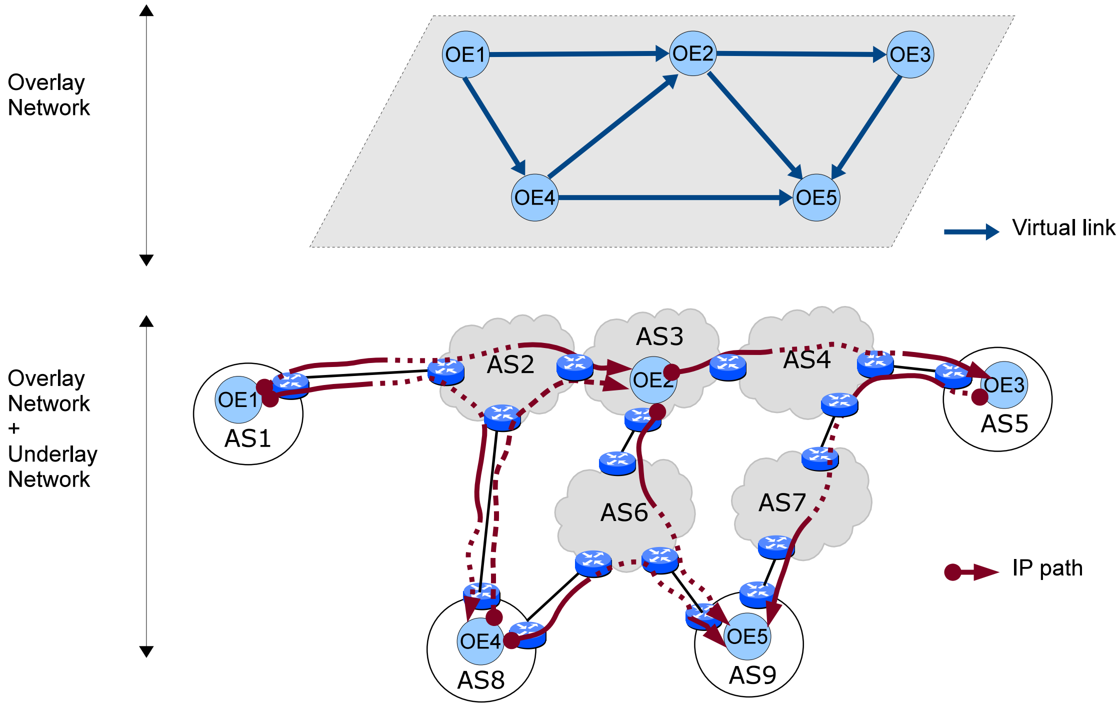


Figure 2.8: Illustration of an overlay network.

relationships. To achieve this goal, OPCA includes an AS Topology and Relationship MAPper (RMAP) to deduce AS relationships from BGP routing table dumps. When a PA detects a link failure or congestion, it queries the RMAP to find an alternate path that is able to reach it and discover the best remote PA that can be contacted to bypass the failure or congestion. In addition to the PA and RMAP components, a mechanism called PA directory is suggested in OPCA; this is a means by which a local PA can obtain the IP address of a remote PA to be contacted.

Figure 2.9 illustrates the OPCA architecture in action after a PA detects a link failure. In the event of this, the following sequence of events would take place. AS7 stops receiving traffic from AS1 via the ingress link AS6-AS7. Once PA5 detects the failure, it queries RMAP, which suggests that the best alternate path for traffic is via AS5 (that is AS1-AS2-AS3-AS5-AS7) and the best point of routing control is PA3. PA5 sends a message to the PA directory requesting the address of PA3. PA5 sends a message to PA3 via path AS7-AS5-AS3 requesting it to block any advertisements that contain the AS6-AS7 segment in the path sequence. PA3 reconfigures the BGP routers at AS3 in line with this request. AS3 installs the new path, AS3-AS5-AS7, in loc-RIBs of the BGP routers, and advertises this path to AS2. AS2 installs in loc-RIBs of BGP routers the new path, AS2-AS3-AS5-AS7, and advertises this path to AS1. Finally, AS1 installs in the loc-RIBs of its BGP routers the new path AS1-AS2-AS3-AS5-AS7 for traffic, which fixes the problem.

The main drawbacks of OPCA scheme are low scalability and inaccuracy. In fact,

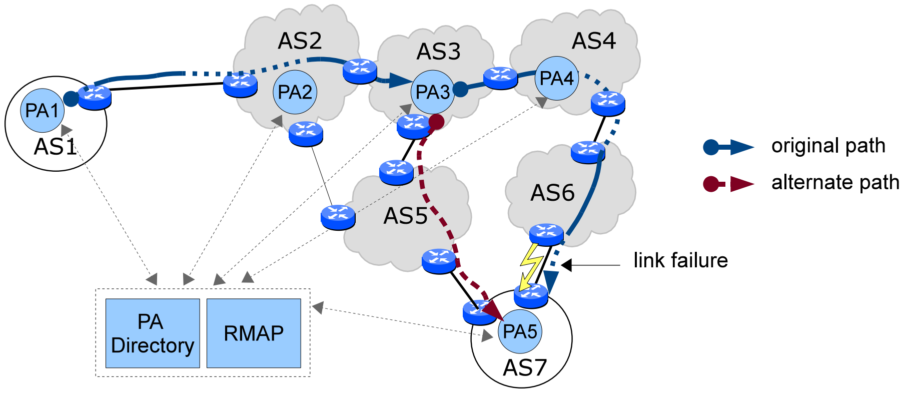


Figure 2.9: Illustration of OPCA architecture in action.

all the PA actions are dependent on the RMAP component and on its ability to deduce the inter-AS topology.

The main difference between OPCA and the pure-overlays, is that the latter circumvent BGP to route packets, which can lead to violations of the business policies between ASs and undesired interactions with the underlying infrastructure [83]. In addition, although well-known studies have shown the validity of pure-overlays to offer better performance than BGP (e.g., throughput, RTTs, loss rates and path availability), it might not be enough to ensure stringent QoS levels since there is a lack of control of the behavior of the underlying infrastructure [84].

There are two main open issues to investigate regarding the implementation of the previous Overlay network-based mechanisms. First, the degree of cooperation between ASs required by OPCA-like solutions is still unclear. This means that further research is required to answer the following questions: *Do we need high levels of cooperation among ASs to achieve good levels of resilience and performance? How can the cooperation be achieved, while ensuring scalability*? Second, given the fact that pure-overlays might violate the business policies between ASs, *if we were able to construct overlays constrained to underlying routing policies, can we (with much less routing flexibility) still achieve the same levels of resilience and performance*?

2.3.2.2 End-point approach: Multi-homing and Intelligent Route Control

Multi-homing consists of increasing Internet connectivity by contracting multiple broadband lines (e.g., Business DSL (Digital Subscriber Line), E1, E2, or E3) from two or three different ISPs. Studies have shown that multi-homed stub ASs experience a potential performance benefit compared to single homed ASs of at least 40%, as well as offering significant improvements in reliability [85]. However, the use of the multihoming technique by itself is not enough to obtain these improvements because inter-domain routing of IP packets still relies on BGP.

Routing mechanisms, referred to as Intelligent Route Control (IRC) systems are,

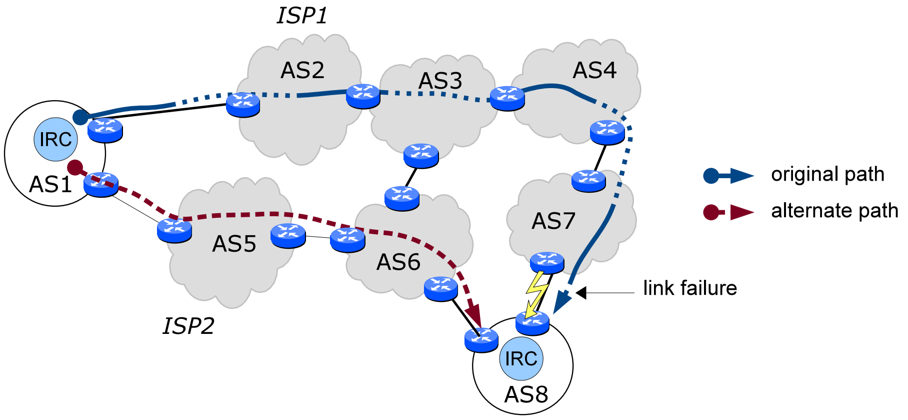


Figure 2.10: A simple scenario of two multi-homed stub ASs employing intelligent route control.

thus, being increasingly used by multi-homed stub ASs, as they provide a holistic way to address local end-to-end traffic challenges (e.g., latency, or loss rate bounds) through shifting some traffic between ISPs in short timescales. One key function of IRCs is, thus, to capture the performance of paths. To address this issue, an IRC usually employs an active probing method, that is based on sending streams of small probe packets toward target destination through each path, and monitoring them [86].

Figure 2.10 illustrates a typical scenario of one AS employing IRC, AS1, where its IRC might improve the performance of the outbound traffic toward the remote stub AS, AS8, through switching among the paths `AS2-AS3-AS4-AS7-AS8` and `AS5-AS6-AS8` across ISP1 and ISP2, respectively. In this case, IRC of AS8 can either operate passively or more actively, by cooperating with IRC of AS1 by capturing path measures or by changing the way traffic enters its AS.

In contrast to the pure-overlays, the IRCs never circumvent BGP to meet the traffic requirements. In this way, the additional complexity needed to cope with BGP inefficiency is set apart from BGP. In other words, although IRCs interact with BGP, they do not require any changes in the BGP routers or support from the ISPs or cooperation with ASs along the paths. An IRC runs as a local middlebox, that requires access to RIBs of routers to collect available paths and to issue command scripts to routers in a shorter timescale than the BGP timescale (to indicate the ranks of paths, generally, by tuning their associated `LOCAL-PREF` attributes). Furthermore, in contrast to OPCA, it does not require any further support from ASs along the paths or from any centralized component.

To conclude this account of the IRCs, it is worth noticing that IRCs are not a new concept, since many companies have been devoting efforts to research and developing IRC products [87,88]. However, little is known about the technical details of commercial IRCs. In its turn, the research community has produced some publications devoted to the design and stability issues of IRCs [54,89].

As well as the research topics described, there are two additional important issues that need addressing. First, *it is unclear if the levels of route control, performance and reliability offered by the intelligent route control are enough.* If not, would it be beneficial to deal with this issue by combining IRC with overlays in a hybrid mechanism and thus have the best of both worlds: the simplicity of intelligent route control and the routing flexibility of overlays?

The second issue arises from the fact that intelligent route control, as well as overlays, is typically selfish in nature. That is, intelligent route control greedily select paths, and is only concerned with its local traffic goals. Unfortunately, this behavior does not necessarily lead to the best routing in the Internet [90]. Regarding this issue, inter-AS cooperative routing seems a promising approach to address the problem of being selfish; however, again, it could demand unreasonable levels of cooperation between the ASs [91, 92].

2.4 Extend or Replace BGP?

The purpose of this section is to examine from a broad perspective, the challenges surrounding the issue of whether to extend or replace BGP to deploy inter-domain QoS routing. Our study includes both the technical and economic aspects of the problem. We begin by discussing the technical or internal challenges to BGP. Following this, we will discuss some of the economic or external challenges to BGP.

2.4.1 Challenges to Inter-domain QoS Routing Deployment

Two key models can be used to underpin the analysis of whether the BGP infrastructure should be used or can accommodate new features and functionality [93]. In the first model, BGP is used as part of a General Purpose Transport (GPT) infrastructure and, in the second model, BGP is used as part of a Special Purpose Transport (SPT) infrastructure.

On the one hand, the main aim of GPT model is to use the BGP data distribution mechanism as a generic application transport mechanism. The main concern of GPT is to observe whether the data distribution application requirements can be satisfied by the BGP data distribution mechanism. On the other hand, SPT assumes that the BGP data distribution mechanism has been designed specifically to carry routing information. One of the main concerns is, thus, to ensure that the additional complexity added to BGP is kept within bounds so that it does not cause instability in BGP.

In addition to the GPT and SPT models, Risk, Interference and Application Fit (RIFT) are three important concepts that might be used to describe the different trade-offs involved when extending BGP [93]. *Risk* is focused on achieving robustness trade-offs, by modeling the impact of the addition of a new application or feature in an existing implementation. *Interference* focuses on how a new application affects the behavior of existing applications. In other words, interference relates to the question of coupling or interdependence among the applications. Finally, *application fit* refers to how the requirements of the data that have to be conveyed match, the BGP data distribution mechanism. As a result, given the concerns that arise from the SPT and GPT models, this implies that SPT is sensitive to risk and interference, and GPT is focused on application fit.

By means of the previous terminology and related concepts, we discuss some of significant challenges that arise from the deployment of inter-domain QoS routing. However, in this discussion, we naturally focus only on the SPT approach, and make an assumption that the QoS information requirements can match the BGP data distribution mechanism. More specifically, when adding QoS to BGP (once the protocol implementation has been modified), we believe that there is an intrinsic risk that the behavior of the BGP will be degraded and destabilized. In particular, we believe this would be evident when multiple classes of service are added. This behavior is similar to the case when BGP carries multiple application data types, which may cause interference among the multiple applications, and also destabilize the BGP routing system.

Figure 2.11 summarizes the main trade-offs between the addition of QoS and the stability, scalability and accuracy aspects of inter-domain routing, which allow the risk and interference profiles to be analyzed. In this chart, the trade-off between the addition of QoS and the algorithmic complexity is also shown, because the latter may influence the scalability of BGP as a whole. After this, the trade-offs between QoS and scalability and between QoS and stability will be discussed.

Regarding the QoS vs scalability trade-off, it is important to understand the effects of adding QoS to the memory requirements and the CPU load at the BGP speakers. The main factors that have an impact on memory requirements at the legacy BGP speakers, include the number of IP prefixes or networks (N), the mean AS distance in terms of hop count (M), the total number of unique AS paths (A), the mean number of BGP peers per BGP speaker (P), and the lengths of the binary words (in bytes) required to store a network (W_{net}) and to store an AS number (W_{asn}). When all these factors are combined, the memory requirement (m_{req}) at legacy BGP speakers is given by Equation (2.3) [94].

$$m_{req}(BGP) = P.\left(N.W_{net} + M.A.W_{asn}\right) \tag{2.3}$$

When adding a QoS attribute to BGP (e.g., QoS_NRLI attribute) an estimate of memory requirement at qos-enabled BGP (q-BGP) speakers (ignoring implementation

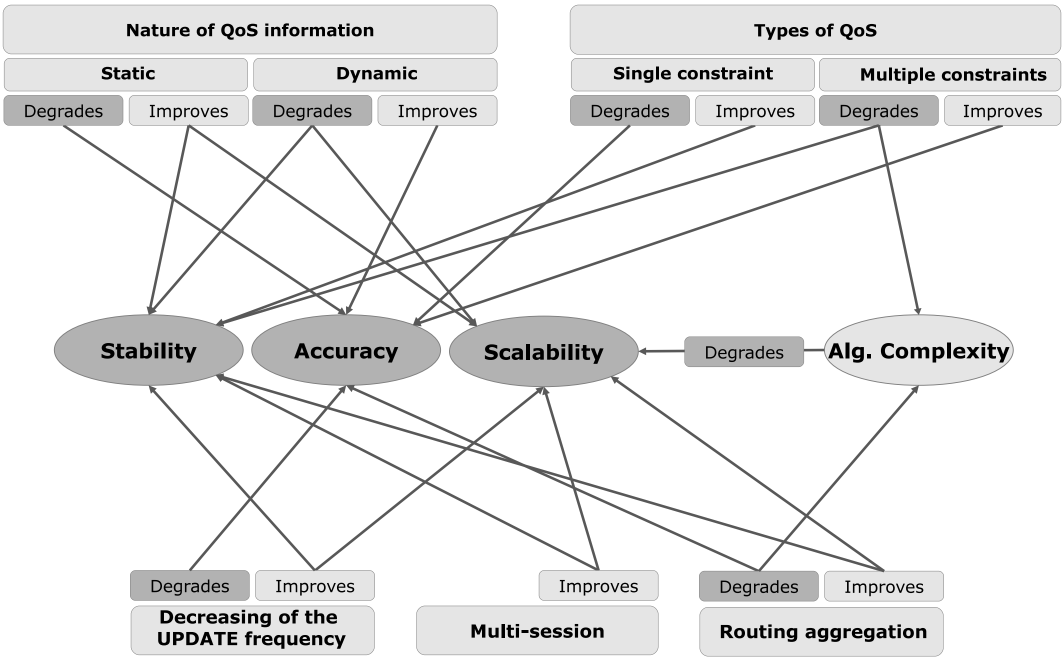


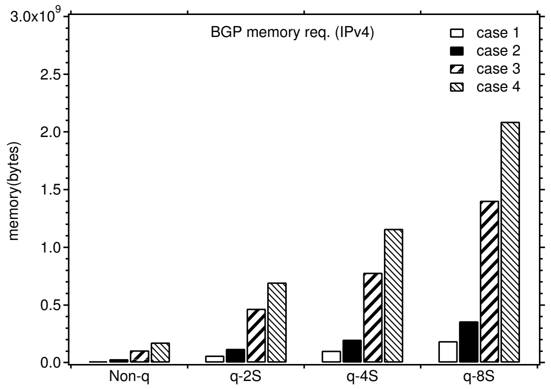
Figure 2.11: qos-enabled BGP design trade-offs.

details) can be given by an extension of Equation (2.3) as shown in Equation (2.4), where W_{qos} represents the minimum length of the binary word to store the QoS information carried on the QoS attribute, and C represents the number of supported services, including the best effort service.

$$m_{req}(qBGP) = C.P.\left(N.(W_{net} + W_{qos}) + M.A.W_{asn}\right) \tag{2.4}$$

In the interests of providing a specific example, Figure 2.12 illustrates the growth of q-BGP memory requirements for the cases shown in [94], considering $W_{qos} = 4$ bytes and up to eight services. As can be observed, it is clear that even for just two or four additional QoS service extensions, a BGP speaker will undoubtedly need many more memory requirements to store the additional amounts of routing state information.

The second significant scalability concern, as mentioned earlier, is the BGP CPU load. On the one hand, QoS extensions will require more paths to be advertised per prefix destination, and thus more processing overhead. On the other hand, there is also some correlation between the BGP dynamics and the utilization of CPU. That is, the number of BGP `UPDATE` message announcements received in a given period of time might increase the router CPU load, as every update demands some processing for route in-filtering, route selection, RIB updates, FIB updates and route out-filtering. Measurements have confirmed this correlation especially when BGP routing tables are unstable [95]. These have also shown that high router CPU loads can increase the convergence times and Internet outages. Moreover, since the network state information



	Networks (N)	Mean AS Distance (M)	Unique Paths (A)	BGP peers (S)
Case #1	100000	20	3000	20
Case #2	100000	20	15000	20
Case #3	120000	10	15000	100
Case #4	140000	15	20000	100

Figure 2.12: qos-enabled BGP memory requirements for IPv4 addressing.

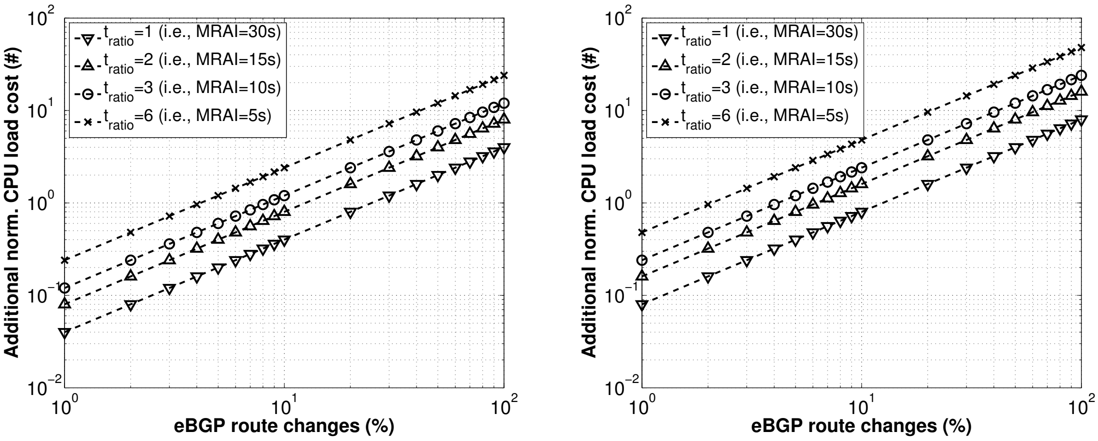
needs to be distributed more often, QoS extensions have the potential to consume idle CPU cycles, and thus can exacerbate this problem.

To illustrate the previous point, we provide in Equation (2.5) a rough estimate of the additional CPU load at QoS enabled BGP routers, denoted as AC_{load}, as a function of the fraction of the number of routes that have just changed f_{rc}, the number QoS services supported C, and the ratio between the timescale of standard BGP data distribution and the corresponding timescale of q-BGP, $t_{ratio} = \frac{MRAI_{BGP}}{MRAI_{qBGP}}$. To deduce the additional CPU load experienced by a q-BGP router, we considered as reference the worse-case scenario experienced by a legacy BGP router. That is, when it needs to process a BGP `UPDATE` burst due to routing changes for all the N destinations in all peers P [96], and it does not compress multiple addresses into one message, since no destination shares a common set of path attributes.

$$AC_{load} = \frac{C^{qBGP}_{load}(N, P, C, f_{rc}, MRAI_{qBGP})}{C^{BGP}_{load}(N, P, MRAI_{BGP})} = \frac{\frac{1}{MRAI_{qBGP}}.N.P.f_{rc}.C}{\frac{1}{MRAI_{BGP}}.N.P} = f_{rc}.C.t_{ratio} \tag{2.5}$$

On the basis of the previous Equation (2.5), Figure 2.13 depicts AC_{load} as function of f_{rc} for several t_{ratio} when q-BGP supports four or eight services. As can be observed, the potential additional CPU load rises fast as long as the number of services and the frequency of BGP `UPDATE` messages are increased. For instance, when only about 6-7% of the routes have changed (which is commonly more than that in ISP transit networks), q-BGP routers supporting four and eight services would need, respectively, about 50% and 100% of the total number of CPU cycles needed to handle the worse-case scenario of regular BGP for a MRAI timer, equal to half of the original value.

It should be noted that we did not attempted to model the BGP router load in



(a) q-BGP supporting four services

(b) q-BGP supporting eight services

Figure 2.13: Rough estimation of the additional CPU load.

detail because it depends on several factors that are really hard to quantify, such as the real reaction of the CPU load to different kinds and volumes of BGP messages and the influence of the prefix compression capability of BGP; as well as the significant influence of the network operating system. For instance, Cisco IOS is usually based on a monolithic architecture where all the processes share the same memory requirements, whereas Juniper JUNOS is based on a modular architecture where a process runs as a module in its own protected memory [97].

To conclude the scalability concern, it should be underlined again that the complexity of the algorithmic component is determinant for inter-domain routing [43]. The processing overhead (and thus the additional CPU load) as well as the frequency of computation dictated by the number of `UPDATE` messages received, also depends on the difficulty of solving the path selection problem. The algorithm is potentially more complex than the original one because it involves path comparisons subject to multiple constraints. In any case, the time required by the algorithm must be polynomial in order to limit the number of CPU cycles consumed during each path selection.

With regard to the QoS vs stability trade-off, the first issue to understand is how to manage the problem of adding a class-based routing feature to the BGP implementation without causing instability. This issue could be solved by adding two mechanisms. First, the BGP data distribution mechanism should be able to advertise multiple classes of routing information. A part of this issue can solved with the introduction of the Multi-protocol Extensions, which enable BGP to transport information for multiple address families and sub-families, distinguished by distinct AFI(Address Family Identifier)/SAFI (Subsequent Address Family Identifier). However, the actual point of multiplexing is located at the BGP layer, which constitutes a serious problem of BGP robustness. That is, one single corrupt message for a given AFI/SAFI might terminate the BGP session and compromise other AFI/SAFI. One solution to this problem is to move the point of multiplexing of this data into the transport layer and, thus, to allow multiple sessions between two BGP peers [98]. However, it still lacks a mechanism that can enable the BGP to advertise multiple classes of information for the same address.

The second issue to understand is the dynamics of the QoS information. On the one hand, when using static QoS information, such as the distribution of an IDentifier (ID) of the supported classes of service or link capacities toward a prefix, it can improve the stability of routing at the expense of the risk of adding a degree of inaccuracy to the routing state. On the other hand, when using dynamic QoS information, such as current raw available bandwidth or latency to a prefix, it can improve the accuracy of the routing information. However, it can introduce an extra component of instability to the inter-domain routing system. In addition, the choice to use dynamic QoS information demands modifications in the BGP decision process to support the selection of best routes that take into account the propagated QoS information.

2.4.2 External Challenges to BGP

Often protocol designers and engineers only focus on the technical aspects of engineering problems, and fail to take into account their economic context. The modest success of QoS architectures (e.g., Intserv and Diffserv) can be partially justified by this decoupling between the technical and economic aspects of the various problems that need solving. This is an important lesson to learn.

From our perspective, as well as a clear demonstration of the potential QoS benefits, the economic features should also be taken into account in any scheme which aims at adding QoS to BGP. In a recognition of this need, two recent proposals include economic frameworks to allow ISPs to provide good QoS at a predictable cost for the end-users by making public the prices charged by ISPs [99, 100].

From an operational standpoint, an important issue that should be addressed is the objections of network administrators to needless complexity and their reluctance to change. Moreover, the provision of additional bandwidth to links is an attractive alternative to QoS; it is simple, it works and it is becoming cheaper. In short, a concrete proposal will not compelling to the majority of ASs, and thus, it will not be widely deployed, if it cannot provide a satisfactory answer to the following three key questions:

1. *How does the network of my AS benefit if a q-BGP solution is adopted?*
2. *Does the extra complexity introduced by q-BGP make the configuration of routers or the debugging of network problems more difficult?*
3. *How should a q-BGP be deployed in a scalable and incremental fashion?*

Understanding the granularity of the routing problem, together with common operational networking practices and tools in an AS, are also important factors to consider.

For instance, even though the AS is the base unit of inter-domain routing, previous proposals for adding QoS to BGP are still based on quite unrealistic models of ASs. In particular, each AS has been modeled as just a single node with only 2-3 peers. But, in reality, current measurements clearly show that there are a significant number of ASs that are composed of hundreds of BGP speakers, and have several hundreds of peers (e.g., the three top ten ASs defined by the number of peers, the ASs 3356, 174 and 7018 have 2433, 2223 and 2222 peers, respectively [101]). Another drawback of previous schemes is the lack of any interface with other auxiliary (or fundamental) layers or mechanisms on Inter-AS QoS provisioning, such as a SLA management layer or TE.

These later issues add another important question that must be clearly answered: *What should be the role of the BGP in providing inter-AS QoS*? Most probably, in the light of these two additional driving forces, QoS-managed ASs would prefer to model BGP as a general purpose transport infrastructure that could assist ASs in SLA trading or control inter-AS QoS interconnections, instead of regarding BGP as a routing protocol. In other words, this perspective is appealing for the development of BGP-based mechanisms to control inter-AS QoS interconnections. The combination of these mechanisms with, for instance, Diffserv Bandwidth Brokers (BB) can be used to exchange and negotiate the conditions about QoS connectivity services (e.g., bandwidth and latency bounds, routing, pricing and penalties) with peers [102]. The role of the BB-like entities should be to allocate and control shared resources (e.g., bandwidth), as well as to make decisions about QoS interconnection policies.

2.5 Target Research and Concerns

In the previous sections, we examined possible approaches to tackle the problem of QoS support of current inter-domain routing namely, QoS extensions to BGP and traffic control schemes, which included a discussion about main the challenges to inter-domain QoS routing deployment. After becoming aware of the potential difficulties and impracticality of an inter-domain QoS routing deployment solely based on q-BGP (qos-enabled BGP), we decided to focus on the end-point approach based on Intelligent Route Control (IRC), which is a cost-effective incremental inter-domain QoS routing framework among a reduced set of strategically selected and non-directly connected multi-homed stub ASs. Following this, we outline a list of the main motivating factors for using IRC:

(i) *IRC approach is simpler to implement and most of the extra complexity to improve end-to-end QoS is pushed to the edge of the Internet.* Indeed, it enables QoS support and resilience capability, without requiring the employment of explicit QoS mechanisms (e.g., scheduling or buffer management) along underlying IP paths or modifying existing protocols. IRC are deployed only at source ASs of traffic and optionally at sink ASs. A commodity computer system that is running, for instance, Linux, would be enough to support and implement the IRCs capabilities for the current Internet.

(ii) *IRC approach never circumvents BGP.* On the contrary, pure-overlays are capable of improving end-to-end QoS by circumventing BGP to route packets, which can lead to violations of the business policies between ASs, and undesired interactions with the underlying infrastructure [83]. In addition, although well-known studies have shown the validity of pure-overlays to offer better performance and reliability (e.g., throughput, RTTs, loss rates and path availability), it might not be enough to ensure the QoS levels required because of the absence of any control feature of the underlying infrastructure [84]. Finally, an overlay network is not ubiquitous. Only nodes on an overlay network can control traffic by tunneling traffic through other overlay nodes.

(iii) *IRC approach needs little (or sometimes no) cooperation from the intermediate ASs.* On the contrary, the degree of cooperation between ASs required by OPCA-like solutions is unclear.

(iv) *IRC approach can handle the rearrangement of inter-domain traffic in short timescales, while providing end-users with some degree of path control.* The q-BGP approach is inadequate to handle the rearrangement of traffic in short timescales, as described earlier. IRC is an out-band solution, i.e., a solution that is not intrinsically supported and signaled by using BGP. Moreover, an IRC box installed in an AS source of traffic – a content provider, for instance – can control the selection of paths for outgoing traffic in a effective manner by adapting BGP routing, depending on the path performance and the end-user network/application-layer QoS constraints for traffic. In a complete model, a peer-to-peer network of IRCs can be formed, where pairs of IRCs cooperate and exchange some network state information in attempt to control how the traffic enters the remote ASs.

2.6 Summary

In this chapter we have surveyed research studies designed to address the problem of inter-domain QoS (Quality of Service) routing, and also shown the main shortcomings of each proposal. However, the discussions about the requirements for the future inter-domain routing architecture and about whether these requirements are best met by an approach that involves introducing changes into BGP (Border Gateway Protocol) or replacing BGP, remain unresolved. In particular, we have emphasized the fact that while some challenging issues arise from the deployment of q-BGP (qos-enabled BGP), others are derived from external issues that confront BGP. In summary, our aim in this chapter was basically to:

- Support the need of tackle the question of inter-domain QoS routing;
- Clearly reveal the most important open issues in the area of inter-domain QoS routing;
- Outline an up-to-date set of related working proposals designed to address some of the challenges raised by inter-domain QoS Routing;
- Clearly reveal the potential difficulties and impracticality of an inter-domain QoS routing deployment solely based on q-BGP;
- Argue that, unless the role of BGP can be revised to include also the culture of operational networking, inter-domain routing will continue to suffer from a chronic failure, that is, a lack of QoS support;
- Provide the target research of this thesis, that is to study the Intelligent Route Control (IRC) for inter-domain control.

Chapter 3

Benefits and Feasibility of Intelligent Route Control

This chapter investigates the intelligent route control technique to improve inter-domain routing. This includes a proposal for a full system design, and a set of evaluations that are carried out to show the potential benefits and feasibility of this approach and to validate the functionality required to fully support intelligent route changes.

Bibliographical Notes. Parts of this chapter have been published in [103, 104]. At the time of the submission of the Thesis, there is also one working-paper under submission for publication.

3.1 Introduction

Research studies have been conducted to address the problem of inter-domain Quality of Service (QoS), and provide a full description of the functionality required to fully support inter-AS (Autonomous Systems) QoS connections [42, 43]. Two core pieces of the proposed functional architectures are dynamic inter-domain Traffic Engineering (TE) and off-line inter-domain TE.

On the one hand, dynamic inter-domain TE is responsible for inter-domain routing, which would be implemented by a qos-enabled version of Border Gateway Protocol (q-BGP) [48]. q-BGP would enable an AS to propagate `UPDATE` messages to its peers containing the information about the QoS connectivity services that it supports (e.g., bandwidth and latency bounds, pricing and penalties) and to enforce business and traffic engineering policies (e.g., to perform load-balancing between multiple, equally-good, QoS peering connections);

On the other hand, off-line inter-domain TE is responsible for the optimization of inter-domain routing. That is, it would enable an AS/Internet Service Provider (ISP) to select the best neighbor ASs to carry the traffic, while meeting certain network optimization objectives (e.g., minimization of the consumption of the bandwidth contracted to a peer and improvement of load-balancing). Overall, in QoS environments, off-line TE would enable an AS to balance the utilization of existing QoS connectivity services among customers, as well as identify the need to establish new peering QoS relationships with neighbor ASs to improve the offering of QoS [105, 106].

Both the above-mentioned TE functions offer the prospect of improving the QoS across the AS boundaries with the advantage of reducing, or simply dispensing with, the need for QoS negotiations at the application layer, and thus avoiding the use of a signaling protocol, such as the resource ReSerVation Protocol (RSVP) [107]. However, these functions are not enough to respond to some of the key challenges for delivering inter-domain QoS routing, and to invoke a few of these challenges, in particular the following: a) fast reaction to link failures and quality degradation, and fast recovery from failure; b) being centered on the QoS level that is required by the end-user network or application-layer.

Addressing the challenge of a fast reaction and fast recovery involves taking account of the response times of BGP to failures or policy changes. Unfortunately, neither BGP, nor its q-BGP version, is not prepared to react to failures/congestion in short timescales, without significantly decreasing the end-to-end QoS; this is due to its slow route convergence problem, and the high computational complexity involved in the processing of route attributes and in the path selection [50–52].

With regard to the challenge about of being centered on the QoS level that is required by the end-user network or application-layer, unfortunately the off-line TE at ISPs is only focused on local and coarse-grained traffic optimizations, which are also executed at large timescales due to the high processing overhead required for the routing optimization process. This implies that the ISP network may be optimized for local network performance objectives, but provide suboptimal QoS levels (higher/lower) with respect to the real needs of the user networks or applications. Moreover, it should be noted that these frameworks might not justify their complexity and deployment costs, since they require significant changes in the present inter-domain routing infrastructure and have the potential to run the risk of increasing the difficulty of managing the routers configuration [72].

To overcome all the above-mentioned problems, *there is a need to explore the possibility of developing* lightweight, cost-effective, *routing schemes aimed at improving the performance of inter-domain routing, or supporting the delivery of end-user network or application-layer QoS across ASs boundaries.* The primary goal of this chapter is to address this challenging issue.

In this chapter, we investigate a technique that has been increasingly adopted by multihomed ASs, in particular by Content Service Providers (CSPs). To be more precise, middleboxes, called Intelligent Route Controllers (IRCs), have been employed to supplement the functionality of the BGP-based infrastructure, to ensure that the best ISP to deliver content to remote peers/users is dynamically selected, while observing the end-to-end performance (e.g., latency, and loss rate bounds) or correspond quality metrics (e.g., Mean Opinion Score (MOS)) [108–110]. In this work, the generic term *QoS* can refer to the set of performance parameters that can be measured in each available path at the point, at which an end-user network/applications access the service, or to a metric that results from the combination of these network performance measures. Thus, performance, and QoS or quality, may be used as interchangeable terms depending on the target actor.

The IRC technique is well-motivated as it provides a holistic means of solving local traffic challenges by shifting a certain amount of traffic between ISPs in relatively short timescales (i.e., from a few seconds to a few minutes) [89] and follows a philosophy that, as some authors argue, allow it to deal with foundational problems in inter-domain routing; this is a part of the routing control (or extra functionality) required to meet the QoS challenges must be decoupled from the BGP-based infrastructure [111, 112]. In other words, when adopting IRCs, the end-to-end performance of traffic can be improved without any need to change the BGP routers and without the cooperation of the ISPs along the data paths.

Despite the strengths of intelligent route control, developing and configuring an IRC system correctly is a challenging task and currently there are no standard design guidelines or recommendations. In particular, it involves dealing with a number of monitoring and algorithmic aspects; these include active probing and passive measurement methods, and routing decision policies and algorithms. Unfortunately, most of these technical details are not clearly documented in the configuration manuals of existing IRC products or are not documented at all [87, 88], which makes it difficult to capture the full IRC functionality, while hampering related research, and hence, the effective deployment of IRC solutions.

In a nutshell, the goal of this chapter is to provide detailed descriptions of the design of IRC systems and report the results of the development of the mechanisms which form the basis of IRCs, including the evaluations that were carried out to determine the benefits and feasibility of the approach, and validate the functionality required to fully support intelligent route changes

The contributions of this chapter are therefore:

(i) An analysis of the key design principles for an IRC system, and a detailed step-by-step design of an IRC system;

(ii) Provide a demonstration that end-user network/application-layer-centered routing based on an IRC strategy is a viable approach to improving inter-domain quality of service. Simulation results show that IRC can achieve a higher overall effectiveness than BGP. The results also show that the IRC-based technique is compatible with the standard Differentiated services (Diffserv) architecture, which corroborates the idea that there is no need for an explicit application-layer signaling protocol;

(iii) A study of the IRC stability. IRC systems may introduce persistent route oscillations, which cause significant performance losses. We investigate three approaches to address this issue: randomized path monitoring, randomized path switching and history-aware path switching. The results of simulations show that adding randomness to the route control process is the most effective solution, and is in line with a related study [89]. However, when used in very short timescales, it can reduce its effectiveness, and for this reason, there is a need to employ a different strategy (see Chapter 4). The results also indicate that randomized path monitoring is an effective alternative to randomized path switching for conservative sampling frequencies, and the use of a sophisticated IRC algorithm, such as history-aware path switching, may not bring about significant benefits in terms of stability;

(iv) A study of the mechanisms that can be used by IRCs to react to path failures and losses of probing packets, including a formal analysis of the time needed by IRCs to detect and recover from a path failure, together with an outline of two mechanisms for handling lost probes. These are designed as revised versions of the familiar Jacobson-based algorithm and the Box-plot descriptive statistical tool, which are used in TCP (Transport Control Protocol) and traffic anomaly detection fields respectively. The simulation results show that when IRCs are blended with these mechanisms, they can adapt their timeout timers to network conditions, and above all, protect them against spurious timeouts, while keeping short reaction times.

The remainder of this chapter is divided as follows: Section 3.2 outlines in detail the key design goals for a QoS-enabled Internet. Section 3.3 briefly highlights a set of extra problems created by the IRC paradigm (e.g., stability and reliability issues) and sketches out possible solutions to these problems, which are explored in greater detail in the next sections. Section 3.4 provides a full description of the functionality of IRCs. Sections 3.5 and 3.6 describe, in more detail, the path monitoring and path switching algorithms.

In the later sections the results are shown of extensive evaluations of the IRC strategy at three different levels. First, Section 3.7 evaluates the potential performance benefits of employing the IRC strategy, as opposed to the use of BGP. Second, Section 3.8 provides a simulation study of the proposed mechanisms that are used for the adaptation of the IRC monitoring timers, and thus, the IRC response to path failures. Third, Section 3.9 examines a simulation study focused on the stability issue in which three classes of techniques are compared, to stabilize the IRC mechanisms.

3.2 Design Principles

This section describes key design principles for enabling inter-domain Quality of Service (QoS) routing. We discuss the ways in which the current inter-domain routing architecture fails to meet these design principles. Here, we pave the way for Intelligent Route Control (IRC) to achieve the goals behind these principles.

3.2.1 Decoupled Routing Control from BGP

The inter-domain routing of IP (Internet Protocol) packets toward a given prefix involves four basic functions: Route Discovery, Route Filtering, Path Selection *and* Packet Forwarding [32].

Route Discovery. At least one route connecting the source of traffic and the remote AS should be found to ensure end-to-end connectivity between both. To achieve this goal, each AS uses external BGP (eBGP) to propagate the best learned routes to neighbor ASs, while their respective neighbors continue the process until convergence of eBGP. Large ASs may also use internal BGP (iBGP) to distribute the routes inside their networks.

Routing Filtering. This function involves filtering the routes that are accepted by an AS and those that can be advertised to neighbor ASs. Each AS can configure the import route filters of BGP routers to filter unwanted routes and/or to influence the selection of the remaining routes by means of route attribute manipulation. It can also configure the export route filters of BGP routers to manipulate the route attributes or control whether a given route should be propagated to the neighbor ASs.

Path Selection. The BGP decision process ranks the set of routes that pass through the import filters in order of preference of their attributes, and selects the best ones for installation in the local RIB (Routing Information Base). The best routes that pass through the export filters are then propagated to neighbor ASs.

Packet Forwarding. Once the best routes has been inserted in the local RIB, the BGP daemon of a router feeds the IP forwarding table, so that the IP packets can be forwarded along these routes.

Unfortunately, Routing Filtering and Path Selection are becoming increasingly complex functions because of the growth of the Internet and the addition of new features to BGP [56, 113]. Any changes to the current implementation (e.g., addition of new attributes and changes to the decision process to enable a corresponding path selection) to face the BGP inefficiency and a lack of QoS support, or in other words to support performance policies, should be avoided as they may introduce an extra routing burden and complicate the configuration of border routers.

Principle: *Path control limitations of inter-domain routing should be addressed in a separate and distributed route control layer.*

In the IRC model, an arbitrary number of standalone IRCs located at source ASs (or pairs of IRCs located at remote ASs) form a distributed route control layer. This model determines that each IRC is responsible for managing BGP dynamically. In particular, each IRC selects the best routes on behalf of the BGP routers to control the allocation of the traffic exchanged between remote ASs, depending on the network conditions and traffic goals.

The IRC model considers that the underlying BGP structure needs virtually no modification, and remains unaware of the distributed routing control architecture formed by the IRCs. The BGP decision process is still being used, but the extra routing intelligence will be inside the IRCs. This comprises the computational logic needed for path selection and the tweaking of BGP route attributes to meet the quality objectives. All the messages exchanged between IRCs are, also, handled by this out–band BGP layer, which gives more flexibility to control the routing of IP packets and coordination between remote ASs.

Another major advantage of the IRC model is that the extra complexity needed to improve the quality of inter-domain routing is pushed to the edge of the network. No intermediate ASs are needed to participate in the control architecture, and hence no IRCs or any kind of modifications or cooperation are needed in transit ASs connecting the remote ASs.

3.2.2 Fast Path Failure Reaction and Recovery

Inter-domain routing based on BGP is not prepared to react quickly to link failures. At the Internet scale, the BGP fail-over process may take several minutes due to various delays in the propagation of `UPDATE` messages and in the path exploration [50, 51]. To address this issue, several improvements have been suggested to reduce the number of messages distributed between border routers after a link failure, and to distribute and process the ones that are carrying good news more quickly (e.g., a new path to a given destination) [114–116]. However, even with these improvements, the convergence timescale is still in the order of a few minutes, which may not suit the requirements of mission-critical or real-time traffic. An inherent feature of this drawback is that these solutions still rely on the in-band message distribution mechanism, and are subject to the hop-by-hop routing paradigm, where each AS is responsible for selecting and advertising alternate paths to adjacent ASs until the convergence of the protocol.

Principle: *ASs should be able to detect link/QoS failures and recovery in a sufficiently short timescale, without significantly degrading the end-to-end quality.*

To achieve this goal, an IRC must react to link or QoS failures much faster than the BGP layer. In the light of this, an IRC system relies on end-to-end monitoring that probes every single link that connects the AS to the Internet. Once an IRC detects quality violations against user-network/application specified thresholds for a given traffic flow, it adapts the local routing on-the-fly. Specifically, an alternative path that is able to bypass the network link failure or congestion is selected approximately in the path monitoring timescale. In short, IRC strategy enhances the end-to-end performance of the underlying BGP layer in very short timescales because no BGP messages will ever be spawned. However, tweaking inter-domain traffic in short timescales by means of BGP may be only feasible for outbound traffic control. This is because tuning BGP in short timescales for inbound traffic control may lead to BGP damping [117].

3.2.3 User or Application-centered QoS

Operators can employ inter-domain Traffic Engineering (TE) to meet the traffic challenges of local AS, such as load-balancing or minimization of the maximum utilization of the peering links or the contracted bandwidth. Once the optimal routing has been computed by the TE algorithm, this is accomplished by tweaking the BGP attributes of an arbitrary number of routes. When the traffic demands are stable, the routing changes carry-out by the TE box potentially results in network operation points close to the optimum, over a period of days, weeks or even months, depending on the granularity of the traffic. It should be noted, however, that this is not always entirely true on account of the fact that the Internet traffic is bursty [118]. In short, apparently, TE ensures that network resources are efficiently used and improve the performance of traffic, since this is routed through egress links or paths with enough capacity.

The major drawback of inter-domain TE is that traffic objectives are not centered on the user's perspective of network quality. On finding a new optimal routing, there is, thus, no guarantee that the end-to-end quality properties for a particular flow will be compliant with its QoS requirements (e.g., one-way delays or round-trip time bounds). Even though user-centered TE was used to improve its traffic performance, it requires much more processing and route tweaks than the traditional scheme, due to the higher dynamics of the end-to-end performance metrics.

Moreover, TE problems are usually formulated by means of convex optimization, such as linear programming, which may imply changing a major part of the routing, whenever the routing is optimized [119–121]. In other words, after optimization the TE approach requires jumping to the optimal routing immediately by changing a huge number of routes with a consequent burden on the BGP infrastructure.

Principle: *To improve the user's perceived QoS, ASs should be centered on end-to-end QoS objectives. Moreover, the path changes to optimize/improve the routing should be incremental.*

To achieve these goals, an IRC should follow the quality of the paths and adapt the routing of an individual (or handful of) flows to the real-time network conditions, independently of the other flows. In particular, an IRC should employ path monitoring and selection processes that are centered on the end-to-end quality metrics of each flow, running concurrently with other instances of other flows.

In contrast with TE, IRC performed routing is much simpler and its impact on the BGP infrastructure is diluted along each window of control, since the optimal routing is achieved gradually. An IRC system should deliberately avoid jumping the actual routing to an optimal routing, immediately after a response to path performance changes. Hence, IRC performed routing is usually suboptimal, although each solution that follows should converge to the optimal. The success of this procedure depends, however, on the IRC timescale and the sequence of path switches.

Finally, it should be pointed out that these IRC achievements are only feasible if the host AS is a multi-homed AS. The effectiveness of the IRC is also dependent on the first-hop path redundancy provided by multi-homing. In other words, the effectiveness of the IRC performed routing could be enhanced as long as the AS-level path diversity increases in terms of topology and end-to-end performance measures. On the other hand, IRCs are not suitable to transit ASs, in particular to large transit ASs (such as tier-1 or tier-2 networks). In effect, adapting the routing of the traffic aggregates at large networks to network performance conditions, might require changing a huge number of paths, and the effects of managing large amounts of inter-domain traffic in short timescales are unpredictable.

3.2.4 Lightweight, Incremental Deployability, and BGP Compatibility

Figure 3.1 shows the growth in the number of advertised ASNs (AS Number) since 1996. There are more than 35000 ASNs, and there is a significant number of ASs owning a few thousands peerings, such as AS7018 or AS3356 [113]. These data serve as a means of gauging the total number of BGP speakers and the high connectivity density of today's Internet, and hence the massive deployment of BGP.

Thus, as discussed earlier, the addition of new features to BGP for supporting differentiated traffic exchanges will introduce an undesirable degree of complexity to inter-domain routing. Before these challenges can be faced, only lightweight and incremental solutions that are able to keep the Internet routing infrastructure almost intact, will have the chance to be extensively deployed by ASs and router manufacturers, es-

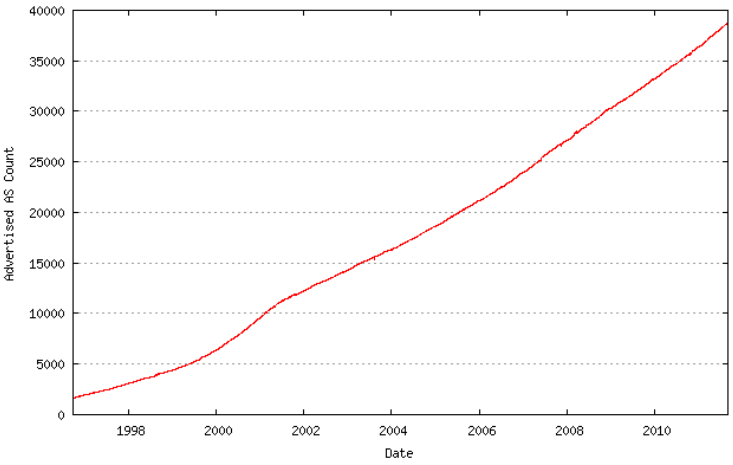


Figure 3.1: The growth of ASNs since 1996 [1].

pecially for business and economic reasons.

Principle: *A solution that aims at providing inter-domain QoS routing, should be lightweight, incremental and BGP compatible so that it can be widely deployed.*

To address this goal, IRC smartly exploits the traffic engineering capabilities of BGP to get better performance. Compelling work has shown that it is not necessary to circumvent BGP routing to improve end-to-end performance [122]. IRC only needs to interface with BGP to import a portion of the paths from routers RIBs to construct a comprehensive view of routing, and at the end of each routing cycle, export the selected paths into the RIBs. This can be accomplished by establishing regular BGP sessions with border routers, and issuing command scripts to collect the paths and tweak their attributes to make the decision process of BGP to select those that perform best. This kind of tweaking allows ASs to perform high-level path selections, and avoid incorrect routing decisions resulting from partial views of the local AS routing state. Moreover, it should be underlined that in this way, the IRCs never circumvent either the BGP decision process or the local business policies.

3.2.5 Compatibility with QoS Architectures and Inter-AS QoS Connection Mechanisms

In the next-generation Internet, QoS-managed ASs will need effective mechanisms to control the Inter-AS QoS connections, possibly combined with mechanisms similar to the DiffServ (Differentiated Services) Bandwidth Brokers (BB). The former mechanisms will provide ASs with the necessary means to negotiate with their peers, the conditions about the QoS connectivity services to be rendered. In turn, the later mechanisms will have the responsibility of allocating and controlling shared resources (e.g., bandwidth), and making decisions about QoS interconnection policies.

With the aid of similar frameworks, IST MESCAL and EuQoS research projects have outlined a set of inter-AS QoS connection mechanisms based on a q-BGP version [42, 44]. Although these frameworks potentially help to improve the end-to-end quality of service, they are only able to achieve coarse-grained QoS objectives. More specifically, they are suitable to achieve the traffic objectives of ASs (i.e., mid or long-term traffic objectives), but they are not suitable to meet the finer end-to-end QoS requirements of end-user networks or applications.

Principle: *A solution that aims at providing inter-domain QoS routing, should be compatible with QoS architectures and Inter-AS QoS interconnection mechanisms.*

To achieve this goal, as well as providing regular best-effort service, each IRC should be able to probe a remote AS for other services or class of services, through all the available egress links of its AS. As the q-BGP protocol, IRCs should also support the concept of multi-session, including the replication of all IRC mechanisms (monitoring, measurement and routing decision algorithms) and their adaptation to the specific requirements of each service or class of service. Thus, if q-BGP is enabled, a session is initialized for each service or class of service.

3.3 Challenges Raised by Intelligent Route Control

Intelligent Route Control (IRC) has the potential to cause new problems, such as rationality, objective overloading, and tweaking issues; and in addition, concerns about stability, reliability and network overhead. Before it can be viable, an IRC system should be designed to address these problems. In this section, we briefly highlight these issues and sketch out possible ways of overcoming the problems. Most of these problems are addressed in the next sections, and a few, later on in Chapter 4.

Rationality of Route Decisions. IRCs must be *rational (or intelligent)* mechanisms during the path selection process, which means that any path switch should only be carried out if it really improves the performance of the traffic. In other words, IRCs acting on behalf of stub ASs should be able to make clever ISP choices, or select equivalent paths, at any time, so that the end-users or applications can increase their degree of satisfaction regarding a given traffic flow or aggregate, which has certain end-to-end performance requirements.

Otherwise, if an IRC is not rational during the selection of paths, it will not be able to make any kind of judgment about its routing preferences. This study uses a cost metric or a utility function assigned to each path or IRC flow, whose values enable IRCs to rank paths. A configurable threshold can be also blended that only allows the traffic to be switched to another path if it can ensure a minimum improvement of performance. This mechanism may also play an important role in avoiding IRC oscillations.

Separation between IRC Objectives and Routing Policies. IRC path selections should be translated into route attribute tweaking (e.g., `LOCAL-PREF` or `MED`) to influence the path choices achieved by the BGP decision process. Similarly, certain routing policies that observe business objectives (e.g., *a multi-homed ASx might prefer to use the ASy rather than the ASz as the ASy is cheaper*) can be translated into particular route attribute tweaking. As a result, the route attributes values computed by the IRCs and computed (or defined) by the operators to enforce local business policies, might interact and lead to an unpredictable routing behavior.

To prevent the overlapping of the IRC and business objectives, there should be a clear separation between the IRC traffic objectives and local routing policy objectives. Concretely, it should be administratively defined at least two non-overlapped ranges of values for each path attribute, assuming that the range for business policies overrides the range defined for IRC policies.

Flexible Tweaking of the IRC Objectives and Algorithms. ASs, depending on their size, and business or network applications, might have different traffic challenges and routing behavior requirements. To fulfill its function, an IRC should provide a plethora of different types of performance metrics, weights and IRC algorithms and policies to the network operators. With this feature, traffic flows can experience differentiated levels of performance, depending on the traffic objectives and IRC algorithms selected by the operator. As well as performance or equivalent quality-only optimization, IRCs can also support cost-only or mixed optimization modes.

Protection Against Route Oscillations. Routing instability of IRC flows should be avoided or controllable. Routing instability, defined as the rapid change of network reachability, has three main effects: an increase of packet losses and latency, and an increase of overhead within routers (e.g., buffers, buses and CPU).

As a routing protocol, an IRC should be composed of effective mechanisms that are able to stabilize routing under highly changing network conditions, by controlling or preventing an excessive number of path switches. To give a specific example, an instance of this kind of mechanisms is the BGP route-flat dampening that imposes a penalty on the routes that change [117]. If this penalty exceeds a given limit, the route is dampened.

The design of mechanisms to dampen the destabilizing effect of flapping routes, should address three main issues in the IRC field. First, it should provide robustness against instabilities caused by self-interference (aka self-load). Self-interference takes place when an IRC switches traffic to a path and as a result its performance drops significantly due to the extra amount of traffic. We mean, for instance, that measured packet latency or load might increase significantly. After this, it adapts the routing by returning the traffic to the original path and so this oscillation might continue persistently. Second, an IRC should prevent instability caused by the coexistence of masses of IRCs that compete for the same network resources and simultaneously perturb the network and interact with each other. Finally, an IRC should address the problem of the synchronization between the probes, that results from the overlapping of their measurement intervals, which is also a potential factor of path oscillation.

Protection Against Spurious Timeouts. IRCs employ active monitoring methods to capture the performance of paths. However, little is known about how exactly the monitoring mechanisms can be used by IRCs, and how they can handle the losses of probing packets.

The choice of the appropriate method for handling the losses is critical, as it can have an impact on the time needed by IRCs for detection and recovery from path failures. RTO (Request Time-Out) timer-based mechanisms can be adopted to declare that a probe is lost after a given interval of time. However, it is crucial to find a proper RTO value that corresponds to trade-off between a very short reaction time (which may lead to premature path changes), and a very long reaction time (which may lead to late path changes).

Spurious timeouts is thus an additional factor of IRC oscillations and performance weaknesses, that should be addressed. In short, the need for adaptive mechanisms to protect IRCs against spurious RTO timeouts should be taken into account carefully during its design.

Reduction of the Network Overhead and IRC Burden. Compelling recent studies like [123] have shown the problem that tracking and controlling most of the traffic of multi-homed stub ASs turn out to be impracticable. This is because the large variability of the topological characteristics of inter-domain traffic, in addition to the limited aggregation of this traffic, suggests that the number of paths that have to be tracked and controlled, is not only highly variable, but also very large.

Despite the variability and lack of aggregation issues, the IRC system, the underlying BGP infrastructure and the network in general should not be overloaded. A realistic approach is to bound the overhead of routing changes by focusing only on the paths to a small fraction of the destinations (e.g., popular destinations or stable traffic volumes), since it is possible to track and optimize a significant fraction of traffic by simply controlling a reduced set of stable paths [67].

Detailed Reporting. IRCs should provide frequent detailed reports about traffic statistics and path switches. For instance, an IRC can publish, on an web page, the results of all the measured performance metrics, including bandwidth utilization, end-to-end latency, losses and jitter, correspond quality metrics such as MOS, path/performance outages and route change activity, as well as the obtained improvements. However, showing all the results in a raw form would no be feasible or necessary, unless some data aggregation would be clearly needed. Depending on the monitoring test/metric, proper levels of aggregation (e.g., aggregation windows of 30 or 60 minutes) it should be chosen. Since the reporting of results is an optional IRC feature whose interest for this research is limited, we will not deal with the reporting functionality of IRC systems.

3.4 IRC Functional Architecture

Figure 3.2 illustrates the overall architecture planned for an IRC system, including its core modules and basic components, as well as its interactions and data flows. This architecture was built to contain a minimalist set of functions, while bearing in mind the importance of the architectural principles for IRC systems that have already been discussed. Following this, we outline the main aspects relative to the functionality of each module, and how its components work together. Afterwards, in Sections 3.5 and 3.6 the key mechanisms of the core modules are described in detail.

3.4.1 Architecture Components

An IRC continuously monitors the performance of each available path for a destination to intelligently select a best quality ISP, or equivalent best quality path. For this reason, the architecture planned for our IRC system comprises two core functional blocks, and a management interface – a Monitoring and Measurement Module (MMM), and an intelligent Route Control Module (RCM). An optional Reporting and Viewer Module (RVM) could be developed to provide a broad set of reporting options, such as statistics about traffic demands, and the reliability, performance and stability of paths throughout each ISP.

3.4.1.1 Monitoring and Measurement Module

Path monitoring and traffic forecasting are central functions for achieving good path switching decisions, since selecting the best path to pick, depends on the performance of the multiple available path choices, and the traffic volume to a destination. Accordingly, the monitoring and measurement module of an IRC comprises a Path Monitor (PM), and a Traffic Forecaster (TF); and a Network Information Database (NID) to store the data.

Path Monitor. This component is responsible for capturing, on a real-time basis, the network performance attributes of paths, which should provide natural information about the available Internet resources. In this monitoring process, two methods can be employed, either independently or combined, to measure the performance attributes of paths: a) Active monitoring; and b) Passive monitoring.

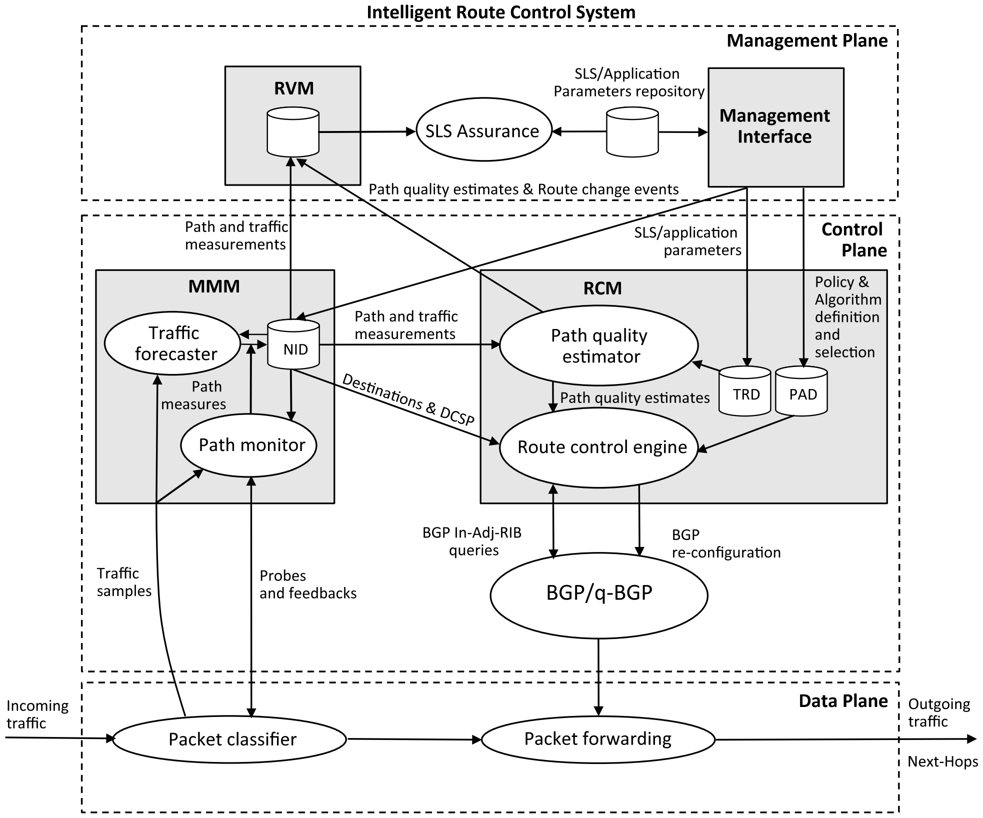


Figure 3.2: IRC systems architecture and data flow.

With regard to active monitoring, it is assumed that the PM is endowed with a mechanism to spawn, at a given frequency, small probes targeting the remote destination through every available link that connects the local AS to the Internet. The streams of probes are then monitored to determine the performance of the network, assuming implicitly that the values measured from the probes represent the available network resources accurately.

Methods like ICMP (Internet Control Message Protocol) time-stamp Request and Reply with a specific mark on the Differentiated Services Code Point (DSCP) field of IP packets (e.g., set to `AF11` (Assured Forwarding Class 1 Low)) can be used to obtain the performance metrics, such as the One-Way Delay (OWD), the One-Way-Loss (OWL) or the Round-Trip Time (RTT) [59].

Other metrics, such as the Available Bandwidth (Avail-Bw) or the available link capacity, can require more sophisticated methods and the sending of n packet bulks of k packets to address difficulties, such as limited system timer resolutions and to avoid extra latency caused by IP fragmentation and reassembly [124,125]. For instance, these methods can estimate the bottleneck capacity, C, by identifying a good packet bulk (say m-th sample), the one with the minimum delay sum S_m, and then making the ratio of the sum of packet sizes of the bulk with the bulk delay dispersion D_m, i.e., $C = \frac{kP}{D_m}$, where P is the size of each probe packet (for extra details see [125]).

In contrast, passive monitoring observes the network traffic without sending extra traffic into the network, while collecting the data regarding the path performance attributes. In a similar way to popular flow monitoring (e.g., Cisco IOS (Internetwork Operating System) NetFlow [126] and Remote Network MONitoring (RMON) [127]) tools, PM listen to the traffic sent to or from its AS and try to record important traffic events. For instance, when following elastic traffic, PM might use the TCP (Transmission Control Protocol) timeouts or the TCP acknowledgment mechanism to derive the RTTs. More specifically, PM can monitor pairs of TCP segments (sent by local TCP sources) and the corresponding TCP acknowledgements (sent by the remote TCP sinks).

There are, however, two main disadvantages in using passive monitoring. First, it requires end-user/application traffic, otherwise PM will have no data to collect. Second, it may require recording large amounts of traffic, and thus the use of additional techniques to reduce the monitoring overhead (e.g., filtering, and sampling).

Traffic Forecaster. This component is responsible for forecasting the traffic demands. During a given time-window, it samples data packets to derive a traffic matrix containing information about outgoing traffic volumes for each destination. This function can be justified on two grounds. First, if the amount of traffic to be assigned to a new path is known, we can address the IRC self-load problem, and thus avoid path oscillations. Second, when computing the corresponding traffic volume rankings, it is possible to identify the top receivers, a.k.a, popular destinations, and thus to configure the IRC system so that it only focuses on the traffic for these destinations. In this way, we can control the overall burden on the IRC system and the overhead on the network caused by probing.

It is worth noting that simple traffic predictors, such as Last-Value, Moving Average or Low-pass Exponential Moving Average (LV, MA and LpEMA), may be used to track the traffic accurately, as well as other more complex predictors, such as Auto-Regressive Integrated Moving-Average (ARIMA) (that combines past traffic volumes and/or errors linearly) [128] and Neuronal Networks (NN) (here the basic idea is to train a NN with past traffic volumes in order to predict future values) [129].

However, when deciding which predictor to choose, we should study the trade-off between its complexity and predictor errors (defined by the difference between the value of traffic volume estimated for the next interval of time and the real value of traffic volume in that interval). In the evaluations of this Chapter, we employed the classical Moving Average (MA) predictor, since, as in the case of other studies, we found that there is no advantage in using complex predictors when the performance achieved is almost the same as with simpler predictors [123, 130]. Further details related to this subject can be found in Appendix A.

Network Information Database. The path performance and traffic demands information captured by the path monitor and traffic forecaster components, will be stored in the Network Information Database (NID). When the latest measures are known from the monitoring and traffic forecasting, the NID signals the Path Quality Estimator (PQE) in the Route Control Module (RCM), as it will be described in the next subsection.

3.4.1.2 Route Control Module

The Route Control Module (RCM) comprises the core functionality of an IRC. The role of the RCM is to dynamically choose the best ISP/path for a destination, depending on both the path monitoring data, and traffic forecasts, as well as the specific quality parameters or requirements of the end-user network/application. In view of the fact that the IRC software might need to map the performance measures into a user/application metric, might know multiple paths to a destination, as well as wanting to provide a stable routing, we have divided this functional block into two main functions. First, the feasibility and quality of each path is evaluated. Second, the best path is selected by observing a switching policy and algorithm. These functions led to the proposal of the following two components: a Path Quality Estimator (PQE); and a Route Control Engine (RCE).

Path Quality Estimator. This functionality helps the IRC to know if a path is feasible and about its potential quality. It includes two coupled features: a) Feasibility tagging; and b) Quality evaluation.Path quality estimation first encompasses path feasibility checking. This consists of verifying whether the performance estimates for each path are compliant with the performance or equivalent quality bounds inscribed in a negotiated SLS (Service Level Specification) or in the application specification. Alternatively, these bounds can be obtained from the E-model from ITU's G.107/G.114 recommendations for good quality [108, 109]. As the outcome of this verification, PQE attaches a boolean tag T to each path. Feasible paths – the ones that are compliant with traffic requirements – are tagged with $T = 1$. Otherwise, the paths are tagged as infeasible, i.e., with $T = 0$. For instance, if we consider that the end-to-end delay must be less than or equal to 40 ms, a path with a delay of 50ms receives a tag $T = 0$. In contrast, a path with a delay of 30ms receives a tag $T = 1$. An infeasible path is not necessarily an invalid path, but it must be ignored by the path selection process.

On the other hand, the goal of the quality evaluation is to translate the path monitoring measures and traffic forecast into end-user network/application quality estimates, so that the IRC algorithm can select the (best) path that optimizes the performance characteristics of the traffic. This translation is based on mapping of the performance measures into a specific quality estimate using, for instance, a cost or a utility function. The resulting numerical values are finally used to evaluate the path quality, assuming that they accurately represent the application or network quality estimates. As described next, then, the best path must be the one that provides the highest quality among all the feasible paths, or alternatively, it might be simply a good enough quality path among all the feasible paths.

Route Control Engine. The intelligent Route Control Engine (RCE) is a central component of an IRC for assigning traffic to egress links, or equivalently to paths. For a BGP-based IRC system, the functionality of the RCE can be divided into two basic functions: a) Path selection; and b) Path switching.

We set out by describing the IRC path selection process. For a destination, an IRC selects the best path among the feasible paths available within the Adjacent RIBs-In of routers (e.g., direct paths, also called active paths, and indirect paths, also called alternative paths). This tasks involves iterating through every alternative feasible path to compare its quality estimate with that of the current active path. During the design of the path selection algorithm, it is important to ensure that outputted paths are selected deterministically, in the sense that their selection has no dependency on the order of comparisons.

Following this, an IRC proceeds to the path switching, and basically it comprises the insertion of the best selected paths into the Loc-RIBs of border routers, and updating their IP forwarding tables. This task requires a proper re-configuration of the underlying BGP routing, to ensure that the active path to forward packets corresponds to the one that has been chosen by the path selection process. A straightforward method is to handle the BGP route attributes properly so that they reflect the order of path that is found. For instance, IRCs might tweak the `LOCAL-PREFERENCE` route attributes to indicate the degree of preferences for the routes, as compared to other available routes for the same traffic.

Finally let highlight that knowing that a path performs better than the active one, should not be a sufficient condition to pick that path because of stability issues. There are two base path switching policies that can be observed by IRCs to address these issues: Choose-Best (CB) and Choose-Good (CG) [89]. According to the former policy, an IRC switches paths *whenever* it finds better paths in terms of quality. According to the latter, an IRC *only* switches paths if the characteristics of the active path are not sufficient to allow it to handle the traffic requirements. If this is the case, it picks any good alternative path. Later on in Appendix B, we introduce and evaluate a combination of these base policies, i.e., a Choose-Best-Choose-Good (CBCG) policy.

In Section 3.6 further details are given about the whole path switching control and related algorithms, and the BGP re-configurations for reflecting a given path ordering.

Traffic Requirements Database (TRD), and Policies and Algorithms Database (PAD). These databases basically store the traffic quality parameters that are used by the PQE to calculate the path quality estimates, and the policies and/or path switching algorithms that can be supplied by the RCE. These data are inserted by the management interface, as it will be described in the next subsection.

3.4.1.3 Management Plane Functions

This subsection describes a set of high level functions that are related with the main management tasks that are required to set-up the IRC, and display the overall performance achieved by the system.

Management Interface. An IRC should provide a flexible definition of the traffic goals to be achieved, as well as the policies that must be enforced during the path selection, and the chosen path selection algorithm. To achieve this goal, an IRC should first provide a management interface, where the operator can select, among a panoply of traffic goals the most proper option for his AS, users or applications. By means of this interface, the operator can select a path switching policy, such as for instance, *choose good delay with reasonable bandwidth* or choose *lowest delay*. In the case of prior negotiations of SLS (Service Level Specification) with peering ISPs, the IRC interface should allow the operator to import the traffic goals inscribed in the negotiated SLSs.

Second, an IRC should be able to represent the traffic goals in the form of a set of criteria, in other words in terms of the quality parameters being assigned to paths. Every traffic goal must be represented by one criterion, but complex goals may be represented only by several criteria. On a optional basis, manual fine-tuning of the weights of each criterion can be provided to the operators. These criteria are afterwards used by the IRC to determine how each alternative path is able to achieve its target traffic goals.

Finally, these set-ups – end-user network/application quality parameters or criteria, path switching policies and algorithms – are then propagated to the MMM and RCM modules to operate accordingly.

Reporting and Viewer Module (RVM). This optional module provides a broad set of reporting options, such as statistics about the traffic demands, and the reliability, performance and stability of paths throughout each ISP.

SLS Assurance. With the aid of the RVM, this function compares the overall path performance and traffic statistics to the performance/QoS levels specified within the contracted SLSs to confirm that the ISP is honoring the agreed services level, as well as if the stub network is not violating them, avoiding thus extra charges.

3.4.2 Joining the Pieces Together

Now we have an architectural overview of an IRC system, we are in a position to describe how all the pieces work together. We decompose a complete intelligent Route Control Cycle (RC) into four main phases, and each phase into a certain number of steps. Following, the numbers in brackets refer to the order number of each step within an RC.

Initialization Phase. This phase encompasses (1) the setting-up of the target destinations, (2) IRC discovery and (3) Peering session establishment with remote IRC (steps 2-3 are only required if there are case of cooperative IRCs). In subsequent RCs, there is no need to perform this phase again if the set of target destinations is invariant.

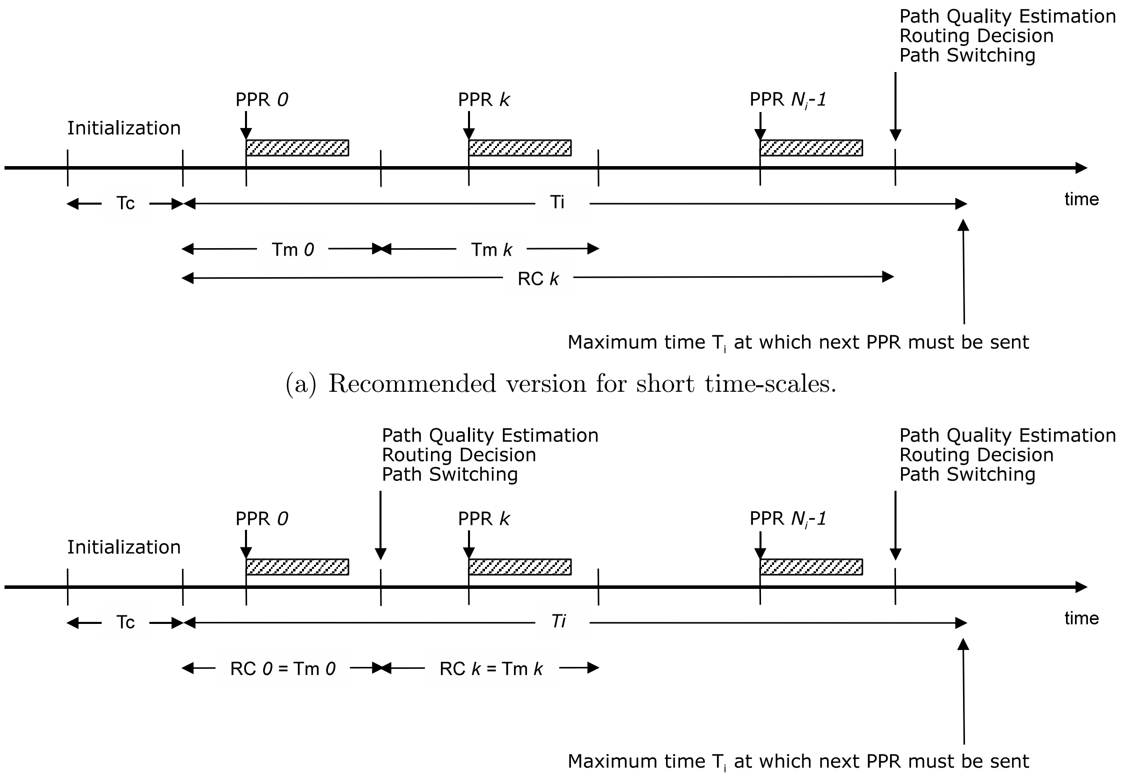
Monitoring and Traffic Forecasting Phases. Once an IRC session has been established, (4) Path Monitor (PM) probes all the available paths to each target destination by means of Probe Packet Requests (PPR). As depicted in Figures 3.3(b) and 3.3(a), probes can be sent at instants equally or randomly spaced. PM on the receiver side returns the path performance statistics captured by the probes, such as OWD, RTT or Available Bandwidth (Avail-Bw), in a Probe Packet Acknowledgment (PPA). On receiving a PPA acknowledgement, (5) PM smooths and exports the path performance measures/estimates to the Path Quality Estimator (PQE). In turn, (6) Traffic Forecaster (TF) measures and ranks the traffic demands using a given predictor. At the end of a given time-window, the Network Information Database (NID) exports these data to the PQE.

Path Quality Estimation Phase. Following this, (7) Path Quality Estimator (PQE) checks the feasibility of each path and tags accordingly. For each feasible path, it combines the performance measures of paths and traffic forecasts to estimate the path quality (e.g, in a routing cost or utility value). Next, (8) PQE export these data to the intelligent Route Control Engine (RCE).

Routing Decision and Path Switching Phases. (9) The intelligent Route Control Engine (RCE) compares the quality metrics to rank the feasible paths and select a single best feasible path for each destination (the top path). Finally, (10) the RCE reconfigures BGP routing by tweaking the BGP route attributes according to the computed ranking. If the top path differs from the active path, BGP replaces the later one from the BGP Loc-RIBs. At this time, the IP packets of the traffic flows will be routed through the best path (or through the best ISP connection).

Figures 3.3(b) and 3.3(a) illustrate two versions of a time-line for an IRC routing cycle with randomized active probing, depending on the path monitoring timescale. In these figures, T_c denotes the interval of time needed for initialization of the IRCs, $RC\,i$ denotes a routing cycle period i, T_{mk} denotes the round-trip time for the pair of probes k (PPR-PPA), and T_i denotes an interval of time through which N_i random sampling times are distributed uniformly.

When path monitoring operates in short timescales, each routing cycle can be timed to coincide with a train of N_i pairs of probe packets to reduce the number of routing decisions, and thus to avoid frequent path changes (see Figure 3.3(a)). In turn, when path monitoring operates in medium and large timescales, each routing cycle can be



(a) Recommended version for short time-scales.

(b) Recommended version for medium and large time-scales.

Figure 3.3: Time-line of IRC routing cycles with randomized active probing.

timed to coincide with each pair of probe packets for fast response to performance changes (see Figure 3.3(b)). Thus, at most N_i routing cycles are performed during a complete measurement time slot T_i.

3.5 Path Monitoring Algorithms

This section describes the monitoring mechanisms used in the IRC system that has been designed during the investigation of the intelligent route control technique, together with the mechanisms proposed for handling the loss of probes, which play an important role in the time the IRC system needs to detect and recover from a path failure. Thus, this section also devotes attention to analyzing the timescale of the IRC reaction to failure events.

3.5.1 Monitoring using Active Probes

In the designed IRC, the performance attributes of paths are obtained through active probing [86]. In the following subsections, we describe the use of the probes, and the sampling process employed by the IRC.

Probe Operation. Each IRC monitors K paths toward the remote peer or AS using small probe packets. Once a probe session with the remote peer starts, the prober mechanism initiates a sequence number counter (`seqnr`) that identifies each probe, and a `next_time_to_probe` timer per path. When this timer expires, the IRC sends a probe packet filled with current `seqnr`, and starts a Request Time-Out timer `RTO`[1]. The `seqnr` is incremented each time a probe is sent. When a reply is received, the IRC updates the performance estimate. If no reply is received within `RTO`, the IRC declares the probe is lost.

As can be observed in Figure 3.4, which illustrates part of a NID and other software active probing mechanisms, each sent probe is temporally stored in a hash-map data structure (the prober calls `put(seqnr,probe)`, where `probe` parameter is a probe). Once the corresponding reply arrives, the prober removes the probe from the hash-map (it calls `remove(seqnr)`) and sends the sampled value to the NID database. Regarding the handling of lost probes, a thread, called expiration thread, iterates continuously over the hash-map (it calls `get_next()`), in order to find probes with an expired `RTO`, If this is the case, it removes the probe from the hash-map (it calls `remove(seqnr)`), and sends a positive infinite number ($+\infty$) to the performance database.

The NID to store the last performance measures is based on various First In First Out (FIFO) queues with a configurable size. There is a FIFO queue assigned to each metric – for instance, one queue for latency and another for available bandwidth –, and only samples aged less than `RTO` stay in the queue. In sum, a FIFO queue works like a sliding window, and the samples older than a given interval, say $dt >> RTO$, are removed from the queue.

The measurement of latency – OWD (one-way delay) and RTT (round trip-time) – experienced by traffic was assumed as the basic feature of the designed IRC, because the performance of the most of the applications (e.g., voice, video conferencing and interactive data applications) is affected by the degree of the transmission latency [108, 109]. By default, the planned IRC measures RTTs as it only requires basic cooperation from the target destination, i.e., response to probe packets.

RTT is measured by means of a process similar to the ICMP time-stamp Request and Reply (see Figure 3.5) [59]. First, the Path Monitor (PM) fills the current time (the number of ms past midnight, UTC) in the ICMP's originate time-stamp field and sends the request. Then, the target AS or system fills in the receive time-stamp when it gets the request, and the transmit time-stamp when it sends the reply. Finally, the RTT is calculated through the difference between the time when the reply is received and the originate time-stamp.

To track RTTs, we use a smoothing predictor based on the *median*, as in Equation

[1] Although different, to facilitate understanding, this acronym was made identical to the timer RTO (Retransmission Time-out) associated with the TCP (Transmission Control Protocol) sources.

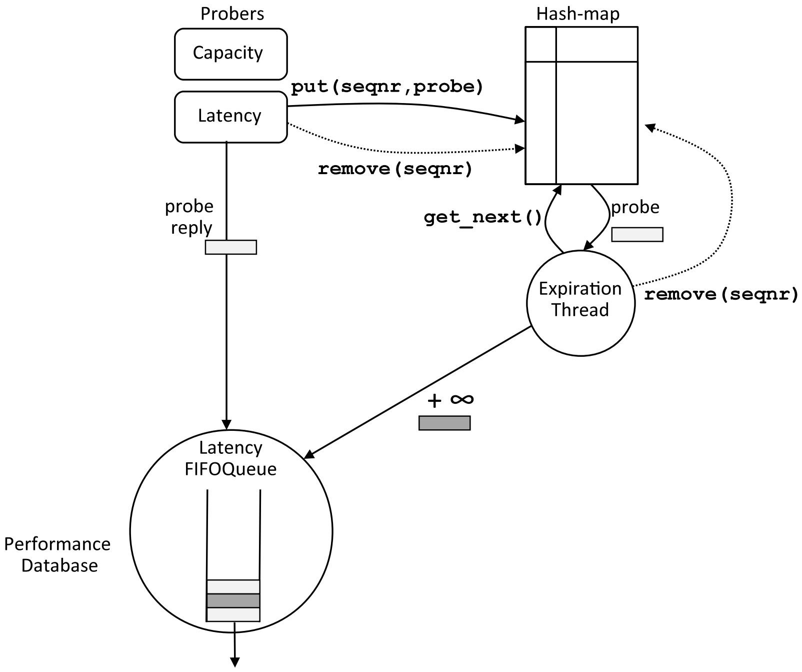


Figure 3.4: Detailed Network Information Database (NID), and active probing software components.

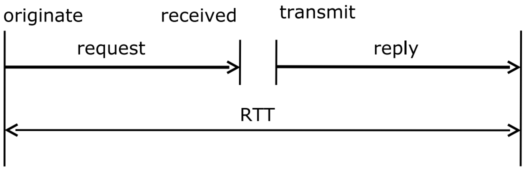


Figure 3.5: ICMP time-stamp Request and Reply method to derive RTTs.

(3.1), where $\overline{RTT}$ is the RTT estimate, W is the number of fresh samples in the queue and n is the sequence number of the last fresh sample. The choice of a Moving Median (MM) estimator, instead of a Moving Average (MA) or an Exponential Weighted Moving Average (EWMA), was made because it facilitates the handling of lost probes and the reactivity control of IRCs in a relatively straightforward manner. These issues are addressed in detail in Section 3.5.2.

$$\overline{RTT} = Median\left(RTT(k)\right),\ k \in [n - W + 1, n] \tag{3.1}$$

Sampling Process. In the context of IRC systems, when using active probing there are two critical issues that must be addressed: a) synchronization between probes; and b) probing overhead and burden over IRCs.

The first issue can be immediately addressed at the monitoring level by adding randomization to the sampling times. In this way, there can be a reduction in the likelihood of overlapping measurement periods through distinct paths, which is an important factor in probe synchronization [89,104]. For this reason, our base design employs a Pseudo-Random Poisson (PRP) sampling process, instead of periodic sampling (as in [131,132]). A pure Poisson method was not adopted because Poisson distribution is occasionally unbounded, and as a result, lengthy spaces between sampling times can be experienced. As a consequence of this, extra inaccuracy might be introduced into the state of the IRCs, which is an unwanted behavior to many real-time applications.

More precisely, according to a PRP process, N_i random sampling times are uniformly distributed over consecutive time slots with a maximum size T_i (in seconds), as in Equation (3.2), where $F(t)$ is the corresponding probability distribution function.

$$F(t) = Uniform\{N_i, t \in [nT_i, (n+1)T_i]\}, \forall t \geq 0 \wedge \forall n \geq 0 \tag{3.2}$$

To achieve the second goal, an IRC must first use a conservative-enough frequency of probes – defined as $f_i = \frac{N_i}{T_i}$ –, while keeping efficient routing. This avoids unnecessary waste of network bandwidth, and processing capacity due to probing. To illustrate this, in the default set-up that is used in the evaluations, we calibrated the probers to send, on average, 800 probes per hour, more specifically, $f_i = \frac{N_i}{T_i} = \frac{8}{36}$Hz, which is much less demanding than the 10Hz used in [89]. Second, when assuming that an IRC is employed to control all the traffic of an AS, and not only the traffic of a single application, an IRC must focus on the traffic toward top receivers, the so called popular destinations, in order to reduce the number of paths to probe (i.e., to about 10-20% of the total number of receivers). This method is supported by the distributions of traffic volumes per prefix, which are generally consistent with a Zipf-like distribution [133]. Hence, this means that the Path Monitor, as well the route control logic, have to be frequently updated (or will be able to compute on the basis of the number of application calls that are requested for each destination) with the set of popular destinations.

3.5.2 Service Outages and Recovery

In this section, we focus on a significant problem – how to ensure that an IRC is able to react in a proper timescale to path failures. After describing the problem, we state, in precise terms, the reaction time of our IRC system, i.e., the time it needs for detection and recovery from a path failure. Since this time depends on the actual value for the Request Time-Out (RTO) time used to declare that a probe packet is lost, we outline two mechanisms for the dynamic adaptation of RTO timers that can be included into the IRC Monitoring and Measurement Module (MMM): a Jacobson-based and Plot rule-based mechanism. Finally, we describe how to blend both mechanisms to enable them to handle spurious timeouts and thus prevent the original settings from running into consecutive spurious timeouts.

3.5.2.1 Description of the Problem

The IRC routing decisions rely on performance measures made on the basis of the active probing method. Unfortunately, an improper handling of lost probes can affect the accuracy of the path failure detection and the IRC time reaction. Examples of this behavior are illustrated below.

Consider a latency-driven IRC and three paths, $p1$, $p2$ and $p3$. Assume that the IRC initially allocates the traffic to $p1$, and that the RTO has just expired. This can lead the IRC to declare that $p1$ is inactive and to start the process of reallocating of traffic from $p1$ to $p2$ or $p3$. However, this does not necessarily imply that the probes have been lost and $p1$ has failed. In effect, a transient congestion can significantly increase the queuing delay and thus the latency of the probes beyond the actual RTO. Hence, a very short RTO can lead to premature path changes. Another related effect is that the number of paths for exploitation by IRCs during the improvement of the performance can be dramatically reduced due to spurious timeouts, which can risk leading lead to performance degradation. In the example, this means that $p2$ and/or $p3$ can also be declared as inactive, as well as $p1$. In contrast, very long RTOs can lead to a slow reaction and thus to late path changes.

In short, to prevent unnecessary or untimely path changes it is crucial to find an adequate RTO and a reaction time value that is at an equilibrium point between both extreme cases.

3.5.2.2 Analysis of the Timescale of IRC Reaction

This section provides a formal description for the reaction time of an IRC and the Request Time-Out (RTO) which is used to declare that a probe is lost.

Determination of the reaction time T_R. Our goal is to find a worst-case value for T_R. When using the Pseudo-Random Poisson process and median to smooth the sampled RTTs, the IRC only detects that a path is down after at least one half-window of lost probes. Thus, T_R can be expressed by the sum of the time needed to transmit one half-window of probes, and the time needed to detect the last probe lost as in Equation (3.3), where $T_X\left(\frac{W}{2}\right)$ refers to the time needed to transmit one half-window of probes, W refers to a fixed number of probes sent during T_w, T_w refers the size of the sampling window in seconds, and $RTO(k)$ refers to the last RTO timer to expire. It should be noted that this rests on assumption that during a path failure, the RTO remains unchanged until a reply arrives.

The latter assumption contrasts with the exponential (Retransmission Timeout) RTO back-off algorithm adopted by TCP, that doubles RTO after each segment retransmission. In fact, since IRCs do not retransmit probes, it does not make sense to apply back off RTO and, if it was applied, it would also introduce an extra waiting time before detecting that a path is down.

$$T_R = T_X\left(\frac{W}{2}\right) + RTO\,(k) \leq T_w + RTO_{max}, k \in \left[1, \frac{W}{2}\right] \tag{3.3}$$

To give an illustration, when assuming that the probes are sent according to an uniform distribution, the expected value for T_R value is roughly equal to $\frac{W}{2}.\frac{T_w}{W} + D_{max}$, where W refers to a fixed number of probes sent during a window of time T_w and D_{max} is the upper RTT bound for traffic. Here, we consider the maximum allowed latency for traffic as the limit value for the RTO timer. The probability of obtaining this value is bounded by the probability of transmitting $\frac{W}{2}$ probes in an interval of $\frac{T_w}{2}$ seconds as in Equation (3.4). As expected, this probability falls when the size of the sampling window increases. To exemplify this point, we get a probability of 0.5 for a window of just two probes. Then, it drops to 0.25 for a window of four probes. Therefore, apparently T_R is dominated by the relative window size compared with the number of probes. This implies that when the relative size of a window is large, it is expected that $T_R < T_w/2$. Otherwise, $T_R \to T_w/2$.

$$P(X_i \leq \frac{T_w}{2}, \forall i \in \left[1, \frac{W}{2}\right]) = \prod_{1}^{\frac{W}{2}} \int_{0}^{\frac{T_w}{2}} \frac{1}{T_w} dt = \prod_{1}^{\frac{W}{2}} \frac{1}{2} = 2^{-(\frac{W}{2})} \tag{3.4}$$

In a worst-case scenario, we may expect T_R to be equal to $T_w + RTO_{max}$, which corresponds to the extreme case where the first probe and probe number $\frac{W}{2}$ losses are being separated by the size of the window T_w and $RTO(k) = RTO_{max}$.

Determination of the request timeout RTO**.** Ideally, the optimal RTO is the actual RTT, which is unknown to the IRC. Thus, the RTO that is to be used by the IRC, must be based on the available estimates of RTTs, such as mean, variance or other statistical attributes. However, a large enough minimum RTO, RTO_{min}, is needed as a security setting against spurious timeouts and thus undesired path switches. In fact, a RTO lower than the actual RTT, leads to spurious path switches. On the other hand, very large RTO values should be avoided, since they might lead to a slow reaction to path failures (especially when T_w is small). In practice, this corresponds to a $RTO >> D_{max}$ and $T_w \approx RTO$. In short, there is a hard problem to address, that is how to tune the RTO in such a way that the IRC is able to clearly differentiate between a high probe loss rate, high latency and true path failures.

Our goal is thus to find a proper range of values for RTO. First, we consider a conservative setting for the minimum RTO. The minimum value for RTO can be statically fixed as the sum of the receiver clock granularity, Cg, and the latency upper bound for traffic D_{max} or simply just D_{max} (see Equation (3.5)), assuming that the time granularity of the operating systems of the probing devices is very small. Second, to define the maximum value for RTO, RTO_{max}, we consider the worst-case scenario, when the IRC is idle. In this situation the instants of probing are located in the antipodes of two consecutive probing intervals.

$$Cg + D_{max} \leq RTO \leq 2T_w \tag{3.5}$$

Although these bounds define a range for the RTO, it is still unclear which value should be used. Thus, the best approach to cope with all the concerns surrounding the RTO, is to use an on-line mechanism that finds the best RTO in a dynamic way by taking into account the last measured RTTs. This approach also implies that RTO_{min} is not constrained by D_{max}, and thus can be lower than D_{max} when the network is not congested, or greater than D_{max} when network congestion is high.

3.5.2.3 Mechanisms for RTO Adaptation

In this section, two mechanisms for RTO adaptation are provided, which are revised versions of a popular algorithm and a simple parametric technique. The first is based on the Jacobson's algorithm which is implemented by TCP (Transmission Control Protocol) to adapt segment retransmission timeouts [134]. The second is based on the box-plot parametric technique, which is adopted in many research fields, to rapid summarize and interpret data, such as on the detection of Internet traffic anomalies [135, 136].

The Jacobson-like algorithm. In a similar way to the TCP retransmission timeout proposal in [137], we assumed that `RTO` is computed as a function of the smoothed `RTT` and the smoothed mean `RTT` variation, $\overline{D}$, as in the first Equation of (3.6), where D denotes the `RTT` variation and β represents the contribution of the estimation error $|\overline{RTT} - RTT|$ to the current estimation $\overline{D}$. As recommended in [134], $\alpha = 0.125$, $\beta = 0.25$.

$$\begin{aligned} &RTO = \overline{RTT} + 4\overline{D} \\ &\overline{RTT} = (1-\alpha) * \overline{RTT} + \alpha * RTT \\ &\overline{D} = (1-\beta).D + \beta\left|\overline{RTT} - RTT\right| \end{aligned} \tag{3.6}$$

The Box-plot rule based algorithm. A box-plot is a visual method that can be employed to depict the RTT data using statistical attributes such as the smallest RTT measured (min), lower quartile ($Q1$), median, upper quartile ($Q3$), and largest RTT measured (max). The difference $Q3 - Q1$ is called the Inter Quartile Range (IQR). Using these attributes, we can "plot the box" to find the upper limit beyond which any RTT sample will be treated as an anomaly (see Figure 3.6). This upper limit can then be used to set RTO as in Equation (3.7). According to this technique, if an idle interval between the instant the probe was sent and current time is $1.5IQR$ higher than $Q3$, the corresponding probe is declared to be lost.

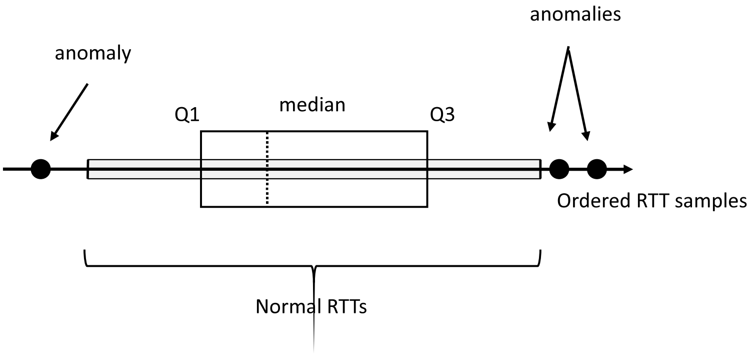


Figure 3.6: Illustration of the use of the Box-plot rule to detect anomalous RTTs.

$$RTO = Q3 + 1.5IQR \tag{3.7}$$

If the RTTs are distributed in accordance with the Normal distribution, the region between $Q1 - 1.5IQR$ and $Q3 + 1.5IQR$ should contain 99.3% of observations [135]. Hence, the choice of $1.5IQR$ boundary makes the box-plot rule equivalent to a 3σ technique for Normal data, where σ denotes the standard deviation of the Normal distribution.

To avoid spurious timeouts arising from the situation where $RTO = Q3 = RTT$ when successive RTTs are equal, we have to assume $IQR = max\,(Q3 - Q1, Q2 * \delta)$, where δ is a small safe margin to ensure (e.g., 0.05) that RTO becomes slightly larger than the RTTs.

Finally, it should be highlighted the disadvantage of this technique, that is it must be trained in advance before it can provide the first accurate RTO estimate. For this purpose, and for computing subsequent RTO estimates, we assumed a series of RTTs corresponded to a sliding window of probes with a large enough size. To be more precise, $6.W$, where W is the size of a window in a number of probes.

Extensions to Jacobson's algorithm and Box-plot. A spurious timeout occurs if no probe is lost, but the RTO is smaller than RTT. To fix this issue and avoid subsequent spurious timeouts, the default values of K used by both Jacobson's algorithm and the Box-plot rule, that are equal to 4 and 1.5 respectively, must be replaced by proper values. In concrete terms, K must be equal or higher than the value for which RTO is equal to the last measured RTT. Subsequently, to avoid excessive RTOs, and thus slow IRC reaction, K is adapted on the basis of a Subtractive Decrease (SD) policy.

As a result, K is given by Equation (3.8), where $K_{default}$ is the default K used by both Jacobson and Box-plot rule algorithms and Δ is a small step given by $\frac{K-K_{default}}{W}$. In the expression, $\overline{RTTQ_3}$ and $\overline{DIQR}$ should be replaced by the corresponding variables of the algorithm in question.

$$K = \begin{cases} ceil\left(\frac{RTT-\overline{RTTQ_3}}{\overline{DIQR}}\right) & \text{after a spurious timeout} \\ \\ min\left(K-\Delta, K_{default}\right) & K > K_{default} \end{cases} \tag{3.8}$$

3.6 Path Switching Control

In this section, we discuss the key issues in the control of path switching, and describe the path switching algorithm used in our IRC system design. In view of the fact that the path switching control is a combination of three functions, namely Triggering of Path Switching, Path Switching Decision, and Shifting of traffic over ISPs, we have divided this section into three subsections.

3.6.1 Triggering of Path Switching

Our IRC system is designed to support two distinct routing schemes: a) on demand routing; and b) traffic engineering-like routing. Figures 3.7 and 3.8 provide an overview of the functionality of each scheme. As can be observed, the key aspect that differentiates both schemes are the events that trigger a routing change.

On demand routing. According to this scheme, the IRC carries out searches for feasible paths whenever a new incoming traffic flow request arrives. In this model, the IRC performs explicit Admission Control (AC) of new arrived traffic flows by probing all available paths within Adjacent RIBs-In to determine if there is any feasible path – a path that complies with the performance requirements of the traffic. The IRC only admits the new traffic if there is at least one feasible path and the level of congestion is still low after admission. Otherwise, the new traffic flow must be blocked. If there are multiple feasible paths, the Route Control Engine (RCE) triggers the path switching decision process to retrieve the best path from among the set of feasible paths.

Since the on demand routing model includes an AC mechanism, one of its main benefits is that it provides a robust method to protect the performance of the admitted traffic, especially during overload situations, which is to block additional traffic. This also provides an effective method of controlling the CPU and memory consumption in the IRCs, and the overhead on the network caused by active probing, because only paths to the destinations of the admitted traffic are monitored. However, this is achieved at the expense of a slower reaction. If preferred, this scheme can be complemented by precomputing a set of feasible paths in advance. Although this alternative might reduce the response times, unless it knows the traffic destinations in advance, it may have an adverse effect on the advantage of this scheme in terms of the burden over IRCs, as well as the additional network overhead caused by the IRCs signaling messages.

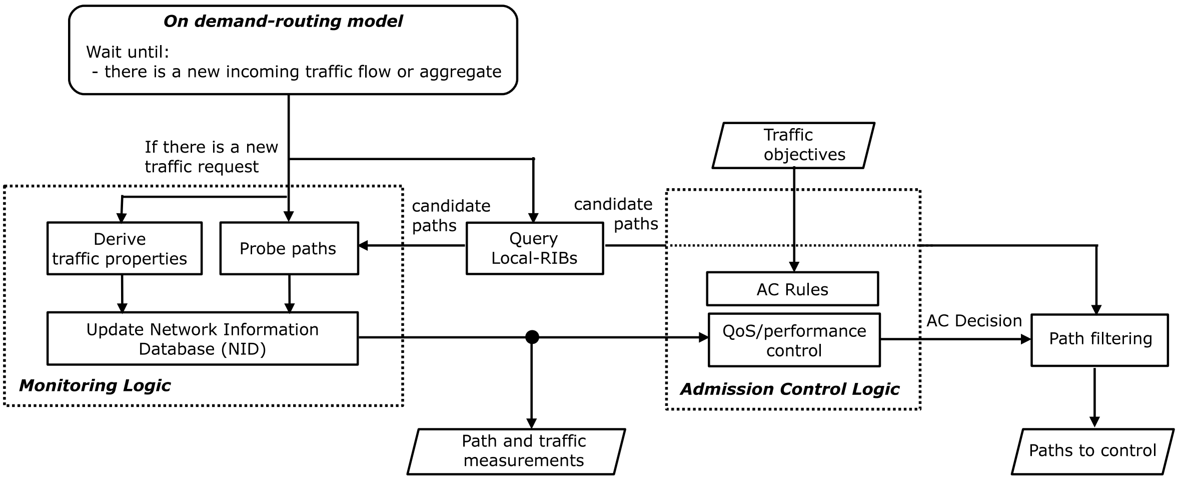


Figure 3.7: On demand-routing scheme.

Traffic engineering-like routing. In this scheme, there is no explicit admission control mechanism. It is assumed that the network is carefully provisioned (e.g., by capacity planning and negotiating a Service Level Agreement (SLA)) to ensure the overload is unlikely to occur; this implies that all the traffic is admitted. However, traffic demand changes (e.g., due to source behavior change of traffic to new popular destinations) can lead to unbalanced loads or overload situations on the network. To overcome this problem, the IRC periodically derives the traffic demands and re-distributes the traffic across the available paths, by complying with the traffic requirements.

Thus, there are two main events that can trigger the path switching decision process that is required for updating the active paths. First, if the Path Quality Estimator (PQE) finds a significant change in the path quality metric(s); and second, if after ending a some window of outgoing traffic sampling it is found there is a new set of popular destinations to follow. The main problem with this model is the extra overhead imposed over the monitoring system to track traffic and select popular destinations. In contrast, the overhead over IRCs is potentially reduced as it only focuses on a small set of destinations. A procedure that can be used to derive the traffic demands, and select the popular destinations is introduced in Appendix A.

From now onwards, we will adopt the traffic engineering-like routing model for IRCs due to its simplicity, and because it most effectively avoids the need for QoS negotiations at the application layer by means of optimizing the use of existing resources. Nevertheless, we think that the AC-based routing can be an advantageous scheme for delivering premium services, in so far as it implies per-traffic flow or aggregate quality

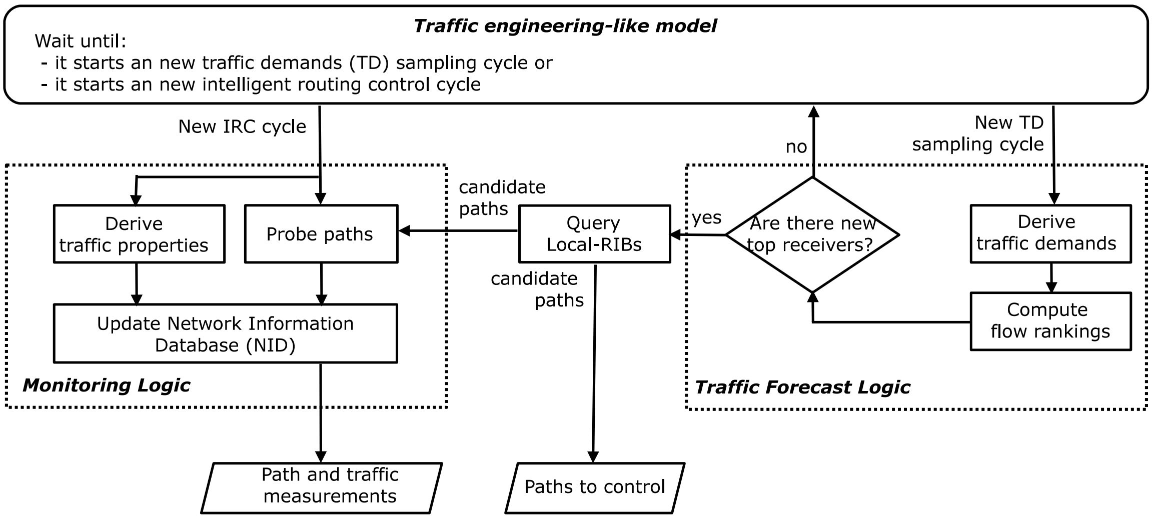

Traffic engineering-like model
Wait until:
- it starts an new traffic demands (TD) sampling cycle or
- it starts an new intelligent routing control cycle
New IRC cycle
no
New TD sampling cycle
candidate paths
Derive traffic properties
Probe paths
Query Local-RIBs
yes
Are there new top receivers?
Derive traffic demands
candidate paths
Update Network Information Database (NID)
Compute flow rankings
Monitoring Logic
Traffic Forecast Logic
Path and traffic measurements
Paths to control

Figure 3.8: Traffic engineering-like routing scheme.

negotiation and the denial of access to the service if the network risks being overloaded. However, studies have demonstrated that the AC benefits to the network decrease as the number of flow requests increase, and above all, when this technique is compared with the Capacity Over-provisioning (CO) technique in the presence of high traffic dynamics and failures, surprisingly, CO requires the same or less bandwidth to ensure the same degree of service [138, 139].

3.6.2 Routing Decision and Path Switching

This section details the issues about the whole IRC path switching process. It begins by discussing methods for path comparison that can be regarded as offering the best prospects for use in IRC systems. Following this, it provides a proposal of a complete path switching algorithm. This section concludes by discussing issues about the accuracy and stability of path switching decisions, which involves introducing the ideas about the use of predictors for tracking paths, and the enforcement of path switching policies to avoid IRC oscillations.

3.6.2.1 Path Comparison

A major challenge is how an IRC selects the path that performs better among a set of feasible paths. This is known as the path selection problem. Depending on the number of criteria being considered by the path comparison logic, this problem can be categorized into two classes: a) single-constrained path selection; and b) multi-constrained path selection, which have distinct degrees of complexity and comparison methods, as outlined below.

Single-constrained-based intelligent routing. If an IRC has to select a path that is constrained to a single attribute (or constraint), denoted as C, the IRC system is single-objective driven, and the path selection problem falls into the single-constrained path selection class. The peculiarity of this problem is that it can be solved in polynomial time and so best paths can be feasibly computed on some computational device [140].

A simple method that can be employed by an IRC to compare single-constrained paths, is to use a metric, denoted as M (a so-called routing metric), derived from performance measures along each path. More precisely, the M expression might be given by a positive quantity – a cardinal value – that is direct or inversely proportional to a number of m performance measures along each path depending on its nature. That is, whether the measures are additive or multiplicative (e.g., $M \propto \sum link\, latencies$, or $M \propto \prod link\, losses$), or concave (e.g., $M \propto \frac{1}{min(link\, bandwidths)}$), respectively. It should be pointed out that in inter-domain environments, given it is hard to obtain all the link components of the end-to-end performance measures, the calculation of M is usually made on the basis on a single end-to-end measure (e.g., one-way delay or round-trip time). Finally, the IRC picks the path with the lowest metric value.

To illustrate the use of this basic method, let us suppose that the IRC has three paths toward the same destination with the following characteristics:

- $P1$: one-way delay = 200ms, available-bandwidth = 2Mbps;
- $P2$: one-way delay = 100ms, available-bandwidth = 4Mbps;
- $P3$: one-way delay = 50ms, available-bandwidth = 1Mbps.

What would be the best path that the IRC is able to retrieve? If the IRC is delay-driven, it selects $P3$ because considering $M_i = l_i$, where l_i denotes the one-way delay measured in P_i, $argmin\left(\{M_1, M_2, M_3\}\right) = 3$. But, if the IRC is driven by the available-bandwidth, it selects $P2$ because considering $M_i = \frac{max(abw_1, abw_2, abw_3)}{abw_i}$, where abw_i denotes the available-bandwidth measured in P_i, $argmin\left(\{M_1, M_2, M_3\}\right) = 2$.

Multi-constrained-based intelligent routing. In this case, an IRC is multi-criteria driven and the problem of selecting best path falls into the multi-constrained path selection class, which is a so-called MCP (Multi-Constrained Path) problem, as the performance of paths depends on multiple attributes (or constraints).

Figure 3.9 represents the decision table that can be used by the IRC software to select the best performing path, where $C_i, i = 1, ..., m$ denote the criteria, $P_i, i = 1, ..., n$ denote n paths for the same destination, a_{ij} represents the score of the path P_j against the criterion C_i, the scale factors α_i denotes the weight assigned to the criterion C_i. The vector of values $x_i, i = 1, ..., m$ denotes the position of the path P_i in the ranking, calculated by the comparison logic function that is integrated within the RCE engine.

$$\begin{array}{cc|cccc|}
 & & x_1 & x_2 & \ldots & x_n \\
 & & P_1 & P_2 & \ldots & P_n \\
\hline
\alpha_1 & C_1 & a_{11} & a_{12} & \ldots & a_{1n} \\
\alpha_2 & C_2 & a_{21} & a_{22} & \ldots & a_{2n} \\
\ldots\ldots & & & & & \\
\alpha_m & C_m & a_{m1} & a_{m2} & \ldots & a_{mn} \\
\hline
\end{array}$$

Figure 3.9: IRC decision table for a MCP problem.

The issue is how to compare multi-constrained paths, and ensure that a path is really the best one and conforms to all criteria. It must be admitted that addressing the MCP problem is not a trivial matter and is commonly considered to be NP-hard, i.e., non-computational treatable in polynomial time [141]. This NP-hardness result opens up questions and perspectives for research in search of approximations to overcome its intractability. With this purpose in mind, we identified two classes of comparison heuristics that can be employed in the context of IRC: a) Elementary Methods; and b) Weighted Methods.

Elementary Methods. The main characteristic of elementary methods is that negligible computational support is needed. Two popular methods are the lexicographic method and the max-min method. When the lexicographic method is used, the path attributes are first ranked in the order of their importance. After this, the paths are ranked by comparing the measured values from the most important to the least important attribute. The path that has the best value with respect to the most important attribute is considered to be the best one. If there is a tie, the method proceeds to the next attribute, till the method returns a single path. When using the max-min method, the objective is to avoid the worst possible performance by maximizing the minimal (the poorest) criterion. The path that shows the highest-weakest criterion is selected as the best one. The drawback of this method is that it is only suitable when all of the path attributes are comparable;

Weighted-based Methods. A pair of weighted-based methods that can be adopted to implement the comparison logic are: a) the Metrics Combination (MC); and b) the Multi-Attribute Utility (MAU) [142, 143].

In the case of the MC method, a cost function is first applied to each performance attribute of a multi-constrained path to obtain comparable dimensionless metrics. After this, a new cost function, called composite metric, is obtained by a linear combination of these metrics $M_i, i = 1, ..., m$, so that a more preferred path obtains a lower cost value, as in Equation (3.9), where $\alpha_i, i = 1, ..., m$ are scale factors such $\alpha_i \geq 0$.

$$M = \sum_{i=1}^{m} \alpha_i M_i, \tag{3.9}$$

Scale factors α_i should be non-negative to ensure that the resulting metric value M is bounded (by a convex cone or by a convex hull). A convex cone is formed by all the conical combinations, i.e., by all metric combinations respecting the restriction $\alpha_i \geq 0$. In practice, this means that the resulting M is somewhere between individual metrics M_i, as in the left-hand graph of Figure 3.10. In turn, the convex hull is formed by all convex combinations, i.e., by all the metric combinations respecting both restrictions $\alpha_i \geq 0, \sum_{i=1}^{m} \alpha_i = 1$. In practice, this means that the resulting M is somewhere in the area limited by the convex envelope, as in the right-hand graph of Figure 3.10.

Both strategies have been adopted in many standard (e.g., Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS)) and proprietary routing protocols (e.g., Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP)).

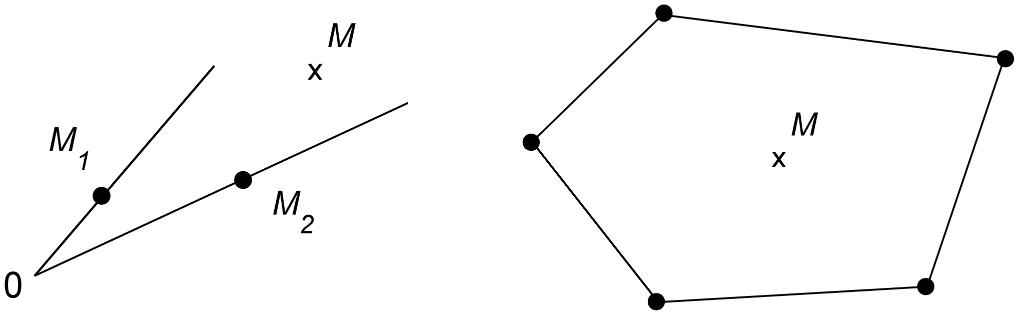


Figure 3.10: Illustrations of a convex cone (left) and of a convex hull (right).

The MAU method is quite similar to the MC method. Rather than using a cost function, it applies a utility function to convert each path attribute to a common dimensionless scale (e.g., $[0, 1]$ or $[0, 100]$), so that items with better performance obtain higher utility values. This simplest form is referred to in the MAU Theory as SMART (Simple Multi-Attribute Rating Technique), in which, transposing to the IRC case, each ranking value x_k of a path P_k can be calculated as the weighted mean of the utility values associated with each path attribute, as in Equation (3.10).

$$x_k = \frac{\sum_{i=1}^{m} w_i a_{ik}}{\sum_{i=1}^{m} w_i}, k = 1, ..., n \tag{3.10}$$

We now exemplify the use of the different methods which have been just described. Let us suppose that the IRC has the following paths toward the same destination with the following attributes:

- $P1$: one-way delay = 200ms, available-bandwidth = 2Mbps, loss rate = 2%;
- $P2$: one-way delay = 100ms, available-bandwidth = 2Mbps, loss rate = 4%;
- $P3$: one-way delay = 50ms, available-bandwidth = 1Mbps, loss rate = 8%.

What would be, now, the best path that the IRC is able to retrieve? The answer is given below depending on the method that is employed.

Lexicographical ordering method. If the IRC considers the ordering (one-way delay, available-bandwidth, loss rate), the delay is the most important path attribute; as a result, the IRC selects $P3$ as this is the path with the lowest one-way delay. But, if the IRC considers the ordering (available-bandwidth, one-way delay, loss rate), the available-bandwidth is the most important path attribute; then it selects $P1$, although in this case there were two path candidates $P1$ and P_2 because they have the same available-bandwidth. As result, the IRC had to tie-break these paths by comparing the next path attribute, the one-way delay, as well as selecting the path with the lowest latency;

Max-min method. This method requires that all path attributes to be comparable, which is not the case in this example. In view of this, the first need of IRC is to convert each path attribute to a common dimensionless scale. A possible strategy is to scale or normalize all the attributes within the range $[0, .., 1]$, when the low values are considered to offer the weaker performance. For instance, in IGRP the bandwidth attribute is obtained by dividing 10^7 by the link bandwidth. Alternatively, having considered the normalized values by the actual worst attribute for each path, it can be obtained by $P1(0.25, 1, 1)$, $P2(0.5, 1, 0.5)$ and $P3(1, 0.5, 0.25)$ respectively. As a result, the IRC should pick $P2$ because its minimal criterion has the highest value (0.5);

Metric combination method. Let us suppose that the scale factors are 0.5 for one-way delay, 0.3 for available-bandwidth, and 0.2 for loss rate. Here, when the high values are interpreted as indicating a weaker performance, the normalized values for each path are obtained by normalizing each individual attribute by the corresponding actual best case, that are $P1(4, 1, 1)$, $P2(2, 1, 2)$ and $P3(1, 2, 4)$ respectively. Consequently, the weighted normalized values are $P1(2, 0.3, 0.3)$, $P2(1, 0.3, 0.4)$, $P3(0.5, 0.6, 0.8)$ respectively. Therefore when the final sums are made of their items, the composite metrics M for $P1, P2, P3$ are 2.6, 1.7, 1.9 respectively. As a result, surprisingly the IRC picks $P3$ because it prefers paths with low delay;

SMART method. Let us suppose that the weights are 0.4 for one-way delay, 0.6 for available-bandwidth, and 0.3 for loss rate. Here, considering an utility function based on the same normalization performed earlier in the max-min method, again, the utility items for each path are respectively $a_{i1}(0.25, 1, 1)$, $a_{i2}(0.5, 1, 0.5)$ and $a_{i3}(1, 0.5, 0.25)$. Subsequently, the weighted normalized utility items are given by respectively $w_i a_{i1}(0.1, 0.6, 0.3)$, $w_i a_{i2}(0.2, 0.6, 0.15)$ and $w_i a_{i3}(0.4, 0.3, 0.075)$. Therefore the ranking values for paths are respectively $x_1 = \frac{1}{1.3} = 0.77$, $x_2 = \frac{0.95}{1.3} = 0.73$, $x_3 = \frac{0.78}{1.3} = 0.6$, where $\sum_{i=1}^{3} w_i = 1.3$. As a results, the IRC selects the path $P1$ because it shows the highest ranking value.

Discussion. Following this, we discuss the pros and cons of the comparison methods for the Multi-constrained-based intelligent routing. On the one hand, lexicographic and max-min methods are very simple methods, and require negligible computational support. However, the drawback of the lexicographic method is that it is biased toward the most important criterion at the expense of less important criteria. In turn, the max-min method is only suitable for comparable path attributes. In addition, neither of these methods takes into account multiple objectives.

On the other hand, the advantage of weighted-based methods is that a mix of multiple objectives can be incorporated in the path characterization. However, the proper choice of the weight for each attribute depends on which is regarded as more important in terms of performance for the AS administrators or applications. Unfortunately, in some cases these choices become hard to make as they may involve a range of trade-offs between different criteria given some of which compete with others. To deal with this problem, Chapter 4 describes the design of a self-adaptive cost metric which can be used by (but is not limited to) IRCs.

3.6.2.2 Path Tracking

Good path switching decisions can potentially bring about performance gains for the switched traffic and routing. To make better decisions, the intelligent Route Control Engine (RCE) is free to predict, from among the paths, which would be really the best path in terms of quality, in the near future. This issue is also significant as it involves the question of how to avoid routing instability. Path tracking is, however, more suited for a path quality evaluation based on a single quality metric, such as a composite cost metric or a utility function, rather than based on multiple quality metrics.

In concrete terms, we attempt to track the path quality estimates (e.g., cost or utility) with two distinct predictors, in terms of complexity: a very simple predictor, the Deterministic Last Value (DLV) predictor, and an adaptive but more complex predictor, the LpEMA (Low pass Exponential Moving Average) predictor. First it should be pointed out that, e_i denotes the actual metric estimate and $M(t_i)$ denotes the cost or utility computed in the last slot of time t_i (or the equivalent in the last routing cycle i).

DLV (Deterministic Last Value). DLV is the basic IRC routing algorithm. It relies on a very simple predictor – the last value predictor. The actual metric estimate e_i is therefore equal to the value of $M(t_i)$, computed in the last time slot t_i, i.e., $e_i = M(t_i)$. The IRC then selects as the best path, the path that has the smallest value of the metric e_i. It worth noting to notice that with DLV, the path switches are deterministic in the sense that when needed they are performed by the IRC with a probability P of 1. In contrast, when using the Fixed Switching Probability (FSP) algorithm (used in Section 3.9 as term of comparison), the IRC picks the best path with a given switching probability $P \in]0, 1[$.

LpEMA (Low-pass Exponential Moving Average). LpEMA introduces a certain degree of path history in the IRC route decisions. To compute the actual metric estimate e_i, the IRC combines the previous metric estimate e_{i-1} and the actual metric $M(t_i)$ using an adaptive Exponential Moving Average, as shown in Equation (3.11) [144], where α_i is an adaptive exponential weight, which is calculated using the classical formula for low pass filter, m_i is the gradient between two metric samples (i.e., $\frac{M(t_i)-M(t_{i-1})}{t_i-t_{i-1}}$), and m_{norm} is the normative gradient calculated over a given time window (e.g., 10 times the interval $t_i - t_{i-1}$). After this, the IRC picks the best path, which is the path that has the smallest metric estimate e_i.

$$\begin{cases} e_i = (1-\alpha_i)e_{i-1} + \alpha_i M(t_i) \\ \alpha_i = \alpha_{max}\frac{1}{1+\frac{|m_i|}{m_{norm}}} \end{cases} \tag{3.11}$$

3.6.2.3 Path Switching Policies

Finally, knowing that a path performs better than the active path is not a sufficient condition to pick it because of stability issues. Thus, the RCE component may invoke a path switching policy which only allows a path switch if a certain condition is met. In this study, three policies for path switching are used, which are as below.

CB (Choose Best). According to the CB policy, an IRC switches paths whenever it finds better paths in terms of performance. Thus, the IRC picks the path that has the smallest value of the chosen metric, independently of any quality bound;

CG (Choose Good). According to the CG policy, an IRC switches to a better path if the performance characteristics of the active path do not comply with the traffic requirements. That is, when it becomes infeasible;

CBCG (Choose Best - Choose Good). This policy is a combination of both the policies CB and CG. Similar to the CB policy, the IRC switches paths whenever it finds better performing paths. However, there is an important difference, that is, if the performance of a path does not fit in with these bounds, it is enough to pick any alternative good path (not necessarily the best).

3.6.2.4 Path Switching Algorithm

The piece of pseudo-code that describes the complete path switching process used by our IRC system is outlined in Algorithm 3.1. According to this algorithm, regardless of the adopted path switching triggering mechanism, once the quality measures and traffic properties are updated, the IRC applies the path selection process to the set of feasible paths. More specifically, it calls the ranking function, denoted as χ, that abstracts the IRC path comparison logic, whose objective is to map the whole set of available paths to a set order of preferences and output the top ranked path. In the final steps, if a given switching policy is met, the IRC switch the traffic to the selected path. Eventually, when there is more than one equally good path, the IRC can apply a set of tie-breaking criteria or enable BGP to break the tie.

3.6.3 Shifting traffic over ISPs

IRCs must enforce the path changes produced by the path selection process into the BGP infrastructure. In this section, we provide two techniques to enforce path changes by means of a careful manipulation of BGP `LOCAL-PREFERENCE` and `MED` attributes. A common aspect of these techniques is that when they were devised they only took account of mappings between costs or utilities assigned to paths, and the corresponding BGP attribute values. However, these could be easily adapted to other kinds of path comparison methods.

Local-Preference Tweaking. The `LOCAL-PREFERENCE` attribute makes it possible to indicate the degree of preference for a route, as compared to other available routes for the same prefix (a higher `LOCAL-PREFERENCE` means more preferred). Therefore, if we denote $P = [P_1, P_2, ..., P_k]$, as the vector of paths to a destination, and the corresponding vector of routing costs denoted as $M = [C_1, C_2, ..., C_k]$, the simplest mapping is the linear mapping as in the Equation (3.12), where $Local - pref(P_i)$ denotes the `LOCAL-PREFERENCE` to assign to P_i within a range $[p_{min}, p_{max}]$ and ΔTie denotes a small optional tie-breaking factor that may be added by the IRC to the `LOCAL-PREFERENCE` value to tie-break equally good routes. Similarly, a linear mapping can be defined when an IRC uses a utility function to estimate and compare the quality of paths, although in such a case, the paths with higher utilities are the most preferred.

$$Local - pref(P_i) = p_{min} + \left\lfloor (p_{max} - p_{min}) \left(1 - \frac{C_i}{max\left[C_i\right], \forall i} \right) \right\rfloor + \Delta Tie \qquad (3.12)$$

MED Tweaking. An alternative solution is by means of `MED` tweaking. The mapping between the routing costs and `MED` values is similar to the previous case, except that the preferred routes have lower `MED` values (see Equation (3.13)). In the Equation (3.13), $MED(P_i)$ denotes the `MED` to assign to P_i within a range $[m_{min}, m_{max}]$ and ΔTie denotes a small optional tie-breaking factor that, here, may be subtracted by the IRC to the `MED` value to tie-break equally good routes. Similarly, a linear mapping can be defined when an IRC uses a utility function to estimate and compare the quality of paths, although in this case, paths with higher utilities are the most preferred.

Algoritmo 3.1 IRC($\{P, A, C\}$)

```
Require: $\{P\}$ - vector of the set of $N$ AS paths for a prefix $p$
         $\{A\}$ - matrix of the set of performance/QoS attributes
         $\{C\}$ - vector of criteria representing the traffic goals
Ensure:  $P_x$ - the active path that meets traffic goals toward prefix $p$
for each IRC cycle do
    for $i = 1$ to $N$ do
        Update the performance attributes of path $P_i$
    end for
    /* Basic IRC path selection process */
    Identify the set of feasible paths $P'$
    $[P', A'] \leftarrow \{P_i \in P : \forall k, a_{ik} \leq C_k\}$
    if $\| P' \| \neq 0$ then
        /* Identify the highest rank path $P_i \in P'$*/
        $x' = argmax \chi(\{P', A'\})$ /* the ranking function $\chi$ compares the performance measures or the quality estimate of all feasible paths*/
        if $\| x' \| > 1$ then
            /* If there is more than one path equally good, break the ties*/
            $P'' \leftarrow \{P_i \in P' : i = x'\}$
            $x' \leftarrow TieBreak(P'')$
        end if
        /*Enforce a path switching policy*/
        $s' \leftarrow policy(P_{x'})$, where $s \in \{0, 1\}$ indicates if the new best path $P_{x'}$ should be activated (i.e., True(1), False(0))
        if $s' = 1$ then
            /*if $s'$ signals to use $P_{x'}$ to reach $p$, switch the traffic to $p$ from active path $P_x$ to best path $P_{x'}$*/
            $P_{x'} \leftarrow P_x$, start using $P_{x'}$
        end if
    end if
end for/* End of IRC path selection process */
```

$$MED(P_i) = m_{max} - \left\lceil (m_{max} - m_{min}) \left(\frac{min\,[C_i]\,, \forall i}{C_i} \right) \right\rceil - \Delta Tie \qquad (3.13)$$

We exemplify the use of both techniques below. Let us consider again the paths $(P1, P2, P3)$ from the previous example, and their costs $[C_1, C_2, C_3] = [2.6, 1.7, 1.9]$ computed by means of the metric combination method. *Assigning LocalPrefs.* Let us first assume that the AS administrator assigns non-overlapping ranges of `LOCAL-PREFERENCE` values to IRC and business policies to avoid policy disputes between both mechanisms. For instance, the following `LOCAL-PREFERENCE` ranges: $[p_{min}, p_{max}] = [90, 99]$ for IRCs and $[100, 109]$ for business policies. Thus, the computed $Local - pref(P_i)$ are respectively 90, 93, and 92. This means that, as expected (and desired), $P2$ has the highest `LOCAL-PREFERENCE`. As result, the decision process of the local BGP router will select and install $P2$ into its Loc-RIB.

Assigning MEDs. In a similar way to `LOCAL-PREFERENCE`, let us assume the administrator assigned non-overlapping ranges of `MED` values to IRCs and business policies to avoid policy disputes; for example, `MEDs` in the range $[m_{min}, m_{max}] = [90, 99]$ for IRCs and $[80, 89]$ for business policies. Thus, the computed $MED(P_i)$ are respectively 93, 90, and 91. This means that, as expected (and desired), $P2$ has the lowest MED. As result, the decision process of the local BGP router will select and install $P2$ into its Loc-RIB. However, this step is not deterministic as will be discussed next.

Discussion. The major advantage of an IRC path switching solution based on the manipulation of `LOCAL-PREFERENCE` is that it overrides any other route attribute, since it is the main criterion of the standard BGP decision process. However, this implies that when an IRC uses this attribute to control traffic it may violate local business policies, unless a solution based on two `LOCAL-PREFERENCE` ranges is used, that is, one range for enforcing business policies and the other for traffic engineering purposes or IRC manipulation.

Hence, relying on subsequent attributes to rank paths, such as comparing the `MED` values of paths can be an alternative solution. However, this choice could only be made at the cost of allowing the traffic switches deduced by the IRC system to become very ineffective. For instance, when using the `MED` values, these are only compared if the routes have equal AS-path lengths. In short, when relying on subsequent BGP route attributes, the effectiveness of the used technique must be ensured, for instance, by artificially increasing the length of the AS paths to force the `MED` values to be checked.

In view of this, in our design, we preferred to keep using the `LOCAL-PREFERENCE`-based technique, on the assumption that the IRCs were properly configured to conform to business constraints, i.e., they do not export `LOCAL-PREFERENCE` to paths, whose values cannot be changed by IRCs.

Another important issue is how to break the ties? In fact, the IRC system may obtain multiple paths with the same cost or utility. An alternative straightforward approach is to let the BGP to break the ties. The drawback of this approach is that the comparison criteria used in the BGP decision process have a poor correlation with the traffic goals, and can hamper the improvements to the traffic. Hence, perhaps the best option is to include within the IRCs their own tie-breaking rules or methods, such as the proposed optional tie-breaking factor. This factor can be randomly generated or combined with other methods, for instance, with the lexicographic method that was described earlier. In this later case, the tie-breaking factor is suitably generated, in accordance with the order retrieved by means of the lexicographic comparison.

Finally, it is worth noting that other mapping functions can be considered, such as non-linear mappings. An essential requirement is that the retrieved order of preference between the set of feasible paths is at least of an ordinal type. This means that, although it cannot allow that the number of times a path is better or worse than another to be quantified, there is no doubt about the exact path ordering in respect to their quality.

3.7 Evaluation of the IRC Benefits and Feasibility

In this section, we evaluate the performance benefits and feasibility of employing Intelligent Route Controllers (IRC) against the use of the Border Gateway Protocol (BGP). The results reported here investigate a worse-case scenario, that is the employment of the choose-good path switching policy of IRCs. However, the main conclusions of this study are also applicable to other policies, such as choose-best and choose-best-choose-good. In Appendix B or in book chapter [103] the interested reader can find a study that we carried out to compare IRC against BGP, including the performance assessment of IRC under different path switching policies.

3.7.1 Evaluation Methodology

The simulations were performed using the J-Sim [145] simulator with the BGP Infonet suite [146], which is compliant with the (previous) BGP specification RFC 1771 [147]. All the IRC mechanisms were developed on top of the BGP implementation available in this platform.

Some extensions had to be added to the Infonet suite to allow the IRCs to have full access to the RIBs (Routing Information Base) of a BGP router, and to have control over the BGP decision process. After this, we have also added the following QoS extensions to BGP:

- An optional transitive attribute to distribute the Class of Service (CoS) identification (ID);

- A set of modifications to the BGP tables to allow the storage of this additional information, following a similar approach to the one described in [47, 48];
- A set of mechanisms to allow BGP speakers to load the supported CoSs;
- A set of mechanisms to provide multi-session support to enable BGP speakers to control more than one CoS for the same IP prefix;
- A mechanism to allow each local IP prefix to be announced within a given CoS;
- A mechanism to allow BGP speakers to set the permissibility based on local QoS policies and supported capabilities.

Simulation Scenarios. In this study, both schemes were evaluated depending on whether the Differentiated Services (Diffserv) feature [6] is enabled or disabled, resulting in two pairs of simulation scenarios:

(a) IRCs & Diffserv enabled (ON) / IRCs & Diffserv disabled (OFF);

(b) BGP & Diffserv enabled (ON) / BGP & Diffserv disabled (OFF).

Network Model. Figure 3.11 illustrates a simplified picture of the network model used. This topology represents a typical scenario where the multihomed stubs employing IRCs are AS1-AS4. The simulated network aims at representing a multi-service part of the Internet composed by access ISPs (ISP1-ISP3) able to provide some limited QoS services to their customers (AS1-AS4), and an over-provisioned Internet core (composed by Tier 1 and big Tier 2 ISPs). Thus, core routers from access Internet Service Providers (ISPs) implement the standard Diffserv Per-Hop Behaviors (PHB): Expedited Forwarding (EF), Assured Forwarding (AF) and Best-Effort (BE) [148,149]. On the ingress of these ISPs we used edge Diffserv capabilities to mark packets with a specific DSCP (DiffServ Code Point) depending on their corresponding service. These marks were applied both to regular IP packets, and to the probes generated. To address concerns regarding complexity, we modeled each AS as a single eBGP router, except in the case of access ISPs, due to Diffserv mechanisms that are composed of a pair of routers, an edge router running eBGP (external BGP) and a core router running iBGP (internal BGP). The destination of all the traffic is AS5.

Synthetic Traffic and Simulation Conditions. In this set-up, the tests were conducted using a traffic mix consisting of up to forty Voice over IP (VoIP) calls, up to twenty video calls, prioritized data, and web traffic. New voice and video connections arrived at the border routers of AS1-AS4 and the corresponding durations are uniformly distributed between $[500, 1500]$, and $[180, 300]$ seconds respectively. Both kinds of connection calls are active along all the simulation runtime, which is $[500, 1800]$ seconds. The following are the corresponding marking scheme (when Diffserv feature is enabled) and source models (which are similar to the ones that were employed in [150]):

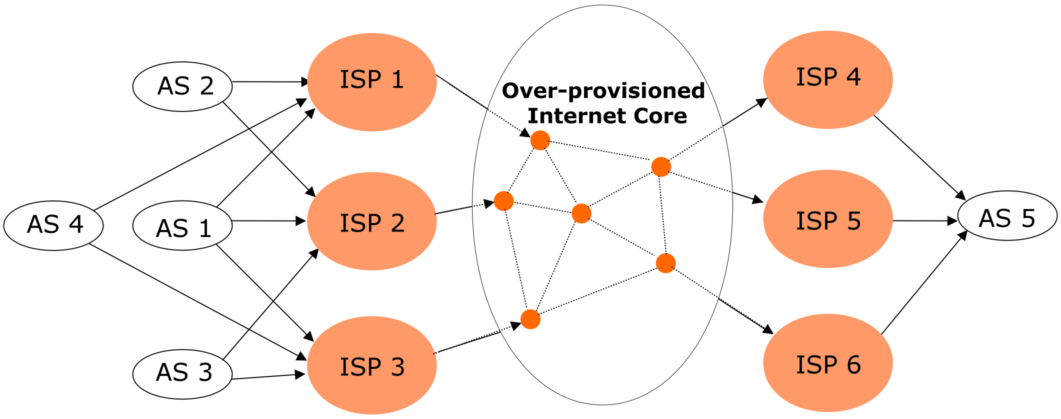


Figure 3.11: Network model.

(i) EF Voice traffic is marked with EF value [149]. EF source is an ON-OFF VoIP generator characterized by Pareto ON-OFF model. The Pareto's shape parameter is 1.9. In the ON state, EF voice source generates traffic at a peak rate of 64 Kbps. All ON-OFF packets have a size equivalent to 174 bytes;

(ii) Video traffic is marked with AF11 value [148]. AF11 source is a video traffic generator characterized by a similar Pareto ON-OFF model. In the ON state, AF11 source generates video traffic at a peak rate of 128 Kbps. All ON-OFF packets have a size equivalent to 576 bytes;

(iii) Finally, the data prioritized traffic is marked with AF21 value [148]. Both AF21 and BE traffic are characterized by a Poisson process (in which the inter-arrival of packets follows an exponential distribution). Both AF21 and BE generate traffic at an average rate of 960 Kbps. The size of the packets generated by the data connections in these simulations is 1112 bytes.

The simulated time was always 1800 seconds of which the first 500 seconds were discarded as "warm-up period" due to BGP convergence. This is the time needed for BGP routers to advertise all the network reachability information so that the paths become available to IRCs at ingress ASs. It is worth to noting that, in the last 300 seconds of simulation, the network becomes increasingly underutilized, because it only remains in the network the last flows of traffic that have arrived until the second 1500.

During all the experiments, it is assumed that a Service-Level Specification (SLS) was previously exchanged between remote multi-homed stub domains, based on a maximum One-Way Delay (OWD) for each service. These maximum OWDs tolerated per-service were chosen to represent reasonable but demanding values for the kinds of traffic sources considered. Thus, the OWD bound for voice and video traffic was set at 150ms, and at 400ms for prioritized data and web traffic, which are in accordance with the E-model Rating from ITU's G.107/G.114 recommendations [108, 109].

Finally, to examine what occurs for different network provisioning conditions, the "bottleneck" links have a capacity C of 3583 Kbps or 4095 Kbps to roughly provide an aggregate capacity of all available paths to the total offered load of 105% or 120% respectively. All the links have a propagation delay of $Pd = 3$ms (900 km/s).

3.7.2 Objectives and Performance Metrics

The main objective of this study is to assess how much IRCs aid is needed to improve end-to-end network performance under variable traffic dynamics and capacities. This is shown by evaluating:

(i) The latency measured for each traffic, denoted as d, against the SLS constraints for the different traffic flows, denoted as D_{max};

(ii) The total number of path switches needed to meet the latency constraint for each kind of traffic;

(iii) And, the traffic transfer efficiency for the different traffic flows, denoted as Ef. The efficiency for a traffic class is defined by $Ef = \frac{F_D}{C_O}$, where F_D is the total throughput received at destinations D, and C_O is the corresponding total throughput sent by source domains O;

In addition, and to aid the assessment of both schemes, we quantify and compare the overall effectiveness of each. In other words, we aim to show at once the ability of IRCs to exploit the available paths and thus reduce the latency of traffic, including its impact on traffic transfer efficiency. In this study, the overall rate of efficiency is supplied by a simple Effectiveness index, as shown in Equation (3.14), which combines the relative improvement in the latency and the traffic transfer efficiency for each scheme. Lower values for this index indicate better overall effectiveness.

$$\text{Effectiveness index} = \left(\frac{d}{D_{max}}\right)\left(\frac{100}{Ef}\right) \tag{3.14}$$

3.7.3 Results

We make a comparison of the behavior of both the BGP and IRC schemes for the same SLS constraints and two network provisioning conditions. The first results concern the latency measured for both schemes. The Figures in 3.12 illustrate the Complementary Cumulative Distribution Functions (CCDF) of the traffic latencies for the relative traffic loads considered in this study. If the probability of the OWD is greater than or equal to x is high (i.e., $P(OWD \geq x)$ is high), it means that there is a strong likelihood that the traffic will suffer a latency greater than or equal to x. In turn, Figures 3.13(a)(1) and 3.13(b)(1) show the averaged latencies for each traffic class for the two network provisioning conditions.

Two main conclusions can be drawn from the analysis of the results of the latency. First, the IRC model substantially enhances end-to-end performance when compared with a pure BGP model. If the CCDF curves of Figures 3.12(a) and 3.12(b) are contrasted, we can clearly observe that when employing an IRC strategy the likelihood of a violation of the maximum OWD bounds decreases significantly. There are, however, two points concerning this issue that we should highlight. The first point is that IRC clearly can protect the most important traffic classes (with more stringent QoS requirements), since voice and video traffic experience a lower likelihood (up to 35% for low capacity and up to 10% for high capacity) to undergo a latency higher the tolerated bound than the prioritized data and web traffic. The second point is that when comparing the overall performance of IRC with that of BGP for all combinations of capacity and state of the Diffserv feature, we conclude that the IRC model is more effective if the network is slightly over-provisioned (in our case 120% cap.), and cannot avoid congestion if there is not enough capacity to carry the traffic (in our case 105% cap.). On the other hand, the CCDF curves show that when Diffserv feature and IRCs are disabled, the pure BGP model may not be able to satisfy the ITU's bounds under stressful traffic load. More precisely, Figures 3.12(a)(4) and 3.13(b)(4) show that the BGP-based scheme has a likelihood of violating all OWD bounds equal or greater than 90%.

Second, Diffserv clearly shows its effectiveness in being able to protect the most important traffic classes, since voice, video and prioritized data traffic have OWDs within the ranges allowed. Although this may sound surprising, it is worth noting that the good performance of Diffserv is obtained at the expense of a strong efficiency penalization over the web traffic, as well as some penalization over the prioritized data. Figure 3.13(a)(2) clearly shows this, especially in the case of BGP. In fact, as can be observed, more than of 50% of the web traffic is dropped.

We will now conduct a general discussion about the stability issue, analyzing the number of path switches that have been performed by IRCs to meet OWD bounds (see Figures 3.13(a)(3) and 3.13(b)(3)). As expected, the IRCs perform a larger number of path switches when the Diffserv feature is disabled. This is especially evident for the scarce network capacity and for the most important classes of traffic. By correlating both the results – the traffic latency that is experienced and the number of path switches, we can also observe that in the case of the less important traffic classes, the performance of IRCs is apparently limited by a lower path diversity. Figures 3.13(a)(1) clearly show that these traffic classes can experience unacceptable latencies, higher than 500ms, whereas the IRCs perform a smaller number of path switches.

Finally, we compare the overall effectiveness of both the schemes. Figures 3.13(a)(4) and 3.13(b)(4) show the results for the overall effectiveness. As would be expected, BGP displays the weaker overall effectiveness. This observation is consistent with the traffic latency and efficiency results. The lack of a BGP response to congestion translates into a strong penalty, especially when total capacity is barely enough to carry the aggregate traffic load and for the most important traffic classes (where the effectiveness index displays the biggest values). On the other hand, the IRC is the most effective scheme, at least in absolute terms. These results show that just a small number of path switches is enough to improve the performance of traffic dramatically in terms of latency and efficiency. This is especially evident when the network is slightly over-provisioned (where the effectiveness index displays the smallest values).

In summary, the simulations of this section show the feasibility and potential performance benefits of the IRC model. The results showed that this strategy outperforms the BGP-model in terms of all evaluated performance metrics – traffic latency and efficiency – and overall performance index. Moreover, we found that the IRC model is more advantageous when the network is slightly over-provisioned. This implies that the IRC model is not suitable for handling extreme network congestion. It should be also referred that IRCs and Diffserv features can achieve a synergistic co-operation in improving end-to-end traffic performance.

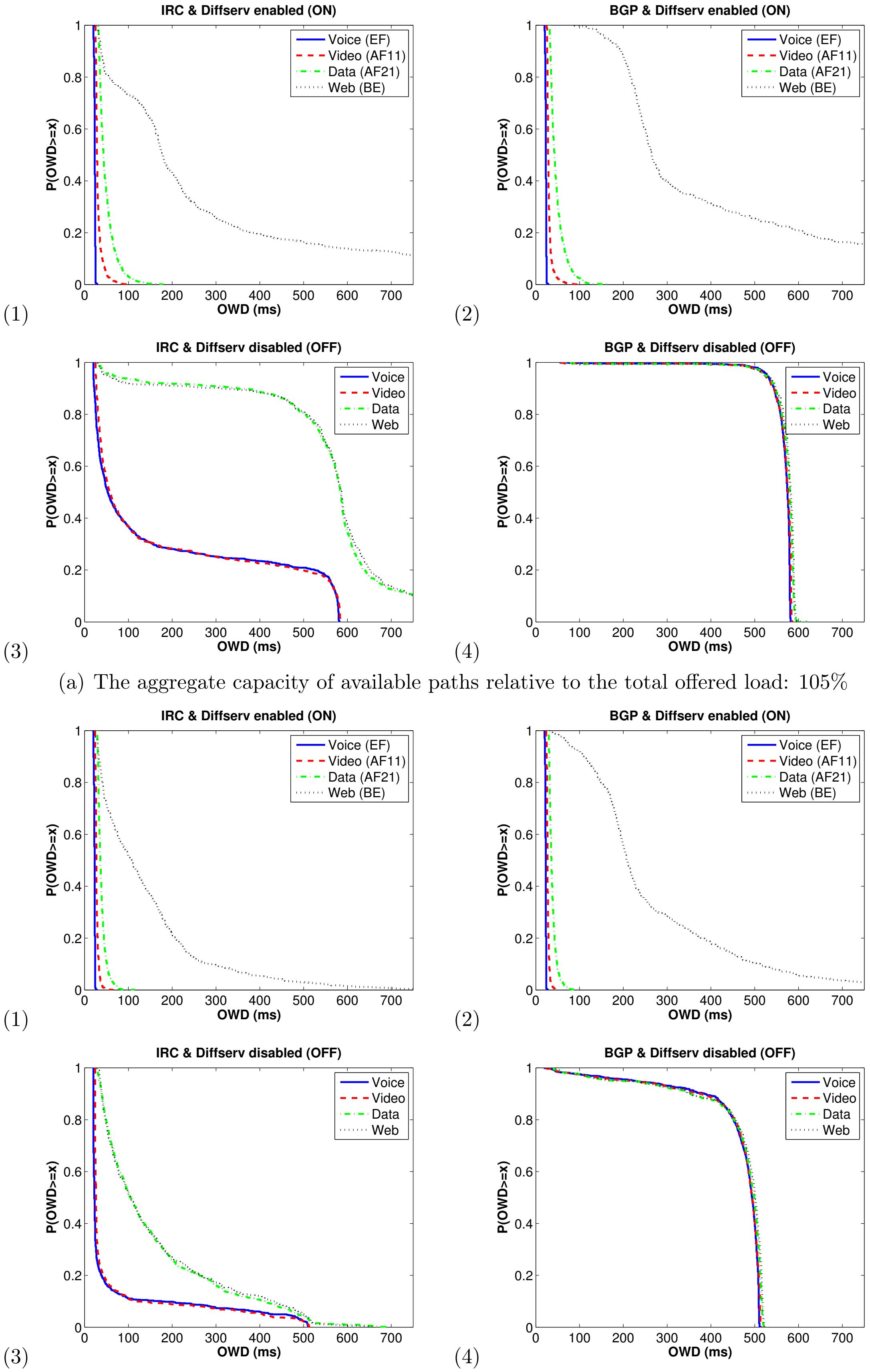
3.8 Evaluation of the Mechanisms for Request Timeout Adaptation, and Response to Path Failures

This section contains the description of the simulation model and the results obtained with the design alternatives of mechanisms for RTO (Request Timeout) adaptation outlined in Section 3.5.2. Furthermore, we provide the response times of IRCs to path failures against the corresponding times obtained by BGP.

3.8.1 Simulation Model and Performance Metrics

We evaluated the performance of IRCs endowed with the Jacobson-like and Box-plot rule-like algorithms for RTO estimation by re-using most of the simulation model described in the previous section. To simplify the descriptions, only the results for the web traffic are given. However, the main conclusions are valid for the other traffic models.

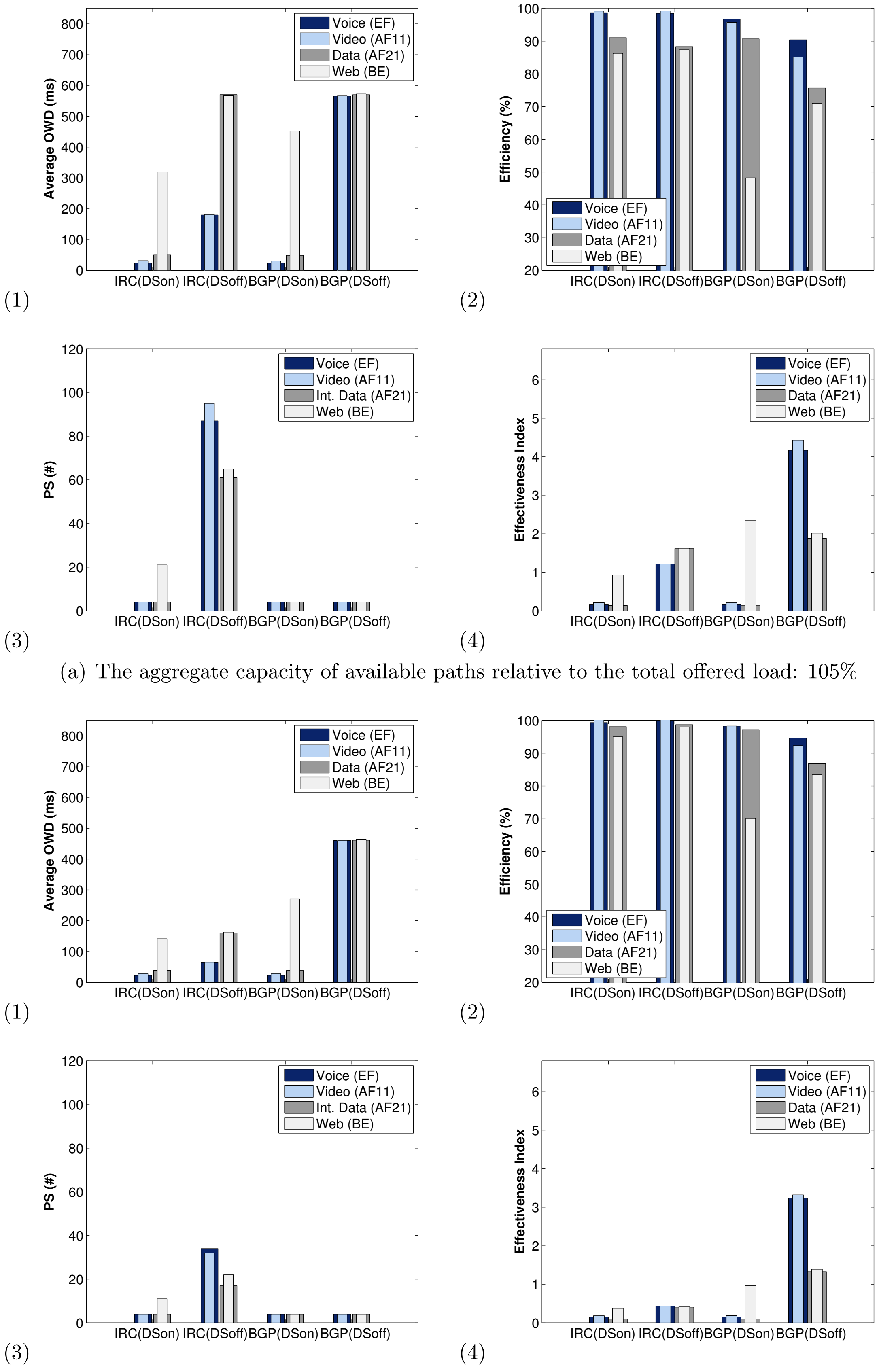
The tests were thus conducted using a Poisson-like traffic model, in which the inter-arrival of packets follows an exponential distribution. In contrast to the previous set-up, we consider the costs of paths as the median of the measured RTTs (Round-Trip Times) along a path during a routing cycle, instead of OWDs. The maximum RTT tolerated for traffic was set to 2x300ms, i.e., 600ms, which is in accordance with the E-model Rating from ITU's G.107/G.114 recommendations for good quality [108,109].



(a) The aggregate capacity of available paths relative to the total offered load: 105%

(b) The aggregate capacity of available paths relative to the total offered load: 120%

Figure 3.12: Complementary Cumulative Distribution Function (CCDF) of OWDs for each traffic whether (left) Diffserv feature is enabled or (right) Diffserv feature is disabled.



(a) The aggregate capacity of available paths relative to the total offered load: 105%

(b) The aggregate capacity of available paths relative to the total offered load: 120%

Figure 3.13: Average OWD, transfer efficiency, total number of path switches, and overall effectiveness index for each traffic whether Diffserv feature is enabled or disabled.

The sampling window T_w was configured to 30 seconds and the number of probes N_w to 6. After convergence of BGP, each simulation runs during a period of 1300 seconds.

We use five performance metrics: the latency of the probes (RTTs) and the corresponding request timeouts (RTOs), the number of spurious timeouts registered, the number of path switches needed to meet performance goals and the efficiency of the traffic transfers.

3.8.2 Results

The simulation results are shown in two groups. The first group depicts the results relative to the overall behavior of the proposed mechanisms. The second group provides as comparison of the mechanisms in more detail.

3.8.2.1 Overall behavior of Mechanisms for Request Timeout Adaptation

Figure 3.14 displays snapshots of the series of the measured RTTs and the estimated RTOs by each mechanism for a stream of probes sent through a randomly-chosen path under the control of an IRC.

Two important observations can be drawn from Figure 3.14. First, it shows that the Jacobson-based mechanism can effectively estimate RTOs and, therefore, results in a relatively small number of spurious timeouts. In fact, RTTs are always virtually smaller than the corresponding RTOs. Regarding its extended version, the series of RTOs shows that the improvement is negligible when compared to the original.

Second, the original Box-plot rule with $K = 1.5$ has a great potential risk of running into spurious timeout expirations. Indeed, we can observe that both the RTTs and RTOs series are close together and there is no margin for extra waiting time to declare that a probe is lost. The extended version of the Box-plot rule algorithm with $K = 1.5$ has the ability to deal with spurious timeouts, but with a dramatic increase of the extra waiting time to declare that a probe is lost. This occurs because the K constant increases significantly each time a spurious timeout is detected (see Equation (3.8)). In contrast, the version of the Box-plot rule with $K = 3.0$ performs quite well, since there is also a long enough extra waiting time.

In short, Jacobson's algorithm still performs quite well when employed in IRCs, while the original setting of the Box-plot rule technique performs rather badly. We also found that both extended versions perform well and outperform the original versions. However, the extended Jacobson is the best candidate for RTO adaptation due to its simplicity and that fact that it needs no training in advance.

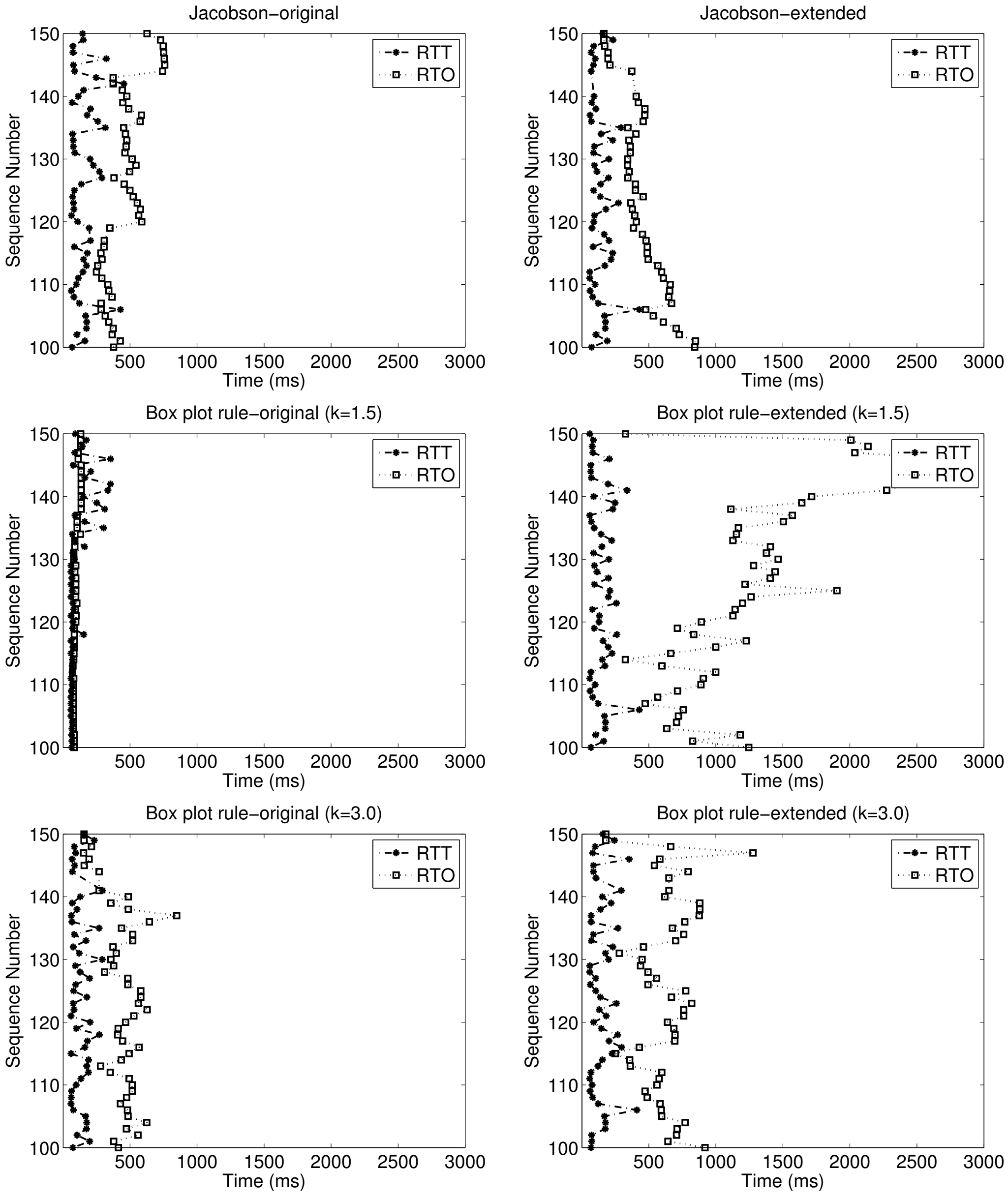


Figure 3.14: Measured RTTs and the estimated RTOs by each mechanism for a stream of probes.

3.8.2.2 Detailed behavior of Mechanisms for RTO Adaptation

Figure 3.15 shows the registered number of spurious timeouts (the graphic at the top) and path switches (graphic at the bottom) for all the settings. In absolute terms, the results on the top graph confirm the above-mentioned trends. The extended versions are more effective than the original ones, as they result in a smaller number of spurious timeouts, and especially the Jacobson algorithm that needs minimum tuning. On the other hand, when the performance of both Box-plot rule versions is contrasted, it is apparent that there is a need for a fine tuning. Furthermore, the large number of spurious timeouts for the original Box-plot rule ($K = 1.5$) confirms the previous observation that there is no safe margin between the RTO and RTT series, which increases the probability of spurious timeouts.

Second, the original Box-plot rule based mechanism ($K = 1.5$) displays the lowest number of path switches, but with a significant penalty over the traffic transfer efficiency. In reality this result is counter-intuitive. Instead of finding extra IRC oscillations as would be expected, the large number of spurious timeouts translates into a dramatic reduction of available paths. Consequently, the poorer ability to adapt path selection to performance changes translates into the weakest efficiency of the traffic transfers, ≈ 82%. In contrast, all the other design alternatives achieved an efficiency of ≈ 93% (see Table 3.1)

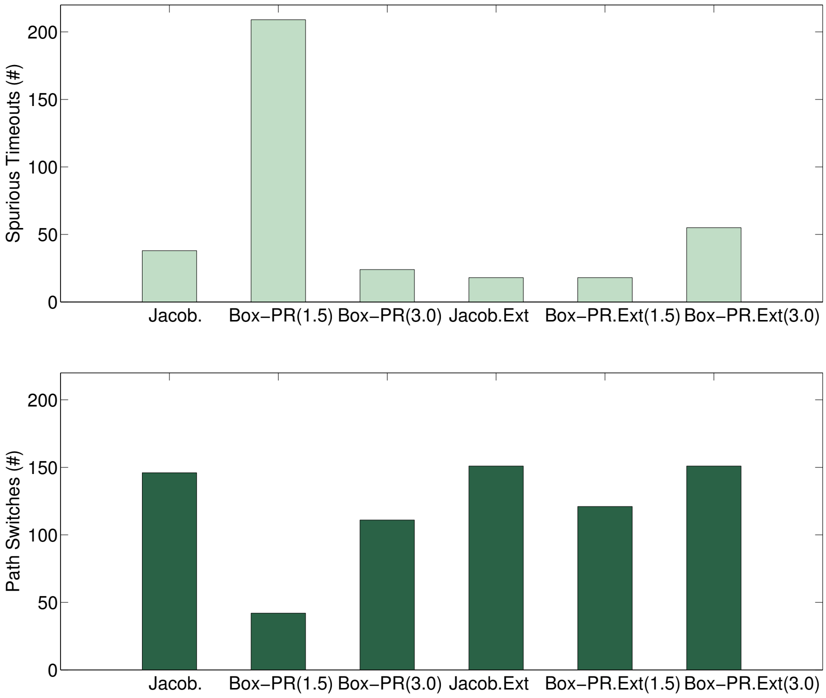


Figure 3.15: Number of spurious timeouts and path switches registered.

Figure 3.16 shows the box plots for the measured RTTs and corresponding RTOs.

Table 3.1: The efficiency of the traffic transfers (%) for the RTO mechanisms.

Jacob.	Box-PR(1.5)	Box-PR(3.0)	Jacob.Ext	Box-PR.Ext(1.5)	Box-PR.Ext(3.0)
93.18	81.76	93.15	93.06	93.26	92.98

These results also reveal the previous trends. The box plots for RTOs clearly show the two facets of the Box-plot rule-based algorithm with $K = 1.5$. On the one hand, the original version displays most of the RTOs distributed over a small range that overlaps the RTT range, and thus leads to a large number of spurious timeouts.

On the other hand, the extended version yields high RTO values distributed over a large range, well beyond the RTTs range which is due to the frequent adaptations of K to handle spurious timeouts, and thus leads to a potential risk of a slow IRC reaction. Moreover, the other design alternatives show RTOs distributed over small ranges beyond the RTT ranges, but close, as desired, to keep the IRCs with a fast enough reaction to failures. The RTTs for the various design alternatives are quite similar, with the exception of the case of the original Box-plot rule with $K = 1.5$, that apparently performs best. In reality, this latter result is biased as this box-plot was obtained with at least less 150 RTTs samples due to the high rate of spurious timeouts (see bottom graph of 3.15).

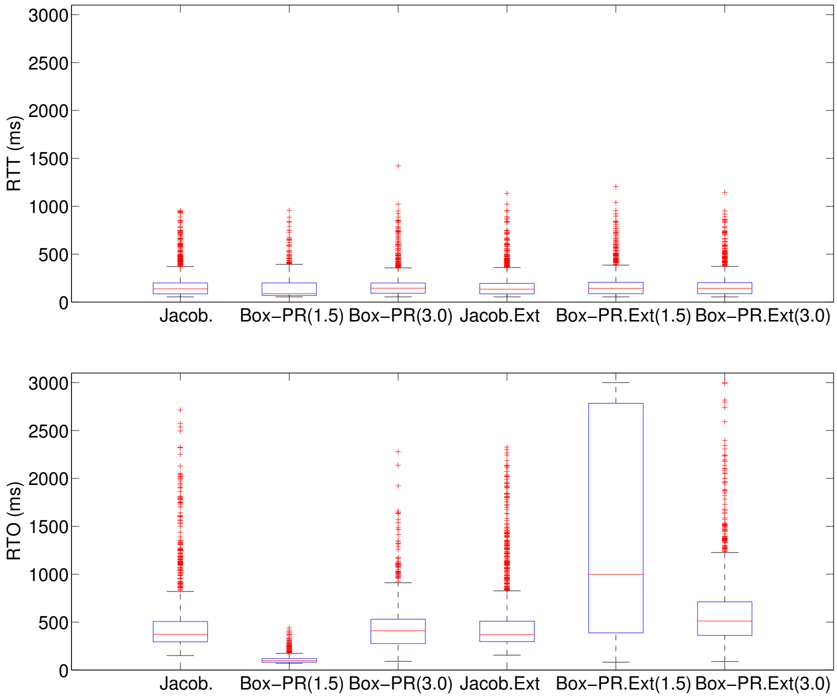


Figure 3.16: Box-plots for RTTs and RTOs.

3.8.2.3 Response to path failures

This section investigates how the IRCs are able to manage and react to a remote link failure. Figure 3.17 contrasts the results obtained for BGP, Jacobson and Box-plot rule based algorithms (extended versions). We have aggressively configured small values for keepalive and hold timers (30 and 90 seconds respectively) to increase the reaction speed of BGP.

From Figure 3.17 it can be observed that in case of link failures occurring a few hops away from a stub AS, the IRCs are able to react, on average, between 9-12 times faster than the time that BGP needs to converge to a new route. In addition, since the IRCs gather end-to-end measurements, their responsiveness becomes independent of the place where the remote link failure occurs. In contrast, the BGP convergence depends on the distance to the failure.

Moreover, Jacobson and Box-plot rule based algorithms provide reaction times (respectively 11212 and 12129 milliseconds), close to one half sampling windows size, i.e., $\frac{T_w}{2}$, (recall that T_w was configured to 30 seconds and N_w to 6 probes), which is in accordance with the theoretical estimation in Equation (3.3). However, the Jacobson version reacts to failures faster than the Box-plot rule-based version (917 milliseconds better on average).

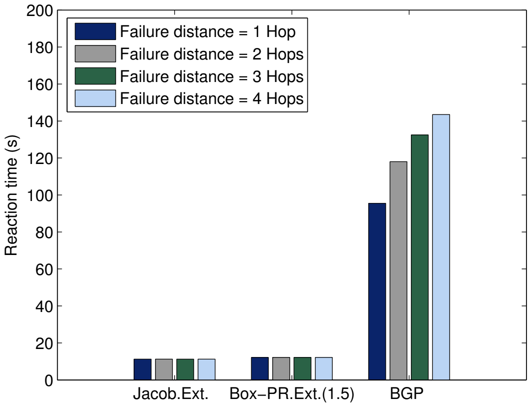


Figure 3.17: Contrasting the response time to a link failure.

3.9 Stabilizing IRC: Randomized Path Monitoring, Randomized Path Switching or History-Aware Path Switching?

The use of the IRC paradigm has a major weakness, that is, oscillations can take place due to factors such as the intrinsic selfish nature of the IRC boxes, self-load effects, and synchronization between the probes sent to target destinations [89, 90].

To illustrate this weakness, Figure 3.18[2] shows the case of oscillations caused by self-loading. In this figure, we can observe that there are periods of intense oscillations because path switching decisions do not take into account the extra load on the new path after switching.

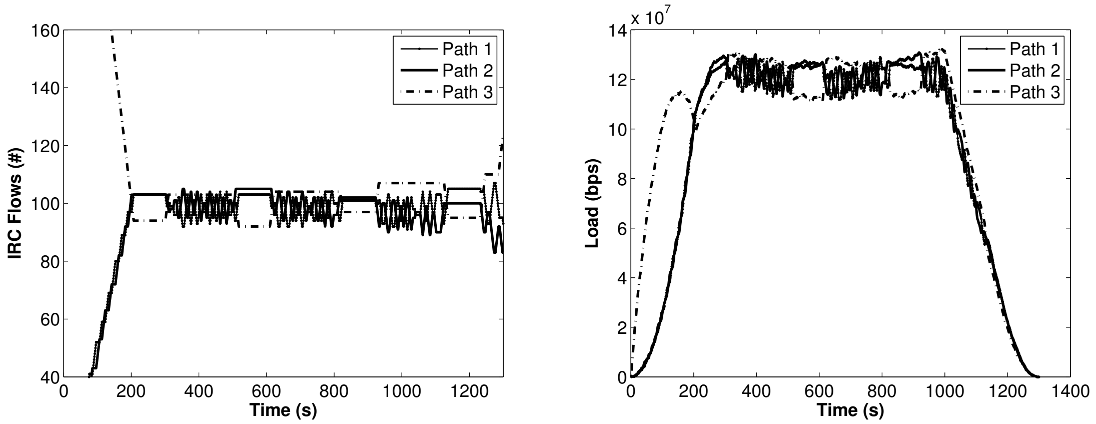


Figure 3.18: Persistent oscillations due to self-load effect.

The cases of self-loading and synchronization between the probes were recently studied in [89]. As a solution, the authors proposed a set of IRC techniques that might reduce IRC oscillations by adding randomness to the path switching process.

In this section, we extend previously-mentioned study in an important direction, that is we seek to answer the question *whether the addition of randomness in the sampling process or the use of recent path history to assist IRC decisions, are also credible alternatives to restrict IRC oscillations?* In short, this section seeks to determine which would be the most promising approach to deal with the IRC oscillations. To achieve this goal, the performance of these three classes of IRC algorithms is investigated in Subsection 3.9.1. The main conclusions of this study are later summarized in Section 3.9.1.4.

3.9.1 Evaluation of IRC Algorithms

This section examines the performance evaluation of DLV-based vs LpEMA-based algorithms for IRC. For comparative purposes, we also include a FSP (Fixed Switching Probability) algorithm because of its ability to remove IRC oscillations, as reported

[2] *This illustration was obtained from a MATLAB-based simulation of the switching of 300 traffic aggregates (each composed of a mix of 80 VoIP flows with Poissonian arrivals) exchanged between two remote triple-homed ASs employing IRC. A choose-best path switching policy was used based on OWD. Three non-overlapped paths with a length of 4 AS-nodes connected both ASs. The link latencies were estimated by means of the M/M/1 queuing model.*

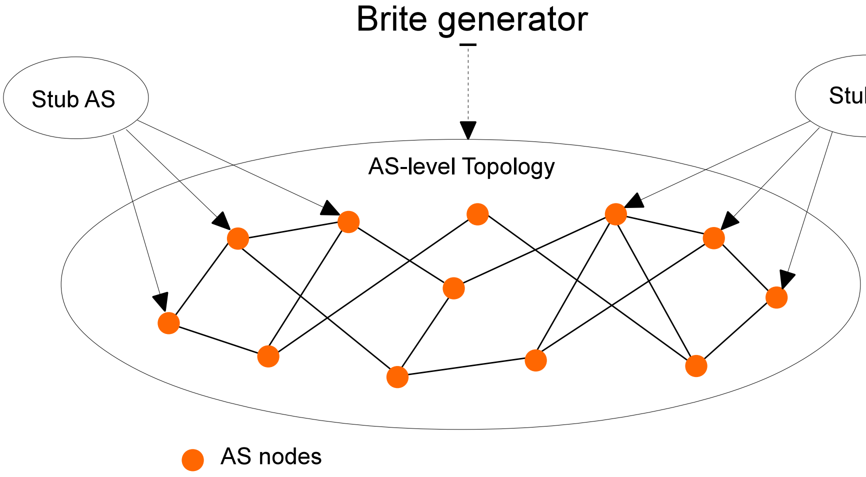


Figure 3.19: AS-level topology.

in [89]. FSP adds randomness to the path switching process. According to the FSP algorithm, the IRC picks the best path with a given switching probability $P \in [0, 1]$.

The evaluation was performed on a J-sim simulation model for IRCs [145]. Two sampling processes were used to compare the three algorithms, namely, a periodic sampling process (Periodic sampling) and a pseudo-Poisson sampling process (p-Poisson sampling), with N_i samples uniformly distributed over a slot of time t, afterwards referred as the window t.

3.9.1.1 Simulation Setup and IRC parameterization

The network topology was built with the aid of the Boston University Representative Internet Topology gEnerator (BRITE) [151]. It is composed of 100 ASs and was generated with a ratio of ASs to inter-domain links of 1:3. During the tests, 300 IRCs sources send homogeneous traffic aggregates to remote prefixes. Each traffic aggregate is composed of a fixed number of multiplexed Pareto flows – to simulate VoIP flows – with Poisson arrivals. The RTT bound for DLV, FSP and LpEMA is set to 300ms.

In this study, we first study the performance of FSP, DLV and LpEMA algorithms in detail. Then, we compare the performance of the best tunings found for these algorithms. The performance of FSP is studied under different values of the switching probability P, varying P between 0.1 and 0.9 (making increments of 0.2). DLV is understood to be a special case of FSP making $P = 1$. On the other hand, the LpEMA configuration relies on α_{max}, typically ranging from 0.5 to 5. However, to study LpEMA performance in the IRC field, it was enough to vary α_{max} in a range of $[0.5, 3.0]$ (making increments of 0.5).

To avoid the self-load effect [89], we combined latency and spare bandwidth into a single metric, i.e., $M = \alpha_1.latency_t + \alpha_2.\frac{1}{abw_t}$, where $latency_t$ is the median of the measured RTTs in a window t of $30s$, and abw_t is the estimated spare bandwidth in the peering link during t. To facilitate the tuning of $\alpha_i, i = 1, 2$, we adopted the framework in [152].

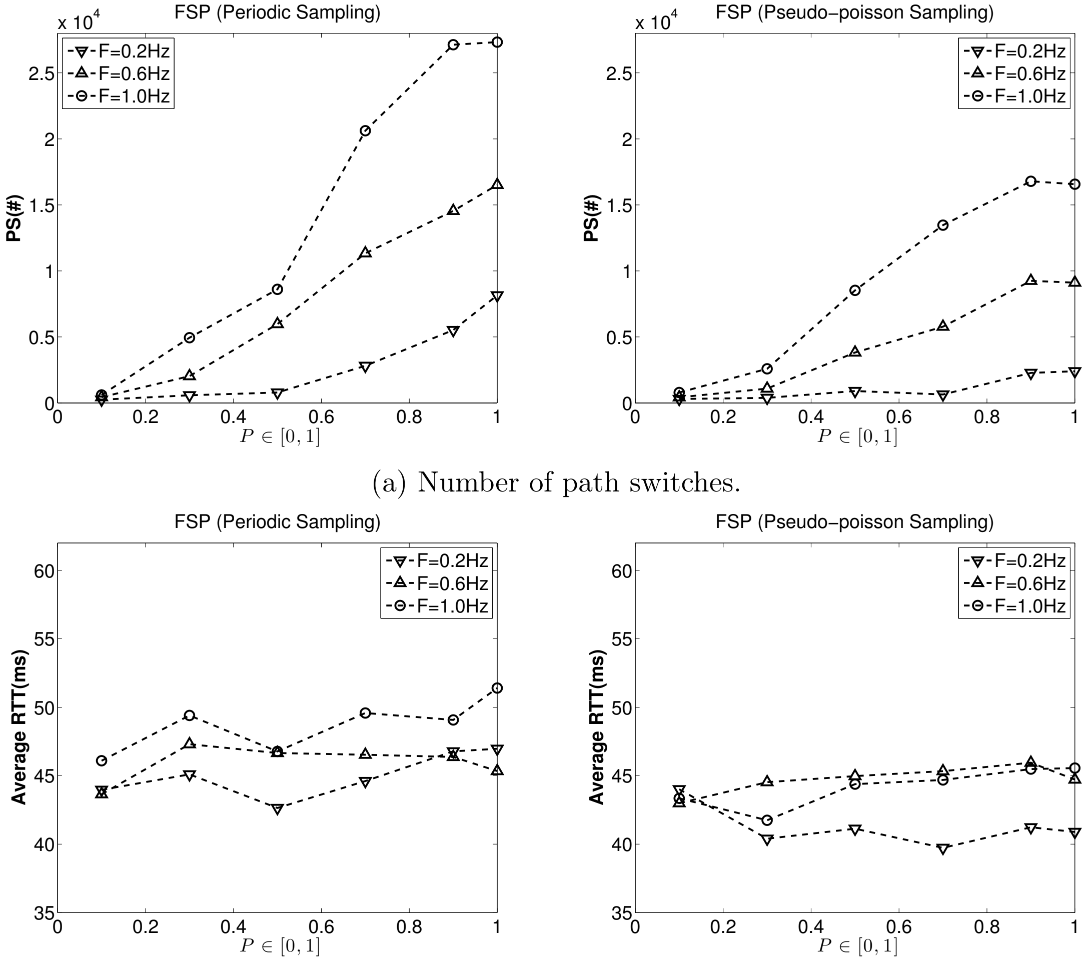
3.9.1.2 Evaluation of the performance of FSP, DLV and LpEMA algorithms

The Figures in 3.20 illustrate the performance of FSP as a function of the path switching probability P. Implicitly, these figures also include the results for DLV, since this algorithm corresponds to the FSP algorithm when making $P = 1$. The results show that for a high P (i.e., $P > 5$), FSP performs poorly in terms of the number of path switches. In effect, when incrementing P, its behavior approaches the DLV's behavior. In contrast, FSP performs better when P is low. In more specific terms, for a range of $P \in [0, 0.5]$, the results show that IRC switches paths more rarely and has almost no impact on the measured traffic performance (see bottom Figures of 3.20).

The results shown in Figures 3.20 give rise to four other important observations. First, when contrasting the performance of FSP for periodic vs p-Poisson sampling, it can be observed that, as expected, the periodic sampling is a potential factor of instability. This trend shows the phenomena of synchronization between probes and/or between routing decisions of competing IRCs. Second, surprisingly, periodic sampling (especially for values of $P > 0.5$) results in roughly double the number of path switches than the p-Poisson sampling process for the analyzed sampling frequencies ($F = (0.2, 0.6, 1)Hz$). This finding shows that when DLV is combined with p-Poisson sampling, it can be considered as a first step towards tackling the problem of the IRC oscillations. Third, as a result, when using p-Poisson sampling, on average, the RTTs measured for traffic are roughly 10% better than the ones measured for periodic sampling ($\approx 5ms$). Lastly these results show that as long the sampling frequency increases, the number of path switches also rises, naturally because of the higher number of path switching decisions.

With regard to the LpEMA algorithm, the Figures in 3.21 show its performance as a function of the parameter α_{max}, which determines the ability of LpEMA to filter sudden variations on the cost metric of a path. These results indicate that the use of a sophisticated IRC algorithm, such as LpEMA, should be carefully planned because may not bring about striking benefits in terms of the reduction of oscillations. Instead, these algorithms might be a potential source of extra IRC oscillations, as described below.

In theory, LpEMA can be viewed as a cascaded DLV and a *low-pass filter* block, and so it should reduce the number of path switches. However, its ideal tuning depends on the particular stability pattern of the network [144]. In our case, we can observe this issue in the results. On the one hand, the number of path switches decays drastically



(a) Number of path switches.

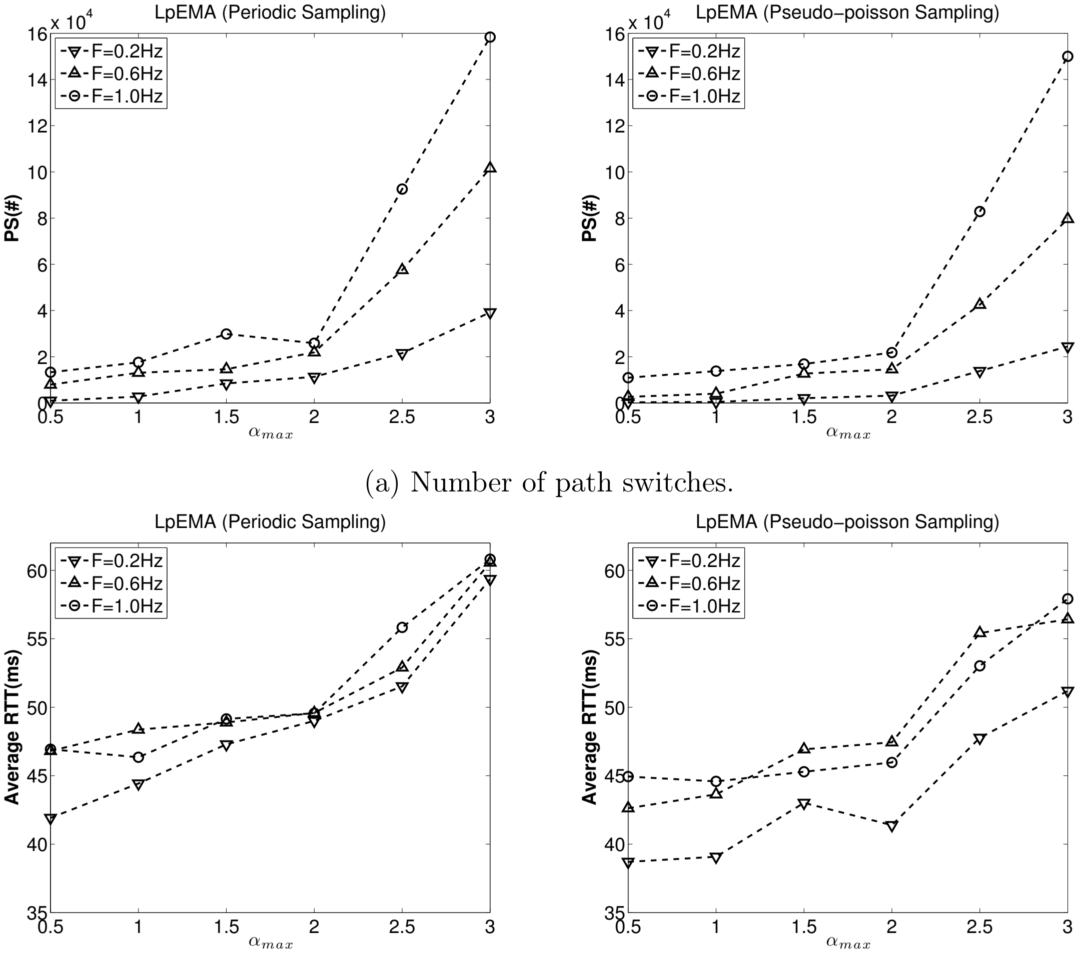
(b) Average of round-trip times.

Figure 3.20: Path switches and traffic latency for FSP as a function of the path switching probability P, for periodic vs pseudo-Poisson sampling.

for low values of α_{max} in a range between 0.5 and 2.0. On the other hand, this number rises drastically for $\alpha_{max} > 2.0$.

Moreover, it is hard to correlate the proper tuning for α_{max} and the resulting number of path switches. When using LpEMA for traffic volume prediction good values for α_{max} are typically around $2.0 - 2.5$, while ensuring small prediction errors during the adaptation process. In contrast, when using LpEMA for prediction of the IRC cost metric and reducing path switches, a good value for α_{max} is apparently around $0.5 - 1.0$, while ensuring the best traffic performance in terms of RTTs.

Nevertheless, the bottom Figures of 3.21 show that the adaptive feature of the LpEMA technique can improve the traffic performance (actually LpEMA can perform better than FSP, and DLV). Finally, these results clearly show that the average of the RTTs is in inverse proportion to the number of path switches performed.



(a) Number of path switches.

(b) Average of round-trip times.

Figure 3.21: Path switches and traffic latency for LpEMA as a function of α_{max} for periodic vs pseudo-Poisson sampling.

3.9.1.3 Comparison of IRC algorithms

In this section, we compare the three IRC algorithms (FSP and DLV, LpEMA) using their best tuning on the basis of previous findings. Accordingly, we set $P = 0.3$ for FSP and $\alpha_{max} = 0.5$ for LpEMA, respectively.

The Figures in 3.22 show the overall comparison of the IRC algorithms, where a comparison is made between the number of path switches performed, subject to the same RTT bound of 300ms, for both periodic and p-Poisson sampling. Three main conclusions can be drawn from the analysis of these results. First, the comparison between the FSP and DLV algorithms shows that IRC oscillations can be reduced by adding a certain degree of randomness in the route control decisions. In fact, these results show a similar pattern to that found in previous study [89]. However, here the IRCs use the RTT performance metric rather than the loss rate metric.

Second, the results also show that the oscillations can be reduced by adding a certain degree of randomness to the sampling process (as observed in the previous section). For instance, Figures in 3.22, illustrate that, for lower sampling frequencies, DLV combined with p-Poisson sampling performs better than when combined with periodic sampling, for instance, when $f = 1.0Hz$, DLV combined with p-Poisson sampling requires almost half of the number of path switches needed for periodic sampling.

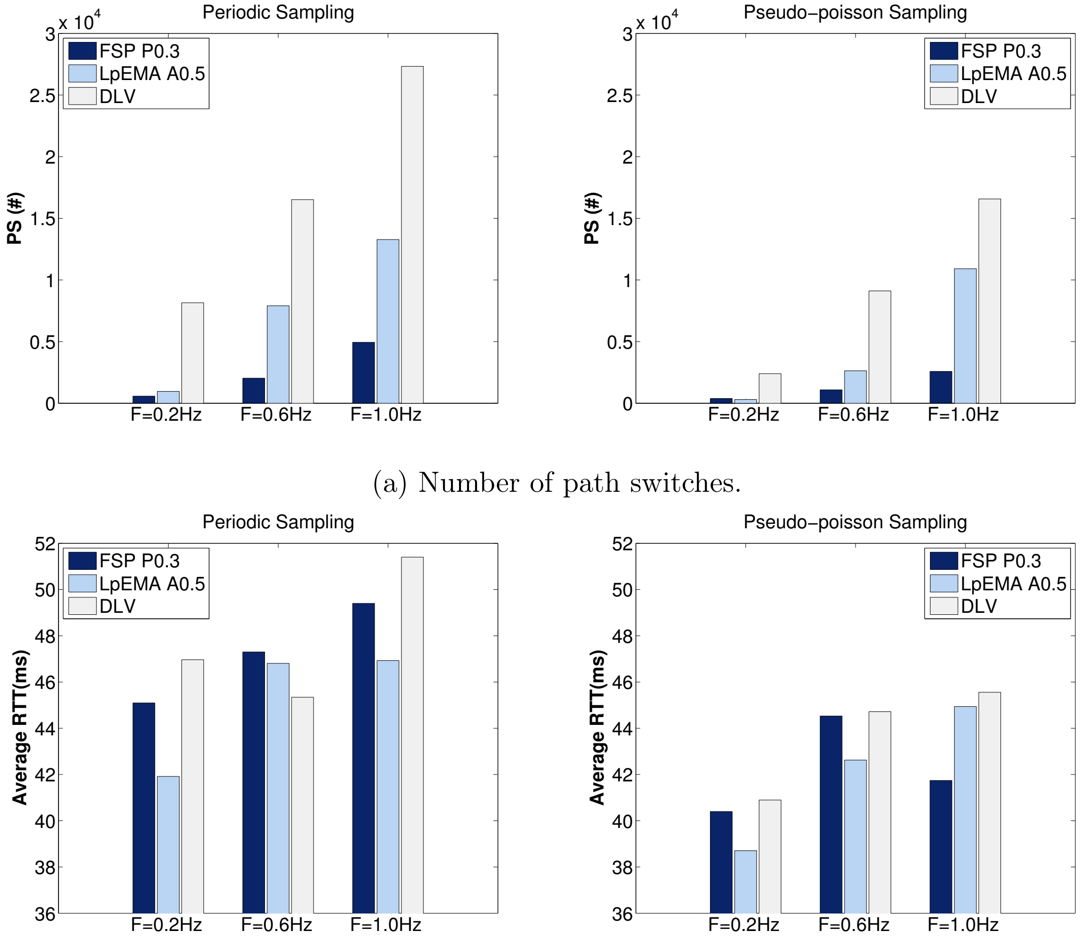
Third, the results in Figures 3.22 show that LpEMA has the potential to improve IRC stability, while ensuring good traffic performance. LpEMA can outperform DLV both in terms of the number of path switches and traffic performance, as well as FSP, but only in terms of traffic performance at the expense of a higher number of path switches. This result is, however, significant because in its essence LpEMA is still a deterministic path switching similar to DLV, although it has an adaptive smoothing feature.

In short, a good tuning for LpEMA has the potential to provide the best trade-off between stability vs traffic performance. However, as discussed earlier, since LpEMA requires careful tuning, it may not yield benefits in terms of the reduction of oscillations.

Finally, to give the interested reader a better understanding of the stability of these algorithms, Figures 3.23 to 3.24 illustrate two empirical Complementary Cumulative Distribution Functions (CCDF) of the number of path switches performed by the three IRC schemes for two bin sizes, used to count the path switches along the simulation. If the probability of a number of Path switches (PS), in a bin, is greater than or equal to x is high (i.e., P(PS in a bin$\geq x$) is high), it means that route oscillations are highly present in every bin. It must be pointed out that a line starting at a value smaller than 1, means that a fraction of the bins do not have any path switches.

Main observations discussed previously are, thus, shown in Figures 3.23 to 3.24 with a finer granularity. It is evident that the probability of path switches in a bin is a function of the used algorithm, the type of the sampling process and the sampling frequency. For instance, when $f = 1.0Hz$ and periodic sampling is used, DLV has a probability of about 50% to perform more than 100 path switches in a bin of 3s. In turn, in the same conditions, FSP has almost a zero probability. In a second example, when using DLV combined with p-Poisson sampling, this has a probability of approx. 25% to perform more than 10 path switches in a bin of 3s. But, when this algorithm is combined with periodic sampling, it has a probability of about 60% to perform more than 10 path switches and about 25% to perform more than 50 path switches. Naturally, this advantage of the p-Poisson sampling to reduce IRC oscillations is less for a higher sampling frequency ($f = 1.0Hz$), where it is only observed that there is an effective reduction in the probability of experiencing a given number path switches for analysis over 30-40 path switches per bin.

We also found that LpEMA with less advantage has the potential to reduce the probability of a given number of path switches during the window of the size of a bin.



(a) Number of path switches.

(b) Average of round-trip times.

Figure 3.22: Comparison of FSP, LpEMA and DLV algorithms – path switches and traffic latency for different sampling frequencies.

There is, however, an interesting aspect of the behavior of the LpEMA performance that can be explained by its adaptive smoothing feature. For random sampling and using low/mid range frequencies, it performs best and better than FSP. On the other hand, its adaptive feature causes IRCs to switch more often when using higher sampling frequencies to improve the traffic performance. This aspect contrasts with FSP, whose behavior is in some respect blind to traffic performance, while it is restricting the number of IRC oscillations.

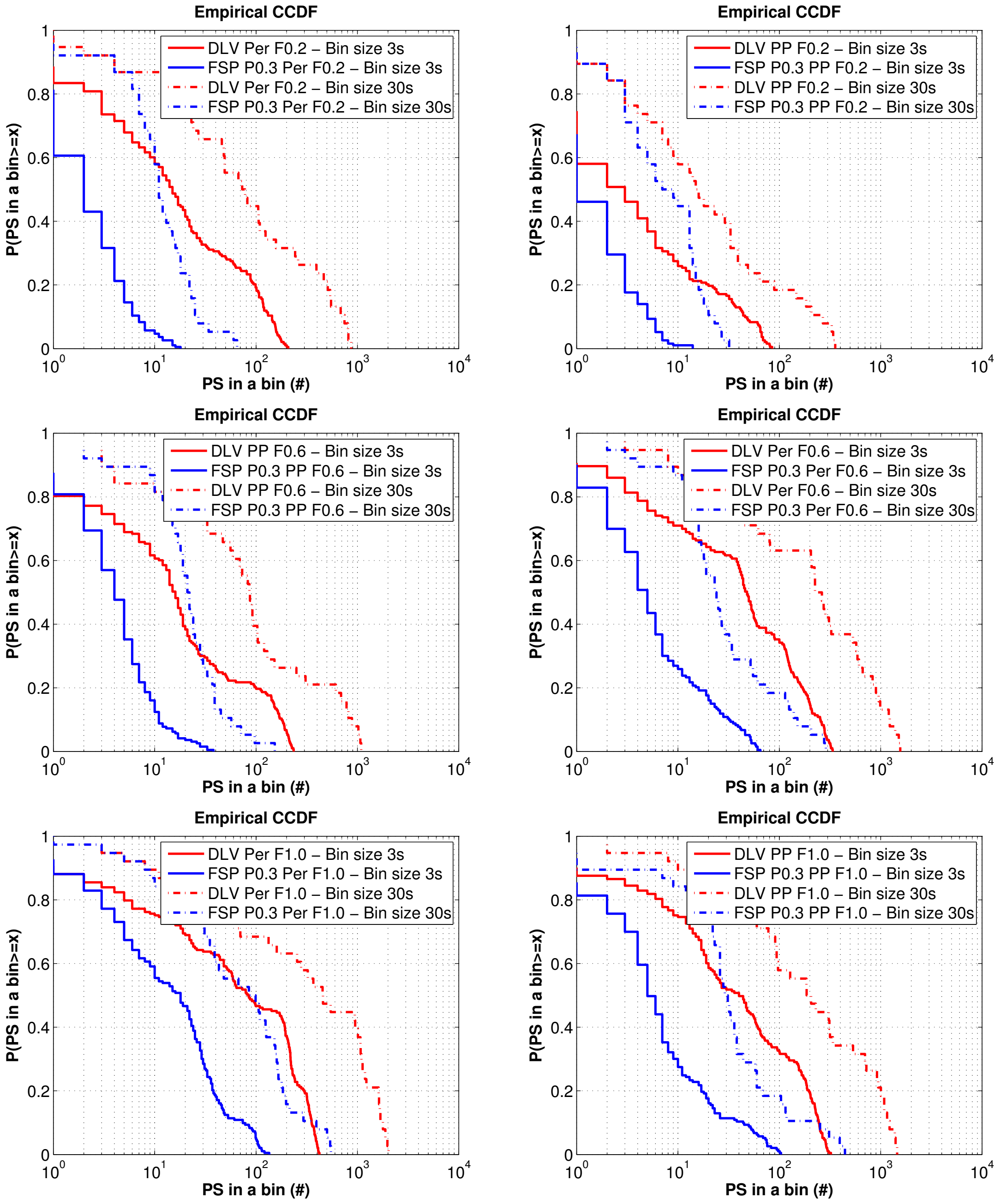


Figure 3.23: Comparison of DLV and FSP algorithms – CCDF for path switches performed in bins of 3 and 30s, for different sampling frequencies: (top) F=0.2Hz, (middle) F=0.6Hz, and (bottom) F=1.0Hz.

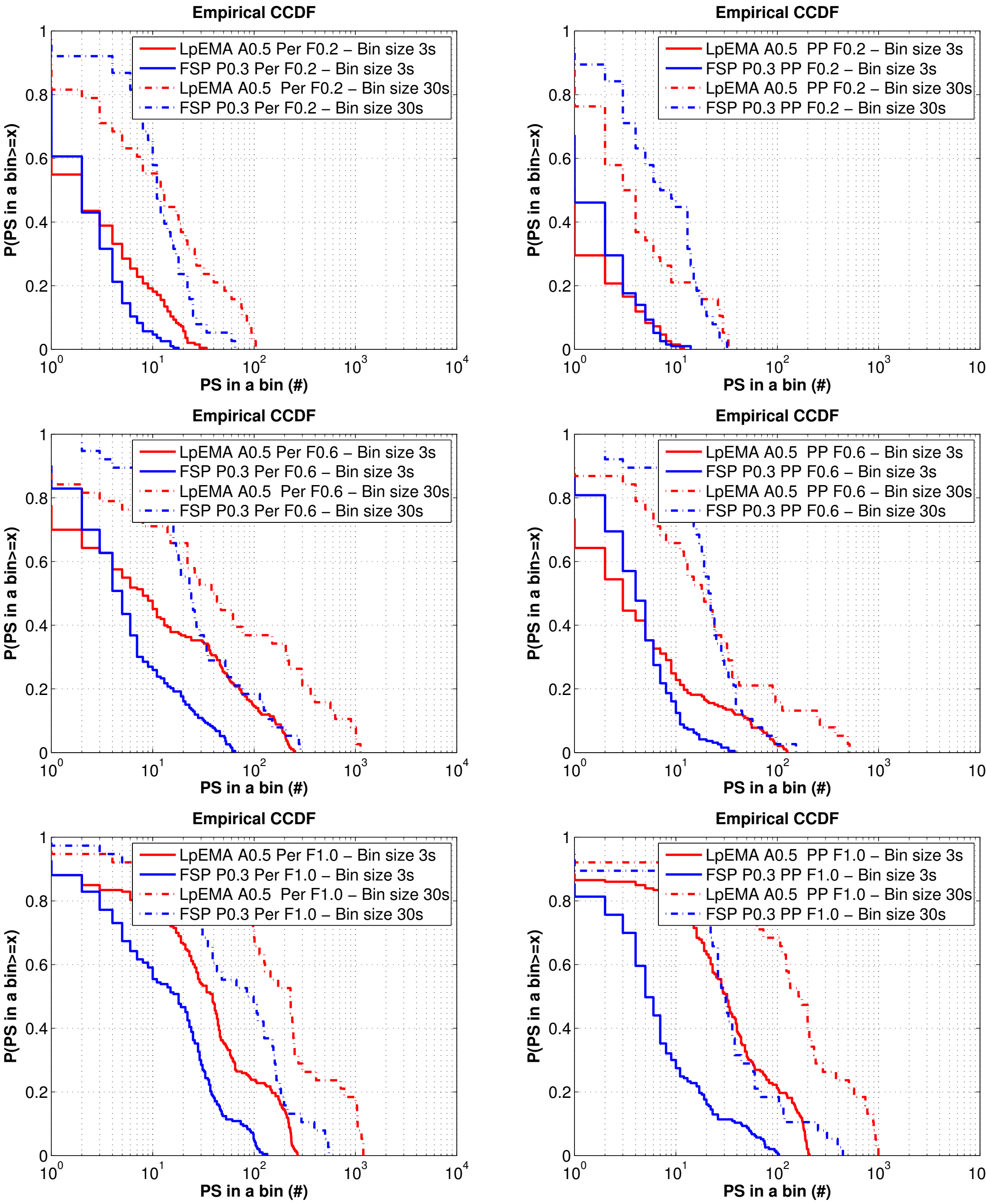


Figure 3.24: Comparison of LpEMA and FSP algorithms – CCDF for path switches performed in bins of 3 and 30s for different sampling frequencies: (top) F=0.2Hz, (middle) F=0.6Hz, and (bottom) F=1.0Hz.

3.9.1.4 Synthesis of the Results

In this study, the evaluation of three classes of Intelligent Route Control (IRC) algorithms – Randomized Path Monitoring, Randomized Path Switching and History-Aware Path Switching – was performed to find out which is the best alternative to cope with the problem of IRC oscillations.

The results showed that the addition of a randomness component to the route control process drastically reduces the number of path switches needed to meet the traffic challenges. They also showed that adding a randomness component to the path monitoring mechanism is an effective alternative to this solution. However, the decision to adopt a randomized path monitoring solution depends on the timescale that is employed by an IRC system to probe all the candidate paths. In particular, short timescales might impair its effectiveness due to the overlapping of the IRC measurement windows.

Finally, the value of using sophisticated IRC algorithms, such as history-aware path switching, is questionable, since these mechanisms require additional tuning, to allow them to conform to the particular stability pattern of the network. Nevertheless, a careful tuning shows that this class of algorithm has the potential to improve IRC stability, while ensuring good traffic performance.

3.10 Summary

This chapter has investigated the use of multihoming Intelligent Route Control (IRC) to enable inter-domain quality of service routing.

In the first stage, there was an outline of the main design principles of intelligent route controllers, architectural components and algorithms. The results from a simulation study have shown that intelligent route control can substantially enhance end-to-end quality of service when compared with a pure Border Gateway Protocol (BGP) model. We believe that although it is possible to envisage significant extensions and enhancements to BGP, decoupled routing control solutions like that investigated in this chapter, can be regard as strong candidates to provide flexible and value-added out-of-band inter-domain quality of service routing. In particular, this is a suitable solution when the inter-domain routing have to adapt and rapidly react to medium or extensive performance changes in the network in a dynamic way, and where the former in-band solutions seem impracticable at the present time.

In the second stage, owing to the fact that little is known about the mechanisms used by IRC systems to react to path failures and losses of probing packets, it was decided, in this chapter, to investigate the time needed by IRCs for detection and recovery from a path failure, and include two mechanisms for the dynamic adaptation of the timers used to declare that a probe is lost: a Jacobson-based mechanism and a Plot-box rule-based mechanism. We have shown that the original setting of the Jacobson-based algorithm performs quite well. However, the original setting of the statistical Plot-box rule tool is not able to prevent IRCs from running into spurious timeouts. Subsequently, we blend both mechanisms with a similar extension to avoid subsequent spurious timeouts after a spurious timeout had been detected. The main purpose of this extension is to fix the problem by finding a proper value for the next timeout timer based on the round-trip time of the sample that has been declared as lost. This strategy has one significant practical advantage: these mechanisms with a little tuning effort provide the ability to find proper values for timeout timers. The results confirmed this claim. They showed that the extended versions of these algorithms are capable of protecting IRC from spurious timeouts.

Finally, in this chapter, an evaluation of the three main classes of intelligent route control algorithms – randomized path switching, randomized path monitoring and history-aware path switching – was undertaken to find out what would be the best alternative to cope with the oscillations associated with this type of scheme. The results showed that the addition of a randomness component to the route control process drastically reduces the number of path switches needed to meet the traffic challenges. The results also showed that the addition of a randomness component to the path monitoring mechanism is an effective alternative to the former solution for a longer timescale of probing. Lastly, the use of sophisticated IRC algorithms, such as history-aware path switching, is questionable, since these mechanisms require additional tuning, to allow them to conform to the particular stability pattern of the network.

Chapter 4

Improving the Performance of Route Control Middleboxes in a Competitive Environment

This chapter provides the design and evaluation of a Sociable Route Control (SRC) model for multihomed stub domains in a competitive environment.

Bibliographical Notes. A part of this chapter has been published in [153]. Whilst much of this chapter is joint work undertaken with the Technical University of Catalonia under the auspices of E-NEXT and CONTENT $6^{\underline{o}}$ FP IST NoE projects, I made a significant contribution to the validation of the SRC model. In particular, I conducted the performance assessment studies, and devised the simulation models outlined in Sections 4.6 and 4.7.

4.1 Introduction

Multihoming combined with Intelligent Route Control (IRC) solutions are becoming a widespread practice in stub Autonomous Systems (ASs) and are aimed at improving the reliability of their Internet accesses, and optimizing costs and traffic performance. In this Chapter, stub ASs will be simply called as stub networks or domains. As described in Chapter 3, IRC systems allow to actively exploit the multi-connectivity of stub networks to the Internet by leveraging the relocation of a part of outbound traffic from one of the ISPs (Internet Service Provider) to another, by using measurement-driven path switching techniques.

Despite these strengths, IRC has one major weakness, which is that the IRC systems seek to achieve a set of local objectives on an individual basis, without taking account of the effects of their decisions on the performance of the network. It was recently found that, in a competitive environment, IRC systems can actually cause significant performance degradation, rather than making improvements [89]. In this work, the authors show that persistent oscillations can occur when independent controllers become synchronized as a result of a considerable overlap in their measurement time windows. To avoid this synchronization, the authors employ randomized IRC strategies, and empirically show that the oscillations disappear after a random component is introduced in the route control decision.

It should be stressed that although randomization offers a straightforward mechanism to mitigate the oscillations, it cannot guarantee global stability. This is giving rise to concern on account of the proliferation of IRC products, since as the number of interfering IRC systems increases, the randomization becomes less effective, and hence, it is more likely that the oscillations will reappear. In the light of this, it is necessary to explore more scalable route control strategies that can safely support the foreseeable spread of IRC solutions.

In principle, two research approaches can be adopted. On the one hand, the research community could formally study the stability properties of the IRC practices, and provide guidelines on how to design IRC systems with a guaranteed stability. However, several challenging stages have to be completed before a formal study of stability can be conducted. For instance, accurate measurements are needed to fully understand the actions of the closed-source IRC systems deployed today (like e.g., [87, 88, 154]), and thus model the stochastic distribution of path switches in a competitive IRC environment. Only after the distribution of path switches has been characterized, will it be possible to conduct a formal study of the stability of competitive IRCs.

In the absence of such a characterization, the practical alternative is to find ways to drastically reduce the potential interference between competing route controllers, without penalizing the end-to-end traffic performance. This is precisely the challenge that is addressed in this chapter.

In this chapter we provide and assess the performance of a Sociable Route Control (SRC) model for multihomed stub domains in a competitive environment. In the SRC model, each controller remains independent, which means that it does not need any kind of coordination with the other competing controllers in the network – the SRCs operate in a standalone fashion, just as conventional IRCs do. The key factor is that each controller is endowed with a social route control algorithm that adaptively restrains its intrinsic selfishness by learning from and evolving together with the network dynamics.

To be more precise, the route control decisions made by the SRCs not only depend on the current state of the network, but also on the dynamics of these states, insofar as the route control decisions are able to evolve and adapt together with the dynamics of the network. Under changing network conditions, it is imperative that each route controller can count on a social mechanism that allows it to adapt by diminishing or even preventing path switches, until the network conditions become stable again. Such changing network conditions might occur when a significant number of conventional IRCs compete for the same network resources during link flaps, or even when there are routing misconfigurations.

The contributions of this chapter are therefore:

(i) We show that although randomization offers a straightforward way to mitigate the problem of oscillations, it leads to a large number of unnecessary path switches.

(ii) We report our results on the development of strategies blending randomization with a lightweight and more sociable route control algorithm.

(iii) We show that, a simple enhancement of randomized IRC systems, like endowing them with a SRC algorithm supported by adaptive filtering techniques, is enough to drastically reduce the number of path switches, and most importantly, this can be accomplished without penalizing the end-to-end traffic performance. Simulation results show that with SRC it is possible to reduce the overall number of path switches approximately by between 40% and 80%, on average (depending on the load on the network), and still obtain better end-to-end traffic performance than with randomized IRC techniques in a competitive environment.

(iv) We investigate and test the behavior of the SRC strategy, that is now, operating on differentiated service-enabled networks and show that this model is also valid for these kinds of networks.

(v) To best of our knowledge, we have carried out the most comprehensive tests so far, to assess the performance of different IRC strategies in a competitive environment.

(v) The social extension set out in this chapter can be easily integrated and used today, since all that is needed is a software upgrade of the available route controllers.

The rest of the chapter is structured as follows. First, we discuss the basics of IRC. Then, we provide an overview of the most important related work. Following this, we analyze some general aspects of different IRC strategies, and describe the SRC approach, together with our main results, and include an analysis of the results of the experiments performed in differentiated service-enabled networks. We conclude with directions for future research in the area of IRC.

4.2 The Basics of IRC

A typical IRC scenario with two different configurations is shown in Figure 4.1. On the one hand, the IRC box on the top of Figure 4.1 is connected via a span port off a

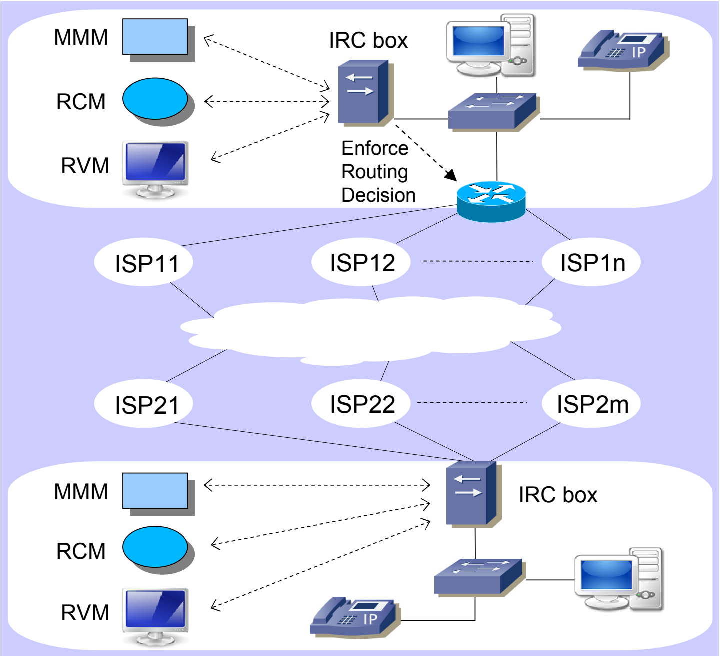


Figure 4.1: The IRC model.

router or switch, so even though the egress traffic is controlled by the box, it is never forwarded through it. On the other hand, the IRC box in the multihomed network on the bottom of Figure 4.1 is placed along the data path, so traffic is always forwarded through it. Typically, the former configuration offers a more scalable solution than the latter, in the sense that it is able to control and optimize a larger number of traffic flows.

Conceptually, an IRC system is composed by the following three modules:

- A Monitoring and Measurement Module (MMM)
- The Route Control Module (RCM)
- A Reporting and Viewer Module (RVM)

The existing IRC systems can control a moderately large number of flows toward a set of *target* destination networks. Typically, in the order of several hundreds, and even thousands using a configuration like the one shown at the top of Figure 4.1 with several border routers. These target destinations can be manually configured or discovered by means of passive measurements performed by the MMM. By using passive measurements, the MMM is able to rank the destinations according to the amount of traffic sourced from the local network, and subsequently optimize the performance for the traffic toward the D destinations at the top of the rank. The MMM also uses passive measurements to monitor the target flows in real-time, and analyze packet losses, latency, and retransmissions—among others—as indicators of conformance or degradation of the expected traffic performance. In order to assist the RCM in the dynamic selection of the best egress link to reach each target destination, the MMM probes all the candidate paths using both ICMP (Internet Control Message Protocol) and TCP (Transmission Control Protocol) probes.

The set of active and passive measurements collected by the MMM allow IRC systems to concurrently assess the quality of the active and the alternative paths toward the target destinations. The role of the RCM is to dynamically choose the best egress link for each target flow, depending on the outcome of these measurements. More specifically, the RCM is capable of taking rapid routing decisions for the target flows, often avoiding the effects of issues such as distant link/node failures, or performance degradation due to congestion. Recall that we have shown these issues in Chapter 3. We have observed that congestion cannot be automatically detected and avoided by BGP and the timescale needed by IRC systems to detect and react to a distant link/node failure is very small compared with that of the general IGP/BGP routing system.

The third module of an IRC system, namely the RVM, typically supports a broad set of reporting options, and provides on-line information about the average latency, jitter, bandwidth utilization, and packet loss experienced through the different providers, summaries of traffic usage and associated costs for each provider, etc.

Overall, IRC offers an incremental approach, complementing some of the key deficiencies of the IGP/BGP-based route control model. It is worth highlighting that the set of candidate routes to be probed by IRC boxes is usually determined by IGP/BGP, so conversely to overlay networks [122], IRC boxes never circumvent IGP/BGP routing protocols. The effectiveness of multihoming in combination with IRC is confirmed not only by studies like [122], but also by the increased trend in the deployment of these solutions.

In this chapter we deal with the algorithmic aspects of IRC systems, so hereafter we shall focus our attention on the RCM in Figure 4.1 – the functionalities of the MMM and RVM modules are essentially orthogonal to the proposals made in this work.

4.3 Related Work

In [155], the authors simultaneously optimize the cost and performance for multihomed stub networks, by introducing a series of new IRC algorithms. The contributions in that work are fundamentally theoretical. For instance, the authors show that an intelligent route controller can improve its own performance without adversely affecting other controllers in a competitive environment, but the conclusions are drawn at traffic equilibrium (traffic equilibrium is defined by the authors as a state in which no traffic can improve its latency by unilaterally changing its link assignment).

However, after examining and modeling the key features of conventional IRC systems, it becomes clear that they do not seek such kind of traffic equilibrium. Indeed, more recent studies, like [89], have shown that, in practice, the performance penalties can be large, especially, when the network utilization increases.

In light of this, and considering the current deployment trend of IRC solutions, it becomes necessary to explore alternative IRC strategies. These new route control strategies should *always* improve the performance and reliability of the target flows, or at least, they should drastically reduce the potential implications associated with frequent traffic re-locations, such as persistent oscillations causing packet losses and increased packet delays [89].

Although most commercially available IRC solutions do not reveal in depth the technical details of their internal operation and route control decisions, the behavior of one particular controller is described in detail in [54]. That work also contributes with measurements evaluating the effectiveness of different design decisions and load balancing algorithms. Akella et al have also provided rather detailed descriptions and experimental evaluations of multihoming in combination with IRC tools, like [85], [122], and [156]. These research publications, along with the documentation provided by vendors, allowed us to capture and model the key features of conventional IRC techniques. A similar approach was followed by the authors in [89]. For simplicity, and as in [89, 122], and [54], we shall consider traffic performance as the only criteria to be optimized for the target flows. Notice that cost reductions are typically accomplished by aggregating traffic toward non-target destinations over the cheapest ISPs.

4.4 The General IRC Network Model

The general IRC network model is composed by a multihomed stub network S, a route controller $\mathbb{C}$, the transit domains, and a set of target destinations $\{d\}$ with cardinality $|d| = D$ to be optimized by $\mathbb{C}$. The source domain S has a set of egress links $\{e\}$, with $|e| = E$. For the sake of simplicity, we will keep the notation in the granularity of destinations (d), but the model can be easily extended to consider various flows per target d.

In order to dynamically decide the best egress link for each target destination d, the MMM in $\mathbb{C}$ probes all the candidate paths through the egress links e of S. The collected measurements are then processed and abstracted into a performance function $P_e^{(d,t)}$ at time t, associated with the quality perceived for each of the available paths toward the target destinations d. Let $N^{(d)}$ denote the number of available paths to reach d. Since usually $N^{(d)}$ represents the number of candidate paths in the Forwarding Information Base (FIB) of the BGP border routers of S, $N^{(d)} \leq E \; \forall \; d$.

We assume that the better the end-to-end traffic performance perceived by $\mathbb{C}$ for a target destination d through egress link e, the lower the value of the performance function $P_e^{(d,t)}$.

In this framework, IRC strategies can be taxonomized into two categories, namely, *Reactive Route Control* (RRC), and *Proactive Route Control* (PRC). RRC practices switch a target flow from one egress link to another only when a *Maximum Tolerable Threshold* (MTT) is met. The MTTs are application-specific, and typically represent the maximum acceptable packet loss, the maximum tolerated packet delay, etc, for a given application. Beyond any of these bounds, the performance perceived by the users of the application becomes unacceptable.

PRC strategies on the other hand, switch traffic before any of the MTTs are met, and can be in turn taxonomized into two categories, those that can be called *Fully Proactive* (FP), and those that follow a *Controlled Proactivity* (CP) approach. FP IRC practices always switch to the best path. Therefore, the dynamic optimization problem addressed by a FP route controller is to:

Find the min$\{P_e^{(d,t)}\}$ $\forall$ *d, t, and enforce the redirection of the corresponding traffic to the egress link found.*

The alternative offered by CP is to keep the proactivity, but switch traffic as soon as the performance becomes degraded up to some extent, typically represented by a *Relocation Threshold* (R_{th}). The dynamic optimization problem addressed by CP-based strategies can be formulated as follows.

Let e^{best} *denote the egress link utilized to reach* d *at time* t*, and let* e' *be such that* $P_{e'}^{(d,t)} = min\{P_e^{(d,t)}\}$ *for destination* d *at time* t*.*[1] *A CP-based route controller would switch traffic to* d *from* e^{best} *to* e' *whenever* $P_{e^{best}}^{(d,t)} \geq R_{th}$ *and* $P_{e'}^{(d,t)} < R_{th}$*.*

After extensive evaluations and analysis, we have confirmed that PRC performs much better than RRC. The reason for this is that proactive approaches are able to anticipate network congestion situations, which in the reactive case, typically demands several traffic re-locations when congestion has already been reached. In addition, we have found that, in a competitive environment, CP-based route control strategies are able to outperform the FP ones. Therefore, our SRC algorithm (outlined in next Section) is supported by a CP-based route control strategy.

4.5 Sociable Route Control (SRC)

In the SRC strategy that we conceive, each controller remains independent, so the SRC boxes do not need any kind of coordination between each other – just as conventional

[1] *We notice that with CP,* e^{best} *might be different from* e'*.*

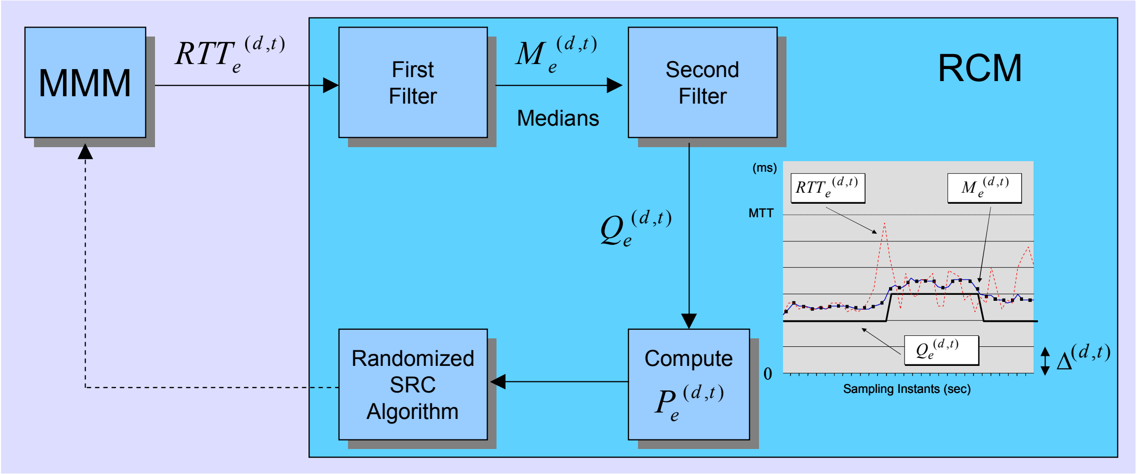


Figure 4.2: Filtering process and interaction between the Monitoring and Measurement Module (MMM) and the Route Control Module (RCM) of a sociable route controller.

IRC systems operate today. Moreover, our SRC strategy does not introduce changes in the way measurements are conducted and reported by conventional IRC systems, so both the MMM and the RVM in Figure 4.1 remain unmodified. Our SRC strategy only introduces changes on the algorithmic aspects of the Route Control Module (RCM).

4.5.1 High-level Description of the SRC Strategy

For simplicity in the exposition, we focus on the optimization of a single application, namely, VoIP, and we shall describe the overall SRC process for the Round-Trip Time (RTT) performance metric.

Our goal is that a controller $\mathbb{C}$ becomes capable of adaptively adjusting its proactivity, depending on the RTT conditions for each target destination d. To be precise, a sociable controller analyzes the evolution of the RTT, i.e., $\left\{RTT_e^{(d,t)}\right\}$, and depending on its dynamics, the controller can adaptively restrain its traffic reassignments (i.e., its proactivity). To this end, the RCM processes the RTT samples gathered from the MMM using two filters in cascade (see Figure 4.2). The first filter corresponds to the median RTT, $M_e^{(d,t)}$, which is constantly computed through a sliding window. This approach is widely used in practice, since the median represents a good estimator of the delay that the users' applications are currently experiencing in the network. These medians are precisely the input to the second filter, where the social nature of the route control algorithm covers two different facets: i) Controlled Proactivity (CP); and ii) Socialized Route Control.

4.5.2 Controlled Proactivity (CP)

On the one hand, the proactivity of the box $\mathbb{C}$ is controlled so as to avoid that minor changes in the medians trigger traffic relocations at S. This prevents interfering too often with other route controllers. For this reason, our sociable controllers filter the medians.

The second filter in Figure 4.2 works like an Analog-to-Digital (A/D) converter, with quantization step Δ, and its output is one of the levels of the converter $Q_e^{(d,t)}$. The right-hand side of Figure 4.2 illustrates how the instantaneous samples of RTT are filtered to obtain the median $M_e^{(d,t)}$, and the latter is then filtered to obtain $Q_e^{(d,t)}$.

As described in Section 4.4, IRC systems compare the quality of the active and alternative paths by means of a performance function $P_e^{(d,t)}$, which as shown in Figure 4.2, is fed by $Q_e^{(d,t)}$. The controller $\mathbb{C}$ would switch traffic toward d, only when the variations of $Q_e^{(d,t)}$ cause that $P_e^{(d,t)} \geq R_{th}$ along the active path. A more detailed description of the route selection process is shown in Algorithm 4.1. For simplicity, only the stationary operation of the algorithm is summarized. The randomized nature of Algorithm 4.1 is discussed latter in Section 4.5.4. The timer in Step 8 is also introduced in Section 4.5.4.

For the RCM described here, we have simply used the outcome of the digital conversion as the performance function $P_e^{(d,t)}$, that is, the number of quantization steps in the quantification level $Q_e^{(d,t)}$. Similarly, R_{th} represents the number of quantization steps that $P_e^{(d,t)}$ needs to reach in order to trigger a path switch.

Overall, the advantage of this filtering technique is that it produces the desired effect (i.e., controlled proactivity), since it prevents minor changes in the medians from triggering unnecessary traffic re-locations at S.

4.5.3 Socialized Route Control

The second facet of the social behavior of the algorithm has to do with the dynamics of the median RTTs. More precisely, with how rapid are the variations in the median values – which are typically computed by IRC systems using a sliding window. The motivation for this is that when the median values start to show rather quick variations, the algorithm must react so as to avoid a large number of traffic reassignments in a short timescale. Such RTT dynamics typically occur when several route controllers compete for the same resources, leading to situations where their traffic reassignments interfere between each other. To cope with this problem, we turn the second filter in Figure 4.2 into an adaptive filter. This filter is endowed with an adaptive quantization step $\Delta^{(d,t)}$ for each target destination d, which is automatically adjusted by the algorithm according to the evolution of the median RTTs. If the RTT conditions are smooth the quantization step is small, and more proactivity is allowed by the controller $\mathbb{C}$.

However, if the RTT conditions may lead to instability the quantization step $\Delta^{(d,t)}$ automatically increases, so the number of changes in the values of $Q_e^{(d,t)}$ is diminished or even stopped until the network conditions become smooth once again. This has the effect of de-synchronizing only the competing route controllers. Therefore, the filtering technique outlined here allows a controller $\mathbb{C}$ to "sociably" decide whether to switch traffic to an alternative egress link or not, in the sense that the degree of proactivity of $\mathbb{C}$ is constantly adjusted by the adaptive nature of the second filter.

For the sake of simplicity, we have focused here on the optimization of a single performance metric (the RTT), but the concept of SRC is general and can be extended to consider other metrics, such as available bandwidth, packet losses, and jitter. When multiple metrics are used, two straightforward approaches can be followed.

On the one hand, a combination of two or more metrics can be used in the same performance function $P_e^{(d,t)}$. For instance, [157] introduces a more general performance function based on a non-linear combination of the quantification level $Q_e^{(d,t)}$ and the Available Bandwidth (Avail-Bw) in the egress links of the source network. This, in turn, can be extended to consider the AB along the entire path to a target destination d, using available bandwidth estimation techniques like the one described in [89]. With this approach, the weights of the different metrics combined in $P_e^{(d,t)}$ can be tuned on an application basis – for example, to prioritize the role of the AB over the RTTs (or vice versa) depending on the application type.

On the other hand, multiple performance functions $P_e^{(d,t)}$ can be used (e.g., one for each metric), and the selection of the best path for each target destination can be done by sequentially comparing the performance functions $P_e^{(d,t)}$, and tie-breaking similarly to the BGP tie-breaking rules [158]. With this approach, the order in which the performance functions are compared can be tuned on an application basis. For example, a controller might select the path with the maximum AB, and if there is more than one path with the same AB choose the one with the lowest RTT.

In either case, adaptive filtering techniques are needed to prevent rapid variations in the performance metrics considered.

4.5.4 Randomization

Randomization is present in Algorithm 4.1 in two different ways, implicitly, and explicitly. On the one hand, the route control decisions in Algorithm 4.1 are inherently stochastic for a number of reasons. For example, due to its adaptive features along time, the fact that different controllers might have configured different thresholds R_{th}, and more.

On the other hand, we explicitly use a *Hysteresis Switching Timer* T_H that we introduced in a previous work [112], and which guarantees a random hysteresis period

Algoritmo 4.1 Randomized SRC Algorithm, SRC($\{d, \{e\}, P_e^{(d,t)}\}$)

```
Require: $d$ - A target destination of network $S$
         $\{e\}$ - Set of egress links of network $S$
         $P_e^{(d,t)}$- Performance function to reach $d$ through $e$ at time $t$
Ensure: $e^{best}$ - The best egress link to reach target destination $d$
 1: $R_{th} \leftarrow K$ /* Configurable threshold to trigger the path switch */
 2: Wait for changes in $P_{e^{best}}^{(d,t)}$
 3: if $P_{e^{best}}^{(d,t)} < R_{th}$ then go to Step 2
 4: /* Egress link selection process for $d$ */
 5: if there exists $e$ such that $P_{e^{best}}^{(d,t)} > P_e^{(d,t)}$ then
 6:     Choose $e'$ as $P_{e'}^{(d,t)} = min\{P_e^{(d,t)}\}$
 7:     Estimate the performance after switching the traffic
 8:     if $(P_{e'}^{(d,t)})_{Estimate} < R_{th}$ then
 9:         Wait until $T_H = 0$ /* Hysteresis Switching Timer */
10:         Switch traffic toward $d$ from $e^{best}$ to $e'$
11:         $e^{best} \leftarrow e'$
12:         $P_{e^{best}}^{(d,t)} \leftarrow P_{e'}^{(d,t)}$
13:     end if
14: end if
15: /* End of egress link selection process for $d$ */
16: Go to Step 2
```

after each traffic relocation. More precisely, traffic toward a given destination d cannot be relocated until the random and decreasing timer $T_H = 0$. A similar approach was used in [89], for one of the randomized algorithms presented there.

4.6 Evaluation

This section presents the simulation set-up developed to assess the advantages of the social route control model. The performance of the SRC strategy is compared with that obtained from the following alternative models:

- Randomized IRC;
- Default IGP/BGP routing.

4.6.1 Evaluation Methodology

The simulation tests were carried out by using the event-driven simulator J-Sim [145]. All the functionalities of the route controllers were developed on top of the IGP/BGP implementation available in this platform, the BGP Infonet suite.

AS-level Topology. The network topology was with the aid of the Boston University Representative Internet Topology gEnerator (BRITE) [151]. The topology was generated by means of the Waxman model with (α, β) set to $(0.15, 0.2)$ [159], and was composed of 100 ASs with 1:3 ratio of ASs to inter-domain links. The simulated network aims at representing a set of ISPs that can provide connectivity and reachability to customers operating stub networks. We assume that all the ISPs operate Points of Presence (PoPs) through which the stub networks are connected.

We considered 12 uniformly distributed stub networks across the AS-level topology as the traffic sources toward the set of target destinations. These source networks are connected to the routers located at the PoPs of three different ISPs. The reason for selecting triple-homed stub networks was that significant improvements in performance are not expected from higher degrees of multihoming [85].

For the stub networks containing target destinations, we considered 25 uniformly distributed destinations across the AS-level topology. This offered an emulation of $12 \times 25 = 300$ IRC flows competing for the same network resources during the simulation runtime.

It is worth highlighting that the size of the AS-level topology used during our evaluations is small compared to the size of the Internet. However, to the best of our knowledge, this is the largest test ever conduct to assess the performance of different IRC strategies in a competitive environment.

Furthermore, given the fact that IRC solutions operate in short timescales, we assumed that the AS-level topology remains invariant during the simulation run time.

Simulation Scenarios. In our experiments, we ran the same simulations separately using three different scenarios:

(a) Default IGP/BGP routing, where BGP routers choose their best routes on the basis of the shortest AS-path.

(b) BGP combined with the SRC strategy at the 12 source stub networks.

(c) BGP combined with randomized IRC systems at the 12 source stub networks.

To make more comprehensive comparison between the different route control strategies, we performed the simulations for three different network loads. We considered the following load factors (L):

(i) $L = 0.450$, low load corresponding to an average occupancy of 45% of the egress link capacity.

(ii) $L = 0.675$, medium load corresponding to an average occupancy of 67.5% of the egress link capacity.

(iii) $L = 0.900$, high load corresponding to an average occupancy of 90% of the egress link capacity.

Synthetic Traffic and Simulation Conditions. The simulation tests were conducted by using traffic aggregates sent from the source networks to each target destination d. These traffic aggregates were composed of a variable number of multiplexed Pareto flows as a way to generate traffic demands, as well as to control the network load during the tests. The flow arrivals were modeled in accordance with a Poisson process and were independent and uniformly distributed during the simulation run time. This approach aims at generating sufficient traffic variability to support an assessment of the different route control strategies.

In addition, we used the following method to generate traffic demands for the remaining Internet traffic, usually referred to as background traffic. We started by randomly picking four nodes in the network. The first one chosen acts as the origin (O) node, and the remaining three nodes act as destinations (D) of the background traffic. We assigned one Pareto flow for each O-D pair. This process continued until all the nodes had been assigned with three outgoing flows (including those in the multihomed stub networks and those in the ISPs). All the background connections were active during the simulation runtime.

In addition, the frequency and size of the probes sent by the route controllers were correlated with the outbound traffic being controlled, just as conventional route controllers do today [87, 88, 154].

Finally, we assume that the route controllers have pre-established performance bounds (i.e., the RTTs) for the traffic under control. For instance, the recommendation G.114 of the International Telecommunication Union-Telecommunication Standardization Sector – (ITU-T) suggests a one-way-delay (OWD) bound of 150 milliseconds to maintain a high quality VoIP communication over the Internet. Thus, for VoIP traffic, the maximum RTT tolerated was chosen to be twice this OWD bound, that is, 300 ms.

4.6.2 Evaluation Objectives

Our evaluations have two main objectives.

(1) To assess the number of path switches: The first objective of the simulation study is to demonstrate that the sociable nature of our SRC strategy serves to drastically reduce the potential interference between competing route controllers. This involved comparing the number of path switches that occurred during the simulation run time for the 300 competing IRC flows in the SRC and randomized IRC scenarios. The number of path switches is obtained by adding up the number of route changes that are required to meet the desired RTT bound for each target destination d.

It is worth emphasizing that in both the randomized IRC and SRC strategies, the route controllers operate independently and compete for the same network resources. This allows us to evaluate the overall impact on the traffic caused by the interference between several standalone route controllers running at different stub ASs. Thus, when analyzing the results for the different route control strategies, it is important to bear in mind that all the competing route controllers present in the network are taken into account.

To contrast the number of path switches under fair conditions, we made the following decisions. First, both the randomized IRC and SRC controllers would be endowed with the same (explicit) randomization technique [112], [89]. This approach prevents the appearance of persistent oscillations that might lead to a large number of path switches in the case of conventional IRC [89]. Second, both types of controllers would follow a controlled proactivity approach. We conducted the simulations modeling the same triggering condition R_{th} for both of them. The main difference is that in the case of SRC, the social adaptability of the controllers can result in the trigger being reached more often, or less often, depending on the variability of the RTTs on the network.

(2) End-to-end traffic performance: The second objective of the simulation study is to demonstrate that the drastic reduction in the number of path switches obtained with our SRC strategy can be achieved without penalizing the end-to-end traffic performance. This was carried out by comparing the RTTs obtained for the 300 flows in the three different scenarios, namely, default IGP/BGP, SRC, and randomized IRC.

4.6.3 Main Results

The left of Figure 4.3 illustrates the total number of path switches performed by both the randomized IRC and SRC strategies, in all the stub networks, and for the three different load factors, L=0.450 (left), L=0.675 (center), and L=0.900 (right). The number of path switches is contrasted for different triggering conditions, i.e., for different values of the threshold R_{th} (shown on a logarithmic scale). No results are shown for the default IGP/BGP routing scenario here, since BGP does not actively perform path switching.

Several conclusions can be drawn from the results shown in Figure 4.3. In the first place, the results confirm that, when compared to a randomized IRC technique, SRC drastically reduces the number of path switches. An important result is that the reductions are significant for *all* the load factors assessed. When compared with randomized IRC, our SRC strategy achieves, for instance, to reductions of up to:

- 77% for R_{th}=1, and 71% for R_{th}=2, when L=0.450.
- 75% for R_{th}=1, and 74% for R_{th}=2, when L=0.675.
- 34% for R_{th}=1, and 36% for R_{th}=2, when L=0.900.

The second observation is that the reductions in the number of path switches offered by the SRC strategy, become increasingly evident as the proactivity of the controllers increases, i.e. for low values of R_{th}, which is precisely the region where the IRC solutions operate today. It is worth recalling that these results were obtained when both route control strategies were supported by the same randomized decisions. This confirms that, in a competitive environment, SRC is much more effective than pure randomization in reducing the potential risk of interference between route controllers. On the other hand, our results show that when the route control strategies become less proactive, i.e. for higher values of R_{th}, randomized IRC and SRC tend to behave in comparatively the same way, which suggests that there are no advantage in SRC compared with a randomized IRC technique.

In assessing the effectiveness of SRC, it is essential to ensure that the reductions obtained in the number of path switches are not excessive, as this has a negative impact on the end-to-end traffic performance. This can be determined by first analyzing the performance of randomized IRC and our SRC "globally", i.e., by averaging out the RTTs obtained by all the competing route controllers. This is shown at the right of Figure 4.3, and in Figure 4.4. The end-to-end performance obtained by "each" route controller individually, is shown later in Figure 4.5.

The left of Figure 4.3 reveals that –as expected– both SRC and randomized IRC perform much better than IGP/BGP for all values of L, and R_{th}, and the improvements in the achieved performance become more evident as the network utilization increases. In particular, SRC is capable of improving the $\langle RTTs \rangle$ by more than 40% for $L = 0.675$, and by more than 35% for $L = 0.900$, when compared with IGP/BGP. As mentioned earlier, $\langle RTTs \rangle$ average is computed over the RTTs obtained by all the competing route controllers in the network.

Moreover, the $\langle RTTs \rangle$ obtained by SRC and IRC are almost the same, and particularly, for $L = 0.675$, SRC not only drastically reduces the number of path switches, but also improves the end-to-end performance for almost all the triggering conditions assessed. It is worth highlighting that a low value of R_{th} together with a load factor of $L = 0.675$, reasonably reflect the conditions in which IRC operates in today's Internet.

Our results also reveal an important factor: by allowing more path switches, some route controllers can slightly improve their end-to-end performance, but these actions have no major effect on the overall $\langle RTTs \rangle$. Indeed, a certain number of path switches is always required, and this amount of path switches is what actually ensures the average performance observed in the RTTs at the left of the Figure 4.3 (this becomes clear as the proactivity decreases).

By analyzing Figure 4.3 as a whole, it is evident that the selection of the best triggering condition actually depends on the load present on the network. In this particular case, the best trade-offs are $R_{th} = 30$ for $L = 0.450$, $R_{th} = 10$ for $L = 0.675$, and $R_{th} = 7$ for $L = 0.900$, which is a reasonable progression to lower values of R_{th}, since the route controllers need less proactivity when the network utilization is low. The corollary is that the triggering condition should be adaptively adjusted as well, depending on the amount of traffic carried through the egress links of the domain. We plan to investigate this in the future.

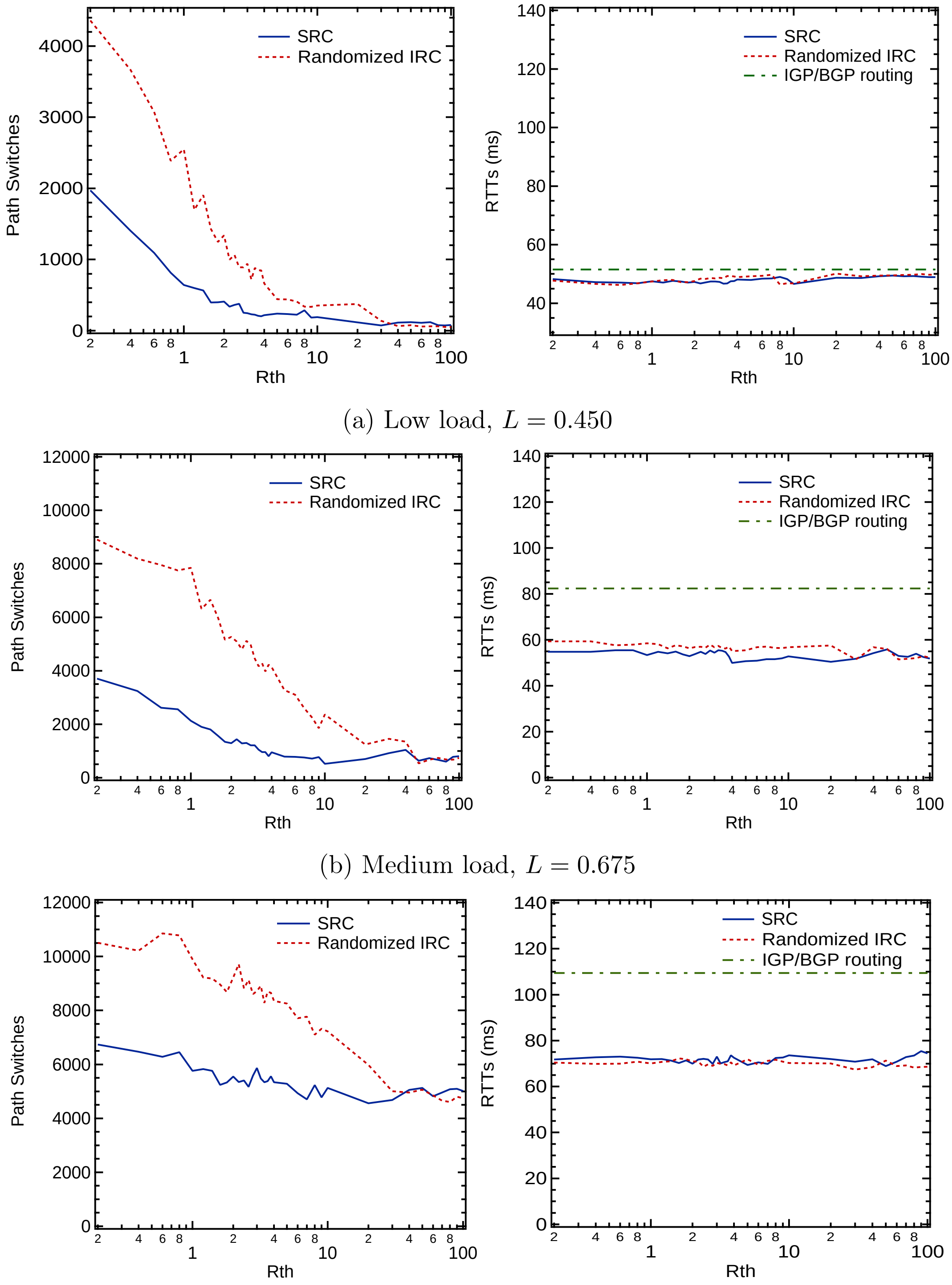
Figure 4.4 compares the distribution of the RTTs obtained by IGP/BGP, SRC, and randomized IRC, for the 300 competing IRC flows, the three different load factors assessed, and $R_{th} = 1$, which as mentioned above, is in the range of operation of the IRC solutions currently deployed in the Internet. The Complementary Cumulative Distribution Function (CCDF) is used to clarify the interpretation of the results.

An important observation is that under high egress link utilization, i.e., $L = 0.900$, there is a fraction of $\langle RTTs \rangle$ for which, in the case of IGP/BGP, the bound of 300 ms is exceeded, whereas both SRC and the randomized IRC fulfill the targeted bound.

To complete the analysis, Figure 4.5 provides a more granular picture than Figure 4.4, which is to show the CCDFs of the RTTs obtained by each of the 12 competing route controllers. The figure shows the findings for the three studied scenarios, and for all the load factors assessed when $R_{th} = 1$. Our results show that the targeted bound of 300 ms is satisfied by both SRC and randomized IRC in all cases, and for all controllers. IGP/BGP, however, shows a distribution of large delays due to the fact that the shortest AS-paths are not necessarily the best performing paths.

Figure 4.5 also shows that, when the boxes are considered individually, randomized IRC achieves slightly better end-to-end performance for some of them, but at the price of a much larger number of path switches: i) $\approx$ 435% larger for L=0.450; ii) $\approx$ 400% larger for L=0.675; and iii) $\approx$ 80% larger for L=0.900, when $R_{th} = 1$.

Overall, it can be concluded that these results provide strong evidence that the two evaluation objectives listed in Subsection 4.6.2 have been achieved.

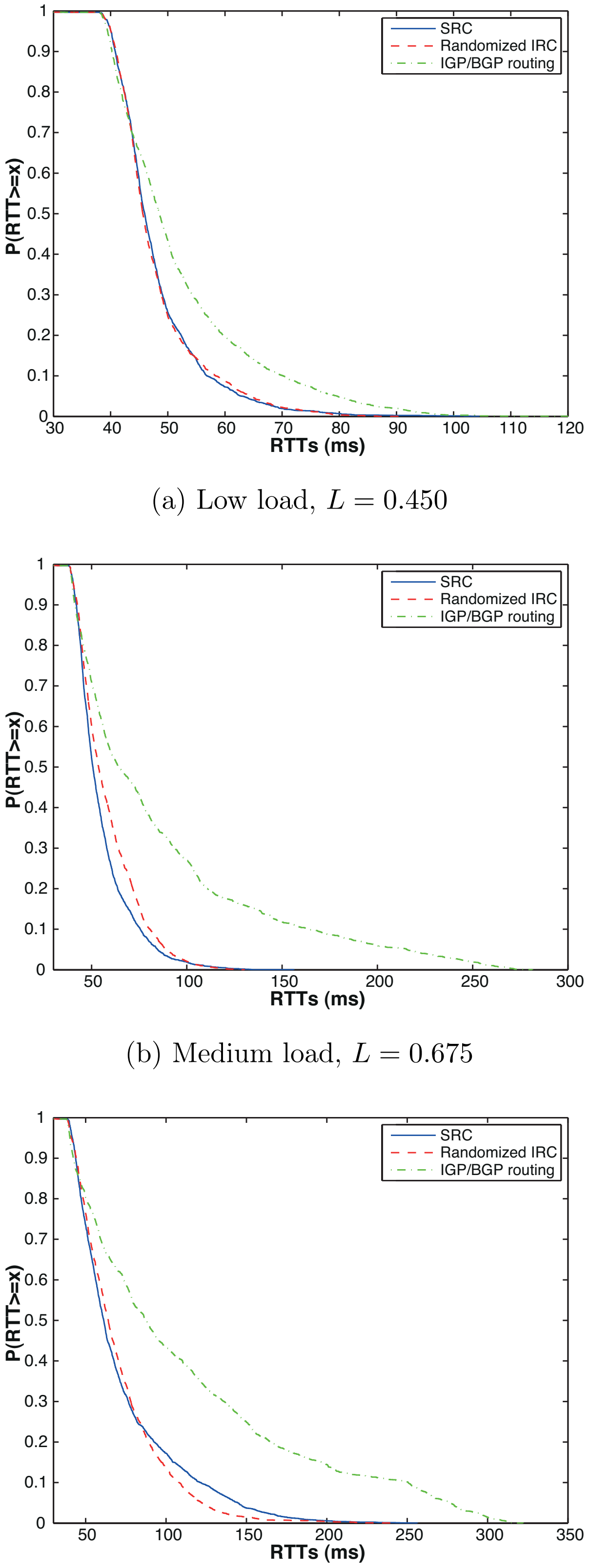


(a) Low load, $L = 0.450$

(b) Medium load, $L = 0.675$

(c) High load, $L = 0.900$

Figure 4.3: (left) Number of path switches and (right) $\langle RTTs \rangle$.



(a) Low load, $L = 0.450$

(b) Medium load, $L = 0.675$

(c) High load, $L = 0.900$

Figure 4.4: Complementary Cumulative Distribution Function (CCDF) of the RTTs for the 300 competing IRC flows, for $R_{th} = 1$, and for (top) $L = 0.450$, (center) $L = 0.675$, and (bottom) $L = 0.900$.

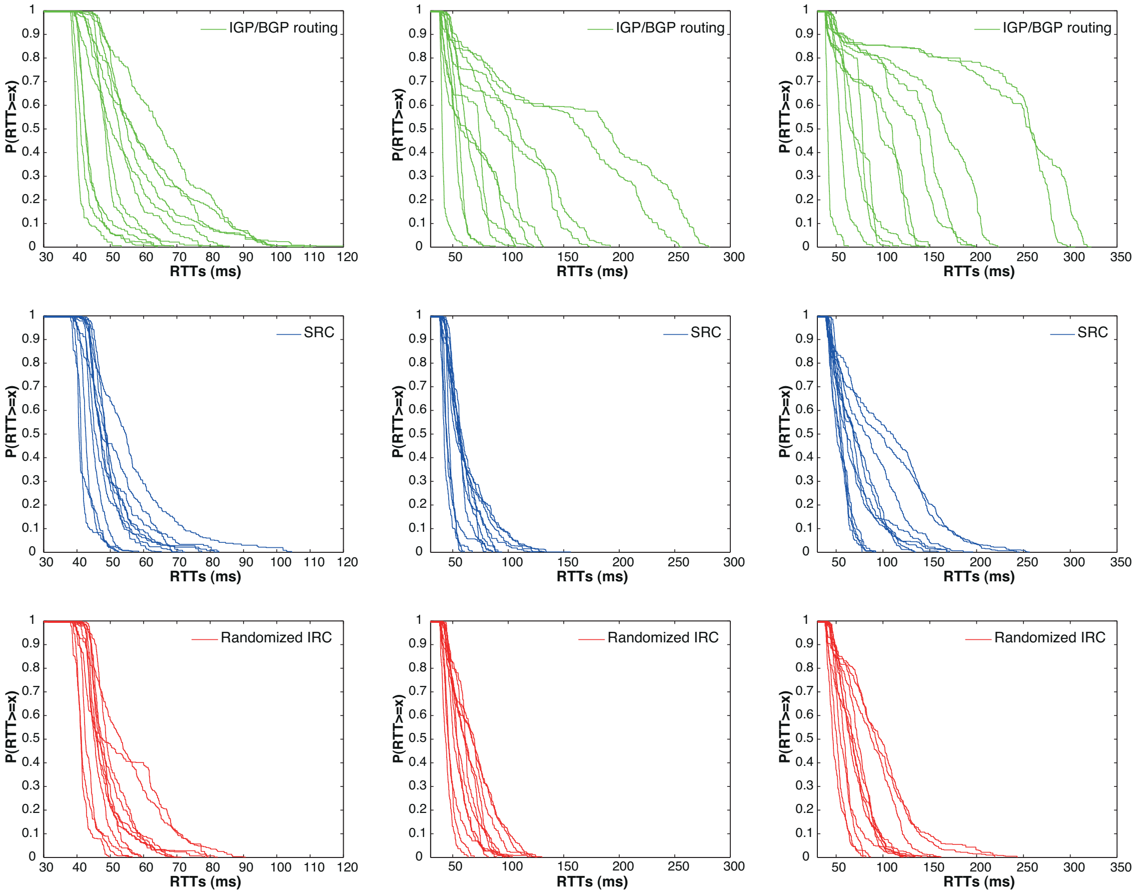


Figure 4.5: (top) CCDFs for IGP/BGP routing, (center) SRC, and (bottom) randomized IRC, for (left) $L = 0.450$, (center) $L = 0.675$, and (right) $L = 0.900$.

4.7 Performance Assessment of SRC on Differentiated Service-enabled Networks

In this section, we investigate and test the behavior of the sociable IRC (SRC) systems that were described in the preceding sections, and are now operating on a Differentiated service (Diffserv)-enabled network [6].

4.7.1 Simulation Set-up and Objectives

The simulations tests were carried out, again, with the aid of the J-Sim [145] simulator with the BGP Infonet suite [146] in which the functionalities of the SCRs have been implemented. To study the performance of the SRC strategy more accurately, while avoiding the self-load effect [89], we combined latency and spare bandwidth into a single metric, i.e., $M = \alpha.latency_t + \frac{\beta}{abw_t}$, where $latency_t$ is the median of the measured One-Way Delay (OWD) in a window t of $36s$, and abw_t is the estimated spare bandwidth in the peering link during t. For the interested reader, the tuning of the weights (α, β) follows the SRC implementation found in (the joint-work in) [152, 160].

Simulation scenarios. In our experiments we run the same simulations separately (and incrementally) using four different scenarios:

(a) BGP, i.e., all traffic is Best-Effort (BE) hence neither DiffServ nor IRCs are present during these simulations;

(b) BGP combined with DiffServ at *the edge of the network*, so no IRCs are present during these simulations;

(c) BGP combined with DiffServ at *the edge of the network*, as well as conventional IRCs;

(d) BGP combined with DiffServ at *the edge of the network*, as well as sociable IRCs.

Network model. For the simulation tests, the network model described in Chapter 3 was used (see Section 3.7). This topology represents a scenario where multihomed stubs employ IRC, and access ISPs are able to provide some limited QoS services to customers by using standard DiffServ capabilities, and an Internet core relies on the over-provision technique to address congestion problems. In this model, it is assumed that the peering IRCs are able to exchange a Service Level Specification (SLS) and agree upon a set of Quality of Service (QoS) parameters concerning the traffic between them.

In all the experiments, we used two slightly different SLSs that were exchanged between remote stub domains, and based on maximum OWD for voice, video and prioritized data traffic. The maximum OWDs tolerated per-service were chosen heuristically to represent reasonable but demanding values for the kinds of traffic sources considered. This is shown in Table 4.1. The aim is to assess the performance of the IRCs when realistic but tough SLSs are in use, so as to increase the probability of SLS violation events.

Table 4.1: Descriptions of Service Level Specifications (SLS): DiffServ Code Points (DSCP) and maximum One-Way Delays (OWD) tolerated (ms) by each service.

SLS ID	Voice (EF)	Video (AF11)	Prioritized Data (AF21)
1	45	55	75
2	60	65	85

Traffic models and default parameters. In this set-up, the tests were conducted by using a traffic mix consisting of up to forty Voice over IP (VoIP) calls, up-to twenty video calls, prioritized data, and web traffic. New voice and video connections arrived at border routers of AS1-AS4 and the corresponding durations are uniformly distributed between [500, 1500], and [180, 300] seconds respectively. Both kinds of connection calls are active along all the simulation runtime, which is [500, 1500] seconds. The source models are similar to the ones that were employed in [150], including the corresponding traffic DiffServ Code Point (DSCP), and are as follows:

(i) `EF` Voice traffic is marked with `EF` code-point [149]. `EF` source is an ON-OFF VoIP generator characterized by Exponential ON-OFF model. The ON and OFF states are characterized by an random variable with mean ON (burst) time of 400ms, and mean OFF (idle) time of 600ms, respectively. In the on state, `EF` voice source generates traffic at a peak rate of 64kbps. The size of ON-OFF `EF` packets is equivalent to 576 bytes;

(ii) Video traffic is marked with `AF11` code-point [148]. `AF11` source is a video traffic generator characterized by a similar Pareto ON-OFF model. The ON and OFF states are characterized by an random variable with mean ON (burst) time of 300ms, and mean OFF (idle) time of 400ms, respectively. The Pareto shape parameter is 1.9. In the on state `AF11` source generates video traffic at a peak rate of 200kbps. The size of ON-OFF `AF11` packets is equivalent to 576 bytes;

(iii) Finally, the prioritized data traffic is marked with `AF21` value [148]. Both `AF21` and `BE` traffic (the background traffic) are characterized by a Poisson process (in which the inter-arrival of packets follows an exponential distribution). `AF21` and `BE` sources generate traffic at 350Kbps, and 500Kbps, respectively. The size of the packets generated by the data connections in these simulations is 1000 bytes.

The size of the probes spawned by the IRCs using the Pseudo-Random Poisson process were correlated with the Class of Service (CoS) being controlled. As we handled three different CoSs (voice, video and prioritized data), we used three different types of probes that matched with the corresponding traffic DSCPs. $N = 8$ probes are sent in each window t.

During the evaluations we configured the following default set of parameters for SRCs:

(i) $Q = 20$ (Scale of the cost M);

(ii) $\Delta = 1.05$ (Conservativeness factor);

(iii) An adaptive period (heuristically chosen) for the quantization step equal to five times the size of the window t (i.e., $5t$).

These default values were tuned after numerous simulation runs to obtain a good trade-off between speed of reaction (selfishness), and stability. All these parameters can definitely be configured (tuned) by the multihomed stub AS customers.

Evaluation objectives. The main objective is to assess the benefits of the sociable IRC model against the conventional IRC model and default BGP routing, in terms of overall stability and traffic performance. This is first shown by evaluating for each model how far IRCs enhance to the overall network stability under variable QoS dynamics, and when different SLSs should be used between remote multi-homed stub domains. As in previous assessments, the total number of path switches needed to meet the QoS constraints within the different SLSs are used as a performance indicator. Following this, we evaluate the overall traffic performance by measuring (i) the end-to-end OWD (or latency); and (ii) the traffic transfer efficiency for each CoS. Again, as in Chapter 3, the efficiency for each CoS is defined by $Ef_{nsj} = \frac{C_{nj}}{C_{sj}}$, where C_{nj} is the throughput at a given destination n for the traffic of class j, and C_{sj} is the corresponding throughput sent by the source domain s for the traffic of class j.

4.7.2 Evaluation of Overall Traffic Performance

Figures 4.6 and 4.7 contrast the end-to-end traffic performance for the four studied scenarios, and the two SLSs, under stressful traffic conditions, i.e., with high load in some links at the edge of the network. We notice that both Figures 4.6 and 4.7 were obtained using the normalized threshold $R_{th} = 1$, which is a sufficiently demanding configurable value. Our aim was to generate some bottlenecks to put pressure on the competition between the IRCs for less-loaded paths, and then observe to what extent they managed to comply with the rather challenging SLSs.

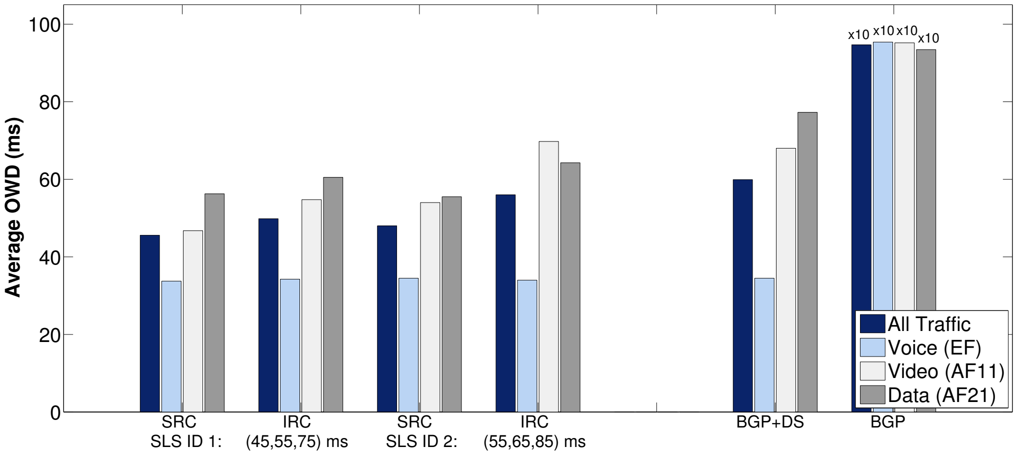


Figure 4.6: Average latency for all the scenarios and the SLSs for each class of traffic.

In terms of latency, Figure 4.6 shows that the first scenario, i.e., the pure BGP model, gives the worst performance. In this case only BGP is running (all the traffic is BE), and clearly BGP is unable to actively avoid the most congested paths. The way traffic will flow in this scenario completely depends on the state of the local routing

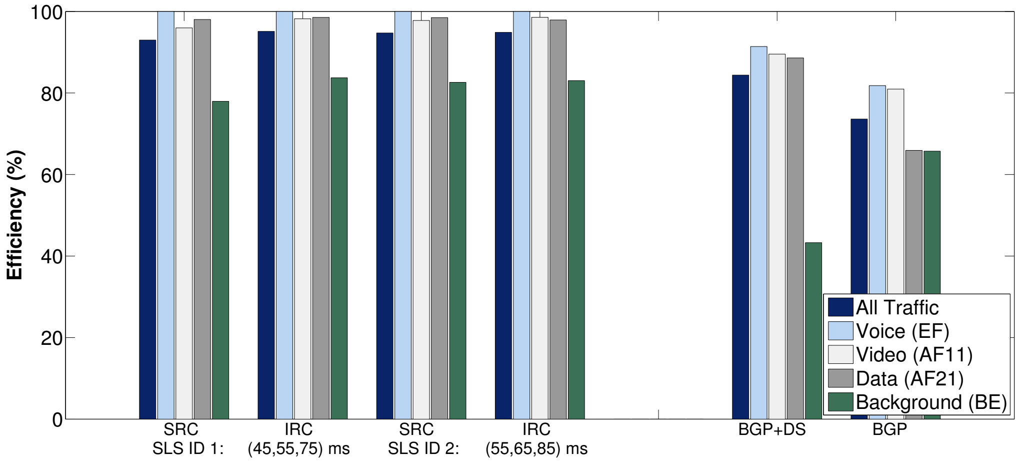


Figure 4.7: Efficiency for all the scenarios and the SLSs for each class of traffic.

policies, and hence, it depends on how the routes are being advertised by BGP. Thus, the only way to improve end-to-end traffic performance in this case is by means of manual intervention. The network administrators have to start tuning their BGP router configurations on a trial-and-error basis. Figure 4.6 clearly shows that under stressful traffic conditions, on average, BGP is generally incapable of complying with the SLSs in Table 4.1. In this case, the average latency is extremely high (over 900ms) because of the contribution of the most congested links to the OWD. Moreover, Figure 4.7 shows that the total number of losses is unacceptably high in this case (for instance, more than 20% for voice and video traffic, and $\simeq$ 40% for data and background traffic).

In the second scenario, i.e. BGP combined with DiffServ at the edge of the network, the latency is improved by more than 10 times. However, the traffic throughput efficiency in Figure 4.7 visibly shows that this is achieved by strongly penalizing BE traffic, which is extremely undesirable. Additionally, the SLS IDs 1 and 2 for video and prioritized data, and the SLS ID 1 for prioritized data, are on the whole still unfulfilled (given it was measured an average latency of 68ms and 77ms, respectively, for video and prioritized traffic). Thus, this scenario does not constitute a definitive solution to the problem of improving end-to-end performance.

The third scenario adds conventional IRC sets to the previous scenario. Figures 4.6 and 4.7 clearly show that the inclusion of the IRCs adds significant improvements, both, in terms of latency and efficiency. The IRCs are able to bypass the most congested links without needing to penalize the BE traffic in a significant way, while also reducing the losses in all the remaining classes of traffic.

Despite these improvements, the conventional IRCs show signs of two important weaknesses. First, they are generally incapable of complying with some particular combinations of SLSs, like SLS ID 2 for video traffic (given it was measured an average latency of 70ms for this traffic). The explanation for this is that the selfishness of the

IRCs, together with the interference caused by the various sets of IRCs operating in the network at the same time, generates a large number of path switches, especially under stressful traffic conditions. This leads to oscillatory behaviors in some IRCs for the CoSs mapped to AF, which results in non-compliance with the SLS during these unsteady states. As a result, the conventional IRCs might not be able to comply with a certain combination of SLSs. The second and most serious weakness of the conventional IRCs clearly resides in the number of path switches needed to fulfill all the SLSs. This disadvantage can be seen at a glance in Figure 4.8, which will analyzed in Section 4.7.3.

The highest performance benefits under stressful traffic conditions were observed in the fourth scenario, i.e. BGP combined with DiffServ and the sociable IRC strategy. The SRCs are able to route around congested links, and outperform the achievements of the conventional IRCs. In particular, in the cases of video and prioritized data traffic in Figure 4.6, the SRCs clearly improve the end-to-end average latency of traffic by respectively 14% and 7% for SLS ID 1, and 23% and 14% for SLS ID 2, compared with the conventional IRCs; giving an average latency improvement for all traffic by 14.3% (8ms better) for SLS ID 1 and 8.5% (4ms better) for SLS ID 2. It should be pointed out that in this scenario, all the SLSs are fulfilled for all the traffic classes involved. Thus, the ability of the SRCs to adjust the performance metric M based on what they learning from the mid and long-term network conditions, clearly improves the route optimization decisions that the IRCs are able to make.

Another important fact is that the ability of the conventional IRCs to reduce the latency of delay-sensitive applications, does not lessen the traffic transfer efficiency as shown in Figure 4.7. In fact, both kinds of IRCs provide almost the same degree of traffic transfer efficiency. From Figures 4.7 and 4.8 it is evident that, even though, in some cases, the SRCs are prone to present slightly more losses than the conventional ones, the improvements in terms of overall network stability are overwhelming, as shown in Figure 4.7.

Finally, from Figure 4.7 it is clear that the efficiency of the background traffic (BE traffic) is greatly improved by the IRCs, when compared with the first two scenarios. Nonetheless, it still registered some losses which can apparently be explained by the actions of the DiffServ mechanisms.

4.7.3 Evaluation of Overall Network Stability

The main goal of the IRCs is to provide steady traffic patterns, while complying with all the SLSs. However, this objective may lead to network instabilities, especially, when a large number of IRCs are present on the network and frequent selfish routing optimizations are made. Hence, the impact of these optimizations on the network must to be reduced as much as possible. During our experiments we observed that, although both types of IRCs are able to bypass congested network segments and avoid

SLS violations, the conventional IRCs display an oscillatory routing behavior. On the other hand, these anomalies are not apparent when sociable IRCs are in use.

The contribution made by the sociable IRC strategy to the overall stability of the network can be evaluated by the total number of path switches that occur in all the multihomed stub ASes. This is assessed for different values of the degree of conservativeness R_{th}, as illustrated in Figure 4.8. This figure shows that the number of path switches introduced by the sociable IRCs is clearly much smaller than those introduced by the conventional IRCs. With $R_{th} = 1$, the total number of path switches is reduced nearly 3 and 7 times, respectively, for SLS ID 1 and SLS ID 2; and with $R_{th} = 2$ and $R_{th} = 3$ nearly 3-4 times (depending on the SLS).

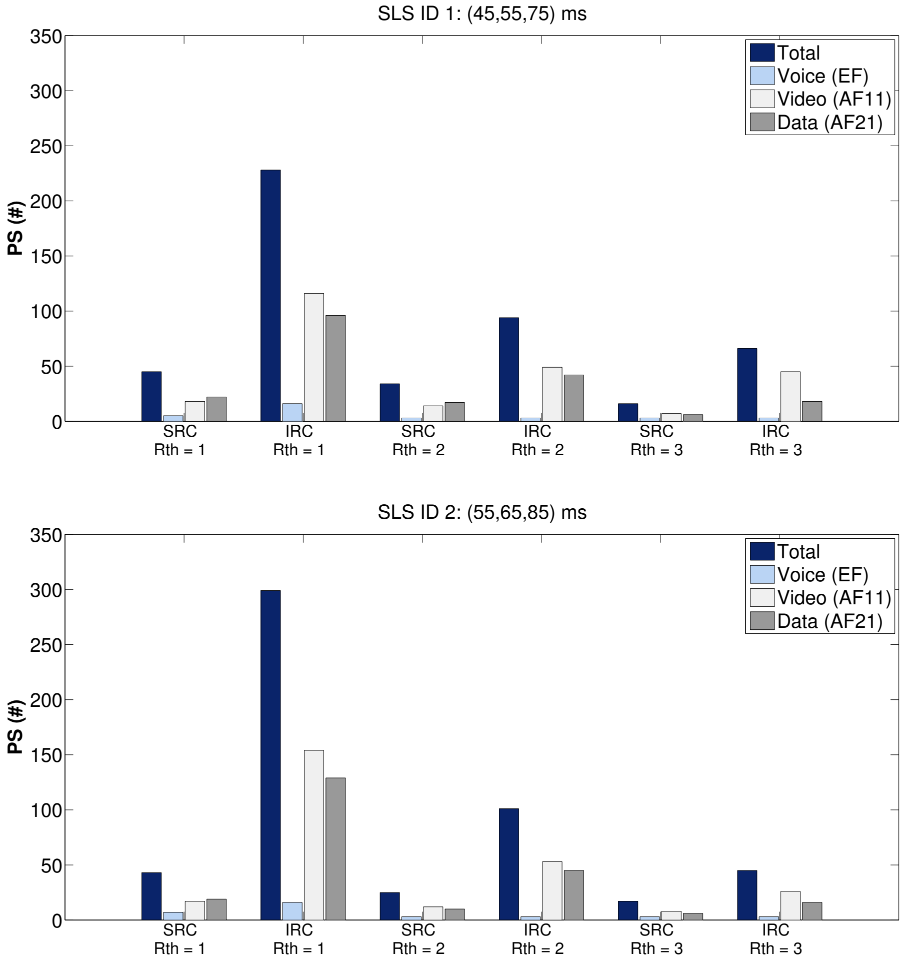


Figure 4.8: Number of path switches for different R_{th} and the SLSs for each class of traffic.

Moreover, the configuration of different thresholds R_{th} in the IRCs helps to dramatically reduce the number of path switches and thus make an additional improvement to the overall stability of the network. However, it is also clear that the sociable IRC strategy is less sensitive to this parameter because the SRCs have the ability to adapt actively their proactivity.

Finally, from Figures 4.9 and 4.10 it is visible that, even through, the SRCs perform less path switches than the conventional IRCs, SRCs are able to improve the end-to-end latency of traffic irrespectively of the R_{th} values because of their proactivity, while the degree of the traffic efficiency is almost the same for all traffic and the SLSs analyzed. The overall improvement in the latency is 6.7% (+3ms better) and 9.5% (aprox. 5ms better) with $R_{th} = 2$, for SLS ID 1 and SLS ID 2, respectively; and for the same SLSs, 8.5% (4ms better) and 9.7% (aprox. 5ms better) with $R_{th} = 3$. In short, this corroborates the idea that performing a reduced number of paths would be enough to improve the performance of traffic.

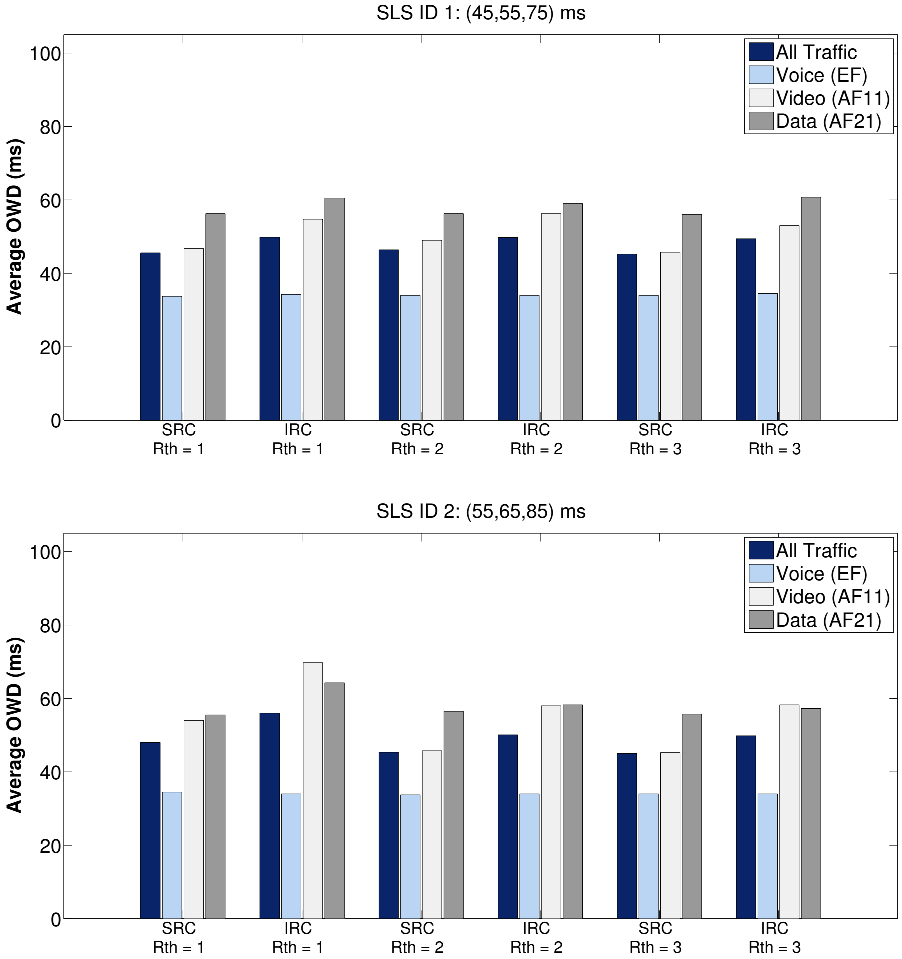


Figure 4.9: Average latency for different R_{th} and the SLSs for each class of traffic.

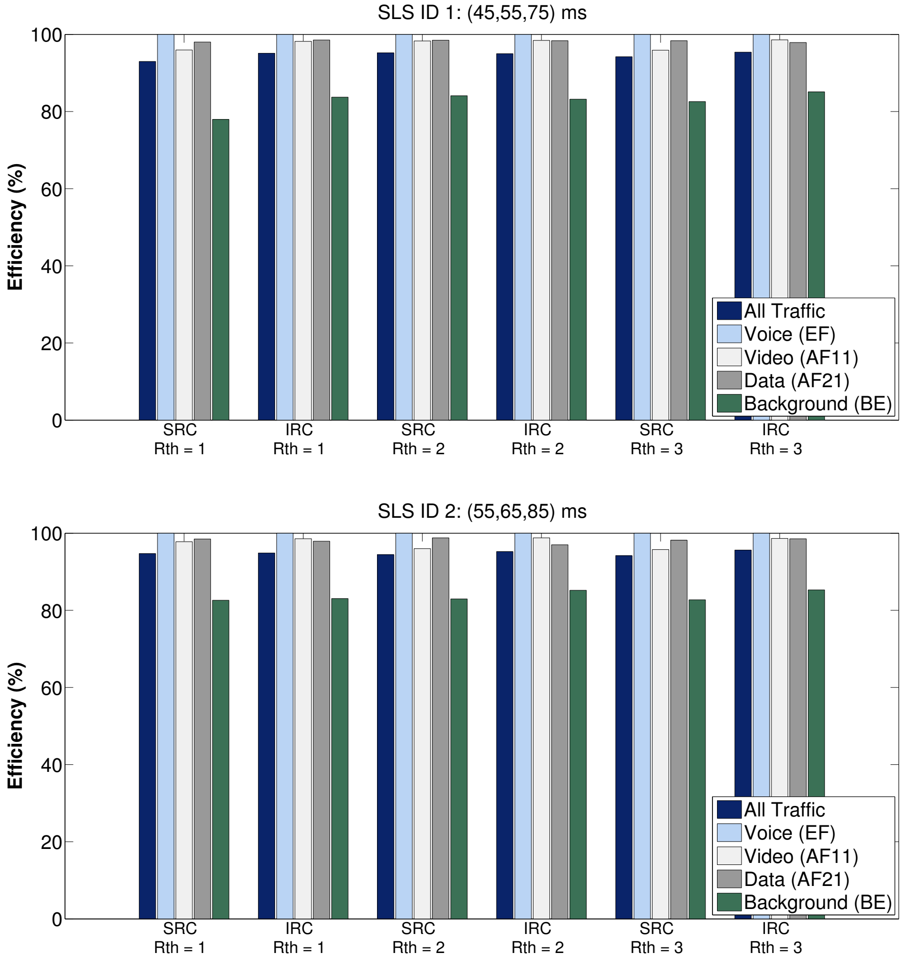


Figure 4.10: Efficiency for different R_{th} and the SLSs for each class of traffic

4.8 Summary

In this chapter, we studied the strengths and weaknesses of randomized Intelligent Route Control (IRC) techniques in a competitive environment. We provided a way to blend IRC with a sociable route control (SRC) strategy, where by sociable, we mean a route control strategy that explicitly takes account of the potential implications of its decisions on the performance of the network and which has the ability to adaptively restrain its intrinsic selfishness, in accordance with the network conditions.

We have shown that in a competitive scenario, our SRC strategy is capable of drastically reducing the potential interference between controllers without penalizing the end-to-end traffic performance. This makes SRC more scalable and promising than pure randomization, because of the widespread proliferation of IRC systems in the Internet.

SRC strategies, like the one described in this chapter, also have a number of practical advantages; for example, they do not require any kind of coordination between the competing IRC boxes; and they can be supported by a lightweight software implementation based on well-known filtering techniques, with no additional requirements other than a software upgrade of existing IRC systems.

Among the issues in this area that remain unresolved, the most important is the lack of a stochastic model to characterize the distribution of path switches in a competitive environment. However, studies like [89] have shown that randomized techniques are effective in desynchronizing certain route controllers when their measurement windows are sufficiently overlapped, although, they cannot guarantee stability. Only after the distribution of path switches has been characterized, will it be possible to formally study the stability of competitive IRC.

Furthermore, even though the proposals and results described here, apply to the optimization of voice traffic, the concept of blending IRC with an SRC strategy is general in scope which means that this work can be extended to control other kinds of traffic flows concurrently, as well as taking account other performance metrics apart from the round-trip time.

Chapter 5

An ISP-friendly IRC Cooperative Framework

In Chapters 3 and 4, there was a discussion of the design of an intelligent route control system, and a proposal for social route control model to improve the stability of an intelligent route control system. This chapter identifies the problem of the interaction between intelligent route control and inter-domain traffic engineering, and proposes a joint design based on a model for sharing information between both systems to obviate the potential ill effects of this interaction.

Bibliographical Notes. Parts of this chapter have been published in [103, 161]. The preliminaries of this work has been published in [162–164]. At the time of the submission of the thesis, there is also one working-paper under submission for publication. This work had the collaboration of Manuel Machado, a member of the Communications and Telematics Research Group at the University of Coimbra, with whom I discussed my proposals, and who developed the inter-domain traffic engineering algorithms that were employed in the evaluations.

5.1 Introduction

Middleboxes like Intelligent Route Control (IRC) and Traffic Engineering (TE) boxes have become an important part of today's networks. Throughout this thesis, especially in Chapters 3 and 4, we have devoted attention to the design of IRC boxes, and discussed and documented the potential benefits of this approach. However, there is a danger that the interaction that may occur between the IRC and TE boxes can jeopardize this undertaking. A proposal to minimize the effects of disputes between these middleboxes in the routing control is outlined in this chapter.

By employing IRC boxes, multihomed stub Autonomous Systems (AS), also described in a more general way as stub networks, can achieve several traffic goals, such as performance, quality of service and reliability, without adding extra complexity to inter-domain routing [89]. To undertake this, IRCs employ active probing to capture the perceived quality offered by the paths, and a path switching algorithm to select the best paths and adapt the inter-domain routing policies accordingly. However, many transit ASs are using traffic engineering boxes to provide better inter-domain routing and to return the network to its optimal performance [67]. One common optimization problem that needs to be tackled by transit ASs is that of the Egress Router Selection (ERS) [165]. This means that the transit ASs generally face the problem of *how to assign the traffic demands to multiple egress points, to ensure that the transit objectives of the the transit network are met*?

Unfortunately, selfishness is a common characteristic of both middleboxes since the paths (or transit providers) are greedily selected by each box, which, if there is a misalignment of traffic goals, can cause traffic performance degradation. An IRC seeks to select low latency or high bandwidth paths, whereas a TE box seeks to minimize either the whole Maximum egress Link Utilization (mMLU), or the monetary transit costs [166].

The IRC-TE interaction can be viewed as an iterative process. Through this process, the actions of each middlebox modify the inputs of the other. The routing changes achieved by the IRCs to improve traffic performance, interact with TE by changing the Traffic Demands (TD) in the transit networks. When the overall change in the TD is significant, the optimal routing computed by TE is no longer valid and thus congestion can occur on intra or inter-domain links of their networks. This problem arises from the difficulty of accurately tracking the traffic variations caused by rapid route changes [118].

In turn, TE changes the end-to-end traffic quality by periodically adapting the routing to the variations in traffic demands to return the network to an optimal regime. In response, the IRCs adapt local routing to the resulting path quality changes, and in this way the first interaction can be repeated. As a result, uncoordinated routing changes that attempt to overcome performance degradation, can produce cycles of influence between both boxes. A major challenge of the research on Traffic Engineering and Internet Routing architectures is, thus, how to handle this conflict.

Most of the significant related work relied on a game-theoretic-based analysis. A precursor study in [167] compared the global Internet performance obtained through selfish routing to the optimum level achieved through global routing coordination. In particular, the authors proved *that if the latency of each edge is a linear function of its congestion, then the total latency of the routes chosen by selfish network users is at most* $4/3$ *times the minimum possible total latency.* However, although this study can give us some insights into the mechanisms of interaction, it is not clear how to apply this theoretical model to our problem. Later on, in [90] the authors found that selfish overlays can significantly reduce the effectiveness of intra-domain TE (i.e., the

MultiProtocol Label Switching (MPLS) optimization [168]), but employing a somewhat more realistic model through the use of measured Internet Service Provider (ISP) topologies. Influenced by the study carried out in [83] the authors constructed a model for the interaction between the overlays and intra-domain TE. However, it is still unclear *if* and *how* these penalties can also occur in inter-domain environments.

This chapter is, thus, divided into two main parts. In the first part, we seek to answer two key open questions: *Does the ability of stubs and ISPs to select routes on the basis of a greedy strategy lead to the best routing of IP packets in the Internet? Why and how do the IRC and TE boxes interact?* For this reason, it is necessary to analyze this kind of interaction through intuitive clearly-defined descriptions and a simulation model. We believe that understanding the potential interactions between IRC and TE can provide some guidelines for future work in the area, such as research on cooperative TE or negotiation-based routing [91, 92, 169].

In the second part of this chapter, a novel ISP-friendly intelligent route control COOPerative framework (COOP), is proposed to handle this conflict. COOP is based on a cooperative strategy between direct neighbors or nearest multi-hop neighbors, and employs a feedback-based method in which IRC and TE middleboxes cooperate to restrict traffic demand fluctuations over the transit networks of the ISPs, while ensuring the network operation regime approaches the optimum. By means of extensive evaluations based on a realistic network and a data trace, we show that COOP can achieve synergistic interactions between intelligent route control and inter-domain traffic engineering.

The contributions of this chapter are therefore:

(i) An analysis of the interaction between IRC and TE middleboxes. We show that the effectiveness of inter-domain traffic engineering can be impaired by routing changes performed by intelligent route control. However, the overall stability can benefit from combining *implicit* sociable intelligent route control with inter-domain traffic engineering;

(ii) An ISP-friendly IRC COOPerative framework (COOP). We show that by integrating an *explicit* cooperative scheme in IRC and TE boxes, it is possible to outperform the current scenario of standalone and selfish IRC and TE boxes. In addition to the advantages provided by IRCs (i.e., to protect stubs ASs from performance and QoS violations against end-to-end performance and quality bounds), with this novel approach, it is possible to improve the predictability of traffic demands and mitigate the effects of selfish costs, such as end-to-end traffic performance losses and TE performance losses. The results show that the COOP framework can produce synergistic interactions between intelligent route control and traffic engineering;

(iii) The construction of a realistic simulation environment to test and assess the feasibility of cooperative strategies in competitive environments (not confined to COOP). This simulation environment enables large scale simulations, and is based on real traffic demand matrices and on a realistic Internet topology that is inferred by means of the characterization of traffic demands and real BGP (Border Gateway Protocol) table dumps. It supports most of the existing IRC algorithms and a genetic inter-domain traffic engineering algorithm. In view of this, the network optimizations that are based on these assumptions can achieve realistic solutions.

The rest of this chapter is divided as follows. Section 5.2 gives some background on inter-domain traffic engineering and a further explanation of the causes of the IRC-TE interaction problem by providing illustrative descriptions. Section 5.3 shows and analyses the presence of IRC-TE interactions through a simulated network. Section 5.4 provides a detailed step-by-step design of COOP, including its architectural principles, components and algorithms. Evaluations are examined in Section 5.5 to show the feasibility of COOP. Section 5.6 concludes the chapter.

5.2 Background and a Description of the Interactions

This section sets out by providing some background information on inter-domain traffic engineering. Following this, it provides a further explanation of the causes for the interactions between Intelligent Route Control (IRC) and Traffic Engineering (TE), making use of illustrative descriptions.

5.2.1 Inter-domain Traffic Engineering Model

This subsection describes the inter-domain traffic engineering model as a routing optimization problem, and includes the main definitions and the heuristics employed in this thesis to address the problem.

5.2.1.1 Definitions

The goal of inter-domain Traffic Engineering (TE) is to perform routing operations that ensure the optimal outgoing traffic performance of a transit network. In this chapter, the traffic objectives are regarded as the minimization of the Maximum Link Utilization (min-MLU) of the egress links of a transit network or the Load-Balancing (LB) among these links.

Figure 5.1 shows an example of a transit network being optimized by an inter-domain TE box. This network is represented by a set of ingress points I and a set of egress points E, where after each routing (re)optimization the incoming flows/aggregates (`in`) at the ingress point links are (re)assigned to egress point links (`out`) so that an optimal distribution or mixture of outgoing traffic is achieved.

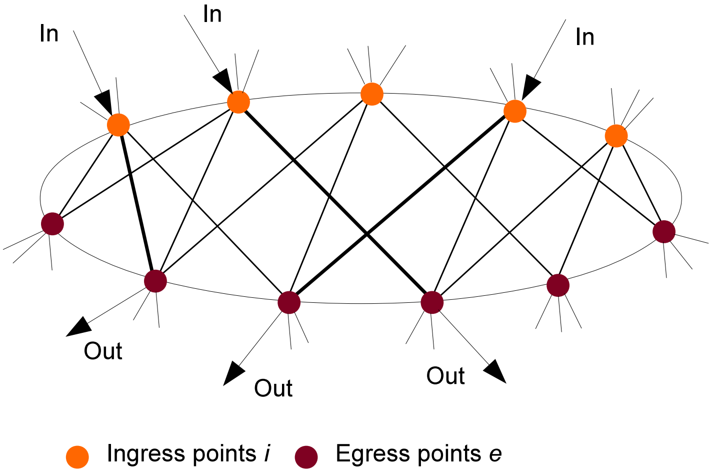


Figure 5.1: ISP network model.

The inputs to the TE algorithm consist of the predicted Traffic Demands (TD) over the transit network, the available egress point choices for each reachable prefix destination and the egress point capacities.

The predicted TDs are represented by the matrix $D = \{d_{ip} \mid i \in I, p \in P\}$, where each entry d_{ip} is the demand for the ingress point i - destination prefix p pair. If each IRC $h \in H$ has T data transfers at rates x_{hp} to distribute over the ingress points I, then each entry d_{ip} is defined as in Equation (5.1), where $r_{hip} \in \{0, 1\}$ is an indicator function to select whether the route from IRC h to prefix p via ingress point i is active (i.e., True (1), False(0)).

$$d_{ip} = \sum_{h} \sum_{p} x_{hp}.r_{hip} \tag{5.1}$$

The set of available egress choices for each reachable prefix p is defined by the topological constraints of the transit network. That is, the set of available egress choices is determined by the routes provided by each egress BGP router, marking the egress points so that the BGP routers have no route to reach this prefix. Thus, this set is represented by a matrix $A = \{a_{ep} \mid e \in E, p \in P\}$, where each entry $a_{ep} \in \{0, 1\}$ indicates whether there is a route to p from the egress point e.

The set of egress point capacities are represented as C, where each entry $c(e)$ is the capacity at the egress point e.

Inter-domain routing is represented by ϵ, where each entry $\epsilon_{iep} \in \{0, 1\}$ is an indicator function that tells whether the d_{ip} is assigned to the egress point e.

The output of the inter-domain traffic engineering process is the set of optimal routes – the optimal routing – that achieves an optimal mixture of outgoing traffic, in compliance with the traffic objectives for the network. Subsequently, these results are translated to a careful tuning of BGP routes attributes. For the interested reader, the set of techniques (e.g., `LOCAL-PREFs` tuning) that can be used for egress traffic control are described in Chapter 2 – Subsection 2.1.5 – and the references therein.

The inter-domain TE problem that has to be addressed is known as the Egress Router Selection (ERS), and can be defined as *how to assign each entry of traffic demands d_{ip} to an egress point e, so as to optimize a certain traffic objective.* In this chapter, two typical objectives – the minimization of the whole MLU (mMLU) and Load-Balancing (LB) –, have been encoded in the ERS problem to ensure that the egress link utilizations are at the lowest levels and thus can minimize congestion, giving us two ERS versions (see Definitions 5.2 and 5.3). Other objectives can be also encoded in ERS (e.g., min-cost routing or maximum business profit) [166].

Definition 5.1: *The link utilization of e for a routing ϵ is defined as the traffic to capacity ratio, as is shown in Equation (5.2).*

$$U_e = \sum_i \sum_p \frac{\epsilon_{iep} d_{ip}}{c(e)}, \epsilon \textit{ is a routing.} \tag{5.2}$$

Definition 5.2: ***Traffic Objective 1 - Minimizing the Maximum Link Utilization (mMLU).*** *An optimal inter-domain routing for a given D is the routing that minimizes the maximum link utilization (mMLU), as is shown in Equation (5.3), where OU is the optimal utilization.*

$$OU \quad = \quad min\, max\, U_e, \forall e \in E. \tag{5.3}$$

Definition 5.3: ***Traffic Objective 2 - Load-Balancing (LB).*** *Load-balancing is closer to the mMLU objective. Both objectives can be used interchangeably, but load-balancing is a more sophisticated and stringent traffic objective. In this case, the ERS is defined as is shown in Equation (5.4).*

$$min\, |U_i - U_j|, \forall i, j \in E \tag{5.4}$$

Both objectives are subject to constraints (5.5) and (5.6). The capacity constraint (5.5) ensures that the total resource requirements of the traffic flows assigned to each egress point do not exceed the available (or contracted) capacity. The assignment constraint (5.6) guarantees that each traffic flow is assigned to exactly one egress point e.

$$\sum_i \sum_p \epsilon_{iep} d_{ip} \leq c(e), \forall e \in E, \tag{5.5}$$

$$with \sum_e \epsilon_{iep} = 1, \forall i \in I \tag{5.6}$$

Definition 5.4: *The performance ratio of a routing ϵ for a given D is defined as the ratio between the current MLU and its optimum OU, as is shown in Equation (5.7).*

$$P(\epsilon, D) = \frac{max\, U_e}{OU}, \forall e \in E,\ \epsilon\ is\ a\ routing. \tag{5.7}$$

The performance ratio $P(\epsilon, D)$ measures how far a routing ϵ is from optimal for a given traffic demands matrix D. $P(\epsilon, D) = 1$ implies that the routing ϵ is optimal. In contrast, higher values of P (> 1) imply that the routing ϵ is farther away from the optimal.

5.2.1.2 Heuristics for Backbone Traffic Engineering

As described so far, the goal of the ERS optimization problem is to minimize traffic objectives, such as mMLU and Load-Balancing (LB), or simultaneous multiple objectives, such as both mMLU and transit cost objectives at the same time. Since in the analysis of the interaction between TE and IRC, just one objective is considered at a time (see Section 5.2.2), the algorithm chosen to address the ERS is based on a genetic single-objective version of [106]. However, this is well-suited to being extended to the case of multi-objective optimization [170], which contrasts with the conventional optimization techniques, such as the simplex method that are only well-suited to single-objective optimization [171].

The algorithm, belonging to the class of evolution strategies that are used in optimization, resembles the process of biological evolution, where each individual is described by its genetic code, called a chromosome. In turn, each chromosome is composed of individual genes. In the problem in hand, a gene is the assignment of a single traffic flow to an egress point of the ISP, and an individual – a chromosome – is a potential solution.

The basic steps of algorithm are shown in Algorithm 5.1. This starts with the creation of the initial generation, where the individuals are created at random. Following this, there is an evaluative step based on the proposed objective function (5.3) or (5.4). After that, and for a number of generations, a new generation of children is created and compared with the corresponding generation of parents. As a result of this comparison, the best elements will compose the next generation of parents. The ranking step is done as proposed in [172]. The last of the generations is the TE solution that is sought. It is worth mentioning that this algorithm has a time complexity of $O(N^2)$, where N is the size of the population.

Algoritmo 5.1 Basic steps of the genetic inter-domain TE algorithm [106].

```
Create the initial parent generation;
Evaluate the generation;
for a number of generations do
    Create the child generation;
    Evaluate both generations together;
    Rank both generations together;
    Replace worst parents with better children.
end for
```

5.2.2 Interactions between IRC and TE Middle-boxes

Figure 5.2 shows a comprehensible model of the potential interaction between the IRC and TE middleboxes. By using this model and an example, this section seeks to answer the question: *Why and how do these traffic control boxes interact?*

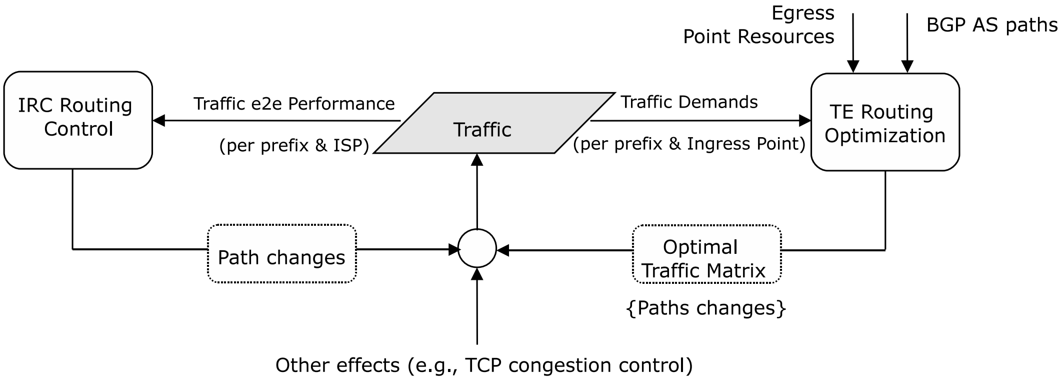


Figure 5.2: Model of the actions of the IRC and TE boxes over the traffic.

The interactions between stub ASs and ISPs, that is between IRC and TE middleboxes, arise from the effects of changing the inter-domain routing at one party over the traffic (or network) of the other party.

In a first interaction, routing changes at stub ASs achieved by IRC boxes interact with ISPs by changing the traffic demands over ISP networks. In the Internet of Figure 5.3, consider that each dual-homed stub, `AS1` and `AS2`, have exactly 6 data transfers (i.e., $T = 6$) for prefixes $\{p_1, p_2, p_3, p_4, p_5, p_6\}$ at a fixed rate $x(.,.)$ of 10 units of traffic. Then, suppose that somewhere in the first time slot ($t = 0$) `AS1` activates the path `ISP2→ISP1→AS3` for the prefixes p_{1to4} and the path `ISP3→ISP1→AS3` for the prefixes p_{5to6}. In turn, `AS2` activates the path `ISP3→ISP1→AS3` for the prefixes p_{1to2} and the path `ISP5→ISP4→ISP1→AS3` for the prefixes p_{3to6}. As a result, these path selections correspond to an initial traffic demands matrix $D_{(1,0)}$ over the target ISP, `ISP1` (input for the first TE cycle (0)), as is shown in the left-hand of Table 5.1. The characteristic of this initial traffic demands matrix is that all traffic units are evenly distributed over the ingress points of `ISP1`. Next, `ISP1` optimizes its outgoing traffic over `AS3`, assuming that this traffic demands matrix is the input of the TE process.

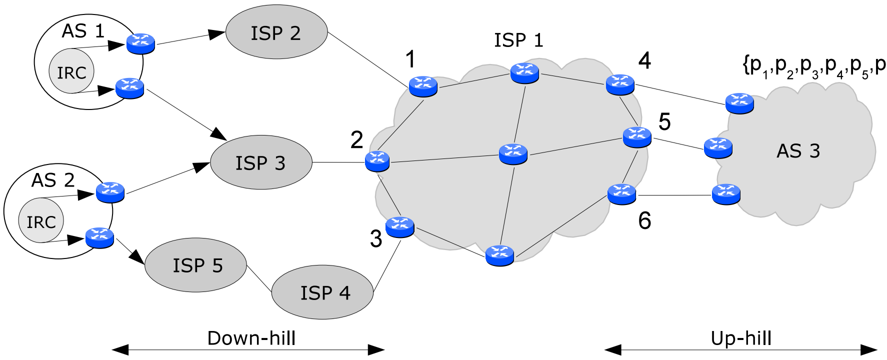


Figure 5.3: An Internet example.

After this, imagine that at an instant of time t', somewhere in the second slot of time (i.e., $t' \in]t, t+1], t = 0$, where (1) refers, in the abstract, to the length of one TE optimization period), IRC boxes from `AS1` and `AS2` decide to move some traffic between ISPs based on actual latency measures. For instance, suppose that both boxes found that the best paths for prefixes p_{3-4} are throughout `ISP3`. Accordingly, both `AS1` and `AS2` activate the path `ISP3→ISP1→AS3` for prefixes p_3 and p_4. As a result, the (real) traffic demands matrix of `ISP1` changes to the matrix shown in the right-hand of Table 5.1, which corresponds to a situation of imbalanced traffic over ingress points of `ISP1`.

This simple example clearly shows that in so far as IRCs distribute traffic across ISPs, several entries of traffic demands matrices over ISPs might be changed. As a result, if the overall change in the traffic demands matrices is far beyond a given tolerated margin, it means that the initial conditions considered by the ISP TE middlebox during the last optimization cycle have been broken (e.g., if $|D_{(1,t')}| > |D_{(1,t)}| + \sigma_1^2$, where σ_1^2 is the tolerated fluctuation in the traffic demands over `ISP1`). In such a case, in practice the outcome is that it makes the performance of the ISP networks deviate

Table 5.1: Illustration of traffic demands changes over ISP1.

	$t = 0$						$t^{'} \in]0, 1]$					
Destinations →	p_1	p_2	p_3	p_4	p_5	p_6	p_1	p_2	p_3	p_4	p_5	p_6
$D_{(1,t)}(1, .)$	10	10	10	10			10	10				
$D_{(1,t)}(2, .)$	10	10			10	10	10	10	20	20	10	10
$D_{(1,t)}(3, .)$			10	10	10	10					10	10

significantly from the optimum regime. In short, this example shows that intelligent route control can reduce the effectiveness of inter-domain TE, as well as imposing an additional burden on the TE process to return the network again to the optimal regime.

In turn, in the second interaction, after finding a new solution for the ERS problem, the ISPs interact with stub ASs by changing the end-to-end performance of inter-domain traffic leaving these networks. In effect, ISPs might decide to shift some inter-domain traffic to other egress points. As a result, a new set of paths is then provided to stub ASs, which might have different performance characteristics (e.g., latency). In response, IRCs at stub ASs adapt their inter-domain routing to these changes, which allows the first interaction to be repeated again and again.

5.3 Analysis of the IRC-TE Interaction Problem

This section examines the results of a pool of simulation tests performed to assess the interaction between intelligent route control and inter-domain traffic engineering.

5.3.1 Methodology

In the simulation tests, it is evaluated the performance of conventional Intelligent Route Control (IRC), Social intelligent Route Control (SRC), default Border Gateway Protocol (BGP) routing and BGP routing with inter-domain TE for either the mMLU objective or the LB objective – abbreviated by TE.mMLU and TE.LB – for the possible combinations that depend on whether the IRC or TE mechanisms have been switched ON/OFF, giving us eight different simulation configurations.

The simulation tests were carried out by means of the J-Sim simulator [145]. The IRCs were developed on top of the BGP implementation available in this platform. In turn, the genetic inter-domain TE algorithms were coded in MATLAB, a high-level language [173].

To emulate the iterative process described in Subsection 5.2.2, we implemented a coordination and communication mechanism between both environments. At the beginning of each TE round, the J-Sim environment provides the TE middlebox with the estimated traffic demands and the routing data collected at the target ISP. In turn, the TE algorithm takes these inputs, and computes the optimal routing according to the mMLU or LB objective. Finally, the required routing changes are communicated to J-Sim which applies them at ISP borders routers and proceeds with the simulation. This process is repeated for each TE round. In the evaluations, the IRC middleboxes play in a timescale that is approximately 50 times smaller than the one of the inter-domain TE middlebox.

5.3.1.1 Simulation Setup

The AS-level topology used in the experiments is similar to the example given in Figure 5.3. The simulated network represents an Internet core composed of 100 Tier-2 ISPs divided into two portions: down-hill and up-hill. Both topology portions were built using BRITE [151] in accordance with the Waxman's model with (α,β) set to $(0.15, 0.2)$, and a ratio of ASs to links of 1:3. Then, both portions were interconnected by the ISP target of the TE optimizations, a full-meshed Tier-1 ISP composed of 8 Points-of-Presence (POPs) with a ratio of POPs to links of 1:3.

During the tests, 300 IRC sources located at down-hill portion send traffic to popular prefixes located at up-hill portion through the target ISP. Each traffic aggregate is composed of a variable number of multiplexed Pareto flows – Voice over IP flows –, according to a Zipf distribution [174]. To be more precise, a Weibull distribution with shape 0.4. The flow arrivals are described by a Poisson process. We ensured that the overall traffic load can provide the network with the most cost-effective regime if there is a perfect traffic distribution across the network. In other words, we do not simulate at a load level that causes the network to being considerably over or under-provisioned to avoid absurd situations (e.g., stable path selections or persistent congestion).

Finally, we configure IRCs, to observe the ITU-T recommendation G.114 to maintain high voice quality [109], which suggests a one-way delay bound of 150ms, giving a Round-Trip Time (RTT) bound of 300ms.

5.3.1.2 Performance Metrics and Objectives

We use three metrics to evaluate the performance of IRC and TE boxes:

(i) **Number of path shifts:** this metric is obtained by adding the number of path changes that is required for the performance of traffic flows to meet the RTT bound;

(ii) **Latency:** the end-to-end latency (or delay) metric is defined as the average of RTTs measured for traffic;

(iii) **Traffic engineering performance ratio:** the TE performance ratio $P(\epsilon_t, D_{t'})$ of an inter-domain routing ϵ_t at ISP and current traffic demands $D_{t'}$ over the ISP is defined as in Section 5.2 (i.e., $P(\epsilon_t, D_{t'}) = \frac{L(\epsilon_t, D_{t'})}{OL(\epsilon_t, D_t)}$, $t \neq t'$, where L is the maximum utilization or load-balancing across the egress links of the target ISP for a given $D_{t'}$ and ϵ_t, and OL is the optimal utilization or load-balancing for a given D_t and ϵ_t).

In the light of this, three main objectives for this simulation study are examined. The first is to evaluate the potential impact of inter-domain TE on the IRC stability. To undertake this, we compare the average number of path shifts performed by IRC when inter-domain TE is switched ON/OFF. The second objective is to assess the implications of the interactions between IRC and inter-domain TE on the traffic performance. This is carried out by comparing the latency for all simulation configurations. The last objective is to evaluate the potential impact of IRC on the inter-domain TE performance. This was achieved by comparing the inter-domain TE performance ratios for mMLU and LB objectives for all the simulation configurations.

5.3.2 Results

Figures 5.4, 5.5 and 5.6 show the number of path shifts, the cumulative number of path shifts and the latency, that are registered and measured at the end of each TE round respectively. These results reveal some important observations, which are discussed below.

First, when the both intelligent route control schemes are compared, it is evident that there is an *overall stability benefit from combining sociable IRC and Traffic Engineering.* In effect, this combination needs no more than roughly 32% (mMLU) or 37% (LB) of the total number of path shifts required by the combination of the conventional IRC model with TE to meet the RTT bound. These results show a similar pattern to the found in the previous study in [152]. That is, by using a SRC strategy the oscillations can, on average, be drastically reduced, while maintaining a similar traffic performance.

Second, the growth of the number of path shifts is almost linear, except when the TE box is switched OFF. This is expected because the overall load of the network is moderate and flows arrivals are poissonian. When TE is switched OFF, the higher latency observed for traffic reveals that BGP default routing is unable to provide enough traffic distribution across ISP egress links to avoid congestion. This is expected because most of the path selections result from the application of the BGP lowest AS-path length criterion. Furthermore, this higher latency for traffic added to a similar number of path shifts observed for both IRC schemes, is a clear sign that the IRCs cannot avoid congestion if there is not enough capacity or if the paths are too overlapped.

Third, we also observe that if inter-domain TE uses the LB optimization objective, both IRC schemes perform worse than when using mMLU. This result is expected because the LB objective is more stringent than the mMLU objective (as will be explored next).

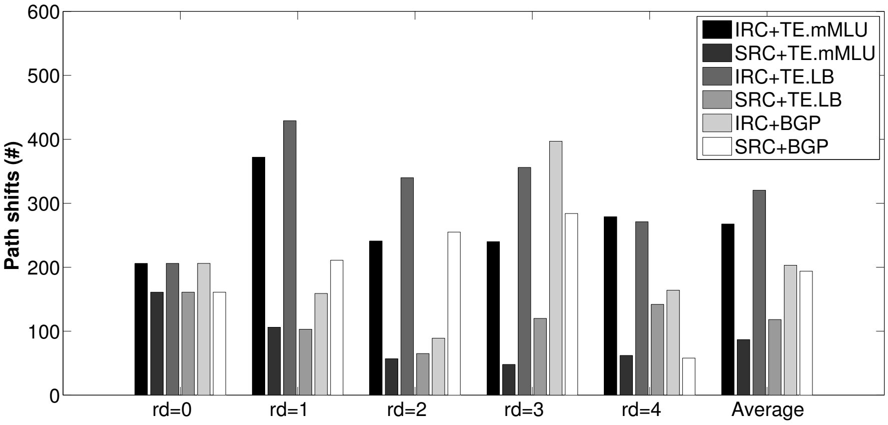


Figure 5.4: Number of path shifts measured in each TE round.

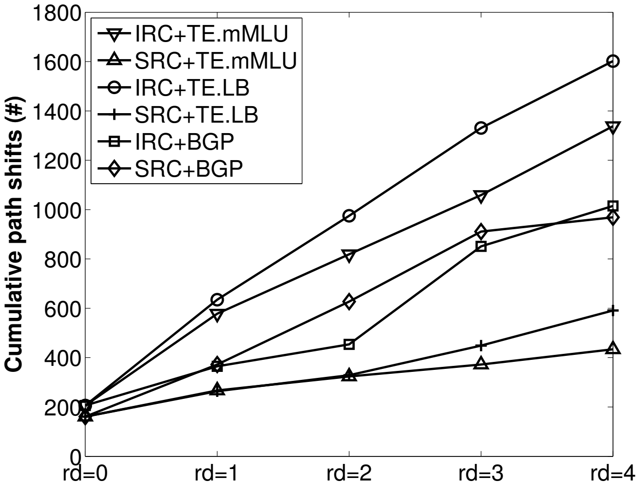


Figure 5.5: Cumulative number of path shifts measured in each TE round.

Figures 5.7 and 5.8 show the values of the statistics with regard to inter-domain TE performance ratios for both the two objectives – mMLU and LB. These results also reveal some important facts. First, *intelligent routing changes at stub ASs can reduce the effectiveness of inter-domain traffic engineering.* As it can be observed, the best ratios, i.e., averages and medians closer to 1, were obtained when the IRC boxes are OFF; and for the more stable combination we registered a TE performance ratio of 3.9 units. This implies that although the inter-domain TE box is able to adapt the routing to the fluctuating patterns of the traffic demands effectively (due to the dynamics of

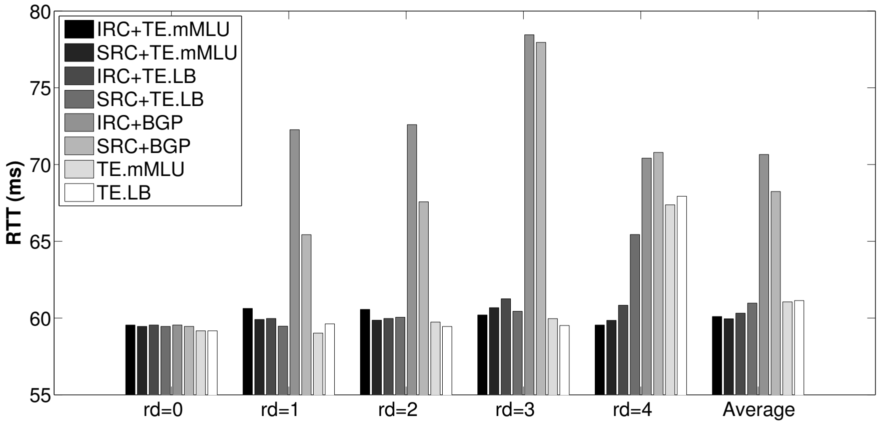


Figure 5.6: Average latency measured in each TE round.

the traffic sources), it is unable to handle stronger traffic changes that are caused by the intelligent route controllers.

Second, as expected, we observe that the network has the worst performance when the inter-domain TE box is OFF. The higher values of the TE performance ratios, expressed in averages and medians, reveals significant traffic unbalances across the egress links of the ISP network.

Third, although inter-domain TE may have a positive impact on SRC scheme, it is also clear that the SRC scheme may experience episodes of strong interaction with inter-domain TE (as it shows bigger excursions of ratios). Furthermore, given the better stability of SRC, this might be a sign that this interaction could involve large traffic aggregates. However, the medians and averages for ratios indicate a lower level of interaction in steady state.

Fourth, if the inter-domain TE uses LB, and the conventional IRC is ON, the ISP network experiences better performance (at least on average), but at the expense of a much larger number of path shifts.

Fifth, the higher values in all statistics for the LB objective reveals that stringent objectives can intensify the interaction between both traffic control mechanisms. Moreover, the high values for standard deviations give a clear sign that there is a high risk of interaction. This result corroborates the fact that intelligent route control is more unstable when the LB objective is used.

5.3.3 Summary of Findings and Remarks

Multihoming Intelligent Route Control (IRC) provides a way for stub networks to improve the performance of their Internet accesses. At the same time, Internet Service Providers (ISP) are increasingly employing inter-domain Traffic Engineering (TE) to

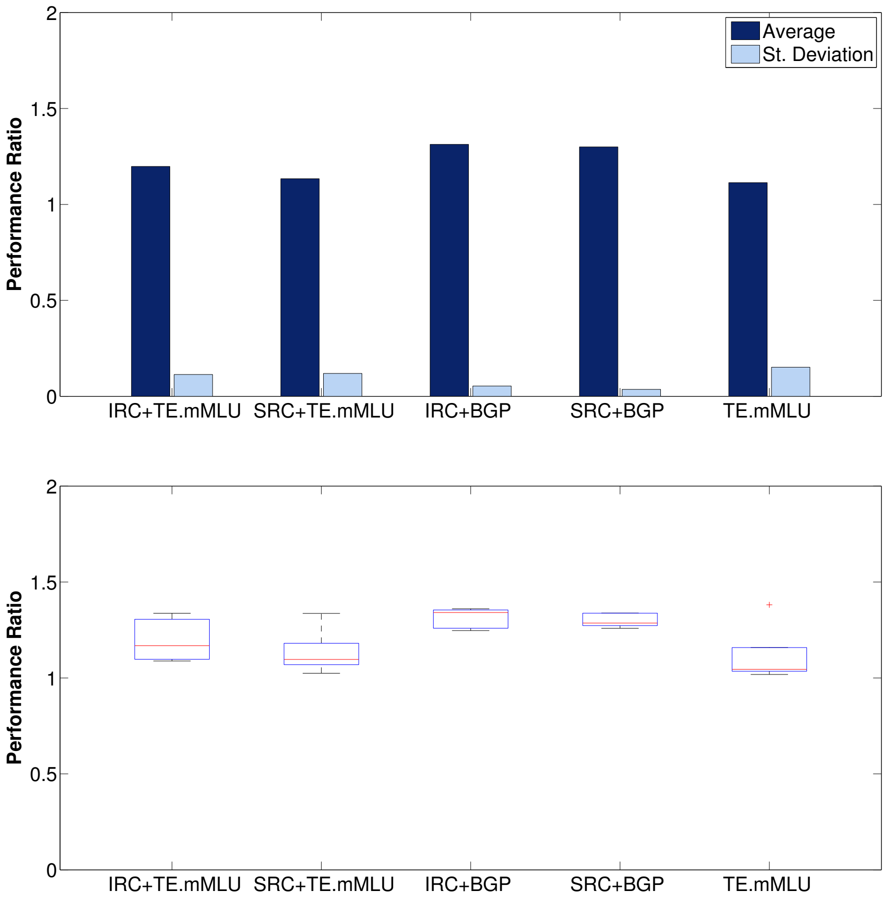


Figure 5.7: Inter-domain TE.mMLU performance ratios.

provide better routing for customers or to meet local traffic objectives. We developed a simulation model to show the presence of interactions between these techniques. In our evaluations, we used two variations of intelligent route control – Conventional and "Social" – and a genetic inter-domain TE algorithm aimed at meeting two similar objectives – the minimization of the whole network Maximum (egress) Link Utilization (mMLU) and Load-Balancing (LB). Below, we summarize our major findings and provide a clue about how to face the intricate problem of the interaction that exist between both mechanisms.

Remark1: With regard to stubs, we observed that the overall IRC stability can benefit from combining Social intelligent Route Control (SRC) with inter-domain traffic engineering, which confirms the validity of the SRC model. Nevertheless, from our perspective, it is not evident that TE route changes can have a significant impact on the overall stability of the IRCs. In fact, it is evident that the pattern of stability of an IRC is mostly determined by the type of IRC strategy employed rather than from the TE actions. Yet the sum of the actions of both middleboxes on the performance of traffic is

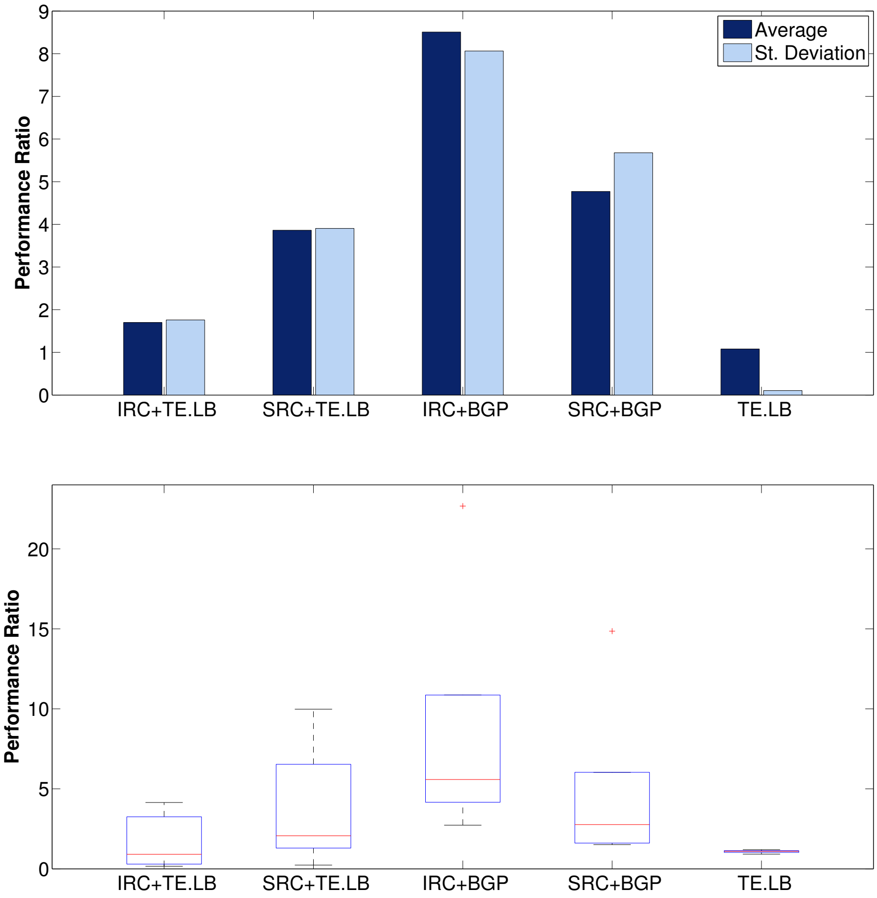


Figure 5.8: Inter-domain TE.LB performance ratios.

clearly advantageous regardless of what IRC scheme is adopted. Moreover, the traffic engineering makes a significant contribution to this result by providing better traffic distribution over the ISP egress points and offering a higher diversity of paths to the stub networks. In short, although the stability issue of IRCs should be taken into account during the validation of any solution taken to address the interaction problem, the biggest concern is still to ensure that the objectives of the traffic in each part are met.

Remark2: The results show that the effectiveness of the TE strategy may be impaired by significant routing changes performed by the IRC boxes, irrespective of whether a social or a conventional IRC strategy is employed. When this important finding is taken account, together with the implications of the previous remark, a solution that is devised to tackle the problem of the interaction between both mechanisms may be slightly biased with regard to the issue of how to obviate the effects of the IRCs over the traffic engineering. As a result, some performance degradation might be expected in the stub part, in favor of a better traffic engineering performance ratio. This degradation may be manifested in two ways: an exacerbation of the number of path shifts and/or a weaker end-to-end traffic performance. However, this degradation should be limited, so that the IRCs are still able to meet the traffic challenges of the stub networks.

5.4 COOPerative Framework (COOP)

This chapter argues that future Intelligent Route Control (IRC) and inter-domain Traffic Engineering (TE) boxes must accommodate diverging interests in terms of traffic objectives between stub and Internet Service Providers (ISP) networks. This section provides thus a joint design of IRC and TE boxes.

The proposed framework, an ISP-friendly IRC COOPerative framework (COOP), rests on the assumption that by introducing a certain degree of cooperation between both boxes, namely in the routing decisions, the selfishness can be reduced, and thus the conflict of interests can be balanced. Its draft was introduced in our 2005 study, presented in [163, 164] and it adopts an approach similar to the negotiation-based routing scheme introduced in a pioneer work also published in 2005 in [169] for pairs of ISPs. Another piece of research on cooperative TE for pairs of ISPs has been carried out in [91, 92].

However this study, to the best of our knowledge, is the first to identify the problem of interactions between IRC and TE and to propose a means of tackling this problem, while at the same time, providing a detailed step-by-step design.

The remainder of this section is organized as follows. Subsection 5.4.1 outlines the COOP framework. Subsection 5.4.2 discusses the key principles that guide the COOP's architecture design. The core subsection, Subsection 5.4.3, sets out a joint IRC-TE architecture for COOP, accompanied by its main algorithms. Finally, a description of the protocol and messages used in the architecture, can be found in Subsection 5.4.5.

5.4.1 Overview of COOP

When traffic demands go beyond a given threshold, the routing settings are no longer valid and should be updated so that the network can return to the optimal regime. In this study, we focus on the control of traffic demand fluctuations that can occur in the transit networks, and are caused by IRCs.

COOP employs a feedback-based method to restrict those fluctuations. On the one hand, inter-domain TE boxes are responsible for monitoring the outgoing traffic and sending explicit feedback information to the IRC boxes. The goal of this feedback is to ensure that the ISP network conditions for maintaining the overall performance are near to optimal, by attracting more traffic or deterring traffic into/from each ingress point. At the same time, the IRCs maintain a utility value assigned to each path, which combines performance metrics with ISP feedbacks.

COOP can be understood as being a closed-loop traffic control framework. First, each IRC reports to ISPs the traffic specification and/or measured traffic performance for every single destination, and requests their feedback (e.g., a preference or cost) via a query message to carry such traffic. If preferred, the inter-domain TE boxes can notify IRCs asynchronously, and use the local traffic measures to compute its feedback.

On receiving a query message, a inter-domain TE box computes the preference/cost by taking into account the difference between the actual performance of the local ISP network and its optimal point, and then returns its feedback to the IRC.

Finally, after receiving the ISP feedbacks, IRCs use them as the parameters for a utility function, so that the IRCs can now select the best paths taking into account both the observed path performance and the associated ISP feedback by selecting the paths with the highest utilities. It is worth noting that the IRCs that are competing for the same egress point resources operate, independently without sharing information between themselves, but it is assumed that as they have been designed as intelligent entities, their combined routing will result in real overall improvements.

As in the case of the TCP (Transmission Control Protocol) congestion avoidance and rate control algorithms [59, 134], that govern the response of the TCP sources to congestion information (e.g., packet losses or Explicit Congestion Notification (ECN) [175]), the proposed solution adopts a similar approach in which cooperation between the major players – routers and traffic sources – is required, as well as a feedback method to restrict the traffic rates. This differs, however, from our framework in that the players are now routing control entities or boxes, which means there is no control to govern the traffic sources. Instead, the feedback-based method aims to restrict aggressive routing changes caused by IRCs to control large traffic demand fluctuations within a target range around its optimal.

As an example of COOP operation, consider again the Internet in Figure 5.3, where transit ISP, `AS1`, can carry out IRC flows from stubs `AS1` and `AS2` toward stub `AS3`. We also assume that `ISP2`, `ISP3`, and `ISP5` advertised to stubs the same set of prefixes over all peering links and that IRC of `AS1` has just configured a routing on its borders routers in response to the latest traffic performance measurements, such that the link `AS1` → `ISP3` can be used to forward packets to most of the prefixes, while `AS1` → `ISP2` is used as an alternative path. This configuration can be achieved by a setting for this subset of prefixes with higher `LOCAL-PREFs` on the BGP routes learned from `ISP3`. This would result in most of the IRC flows that are originated in `AS1`, entering `ISP3`, and thus the ingress point 2 of `ISP1`. Similarly, consider that the IRC of `AS2` sets up routing such that most of the IRC flows also enter through ingress point 2 of `ISP1`. Therefore, if we assume that it is egress point 5 that is carrying out the corresponding outgoing traffic, there would probably be an excess of traffic at this egress point, which would serve to degrade the overall performance of `ISP1`.

Based in the assumption by COOP that costs are used as ISP feedbacks, `ISP1` can handle the excess of incoming traffic at ingress point 2, by trying to deter some IRC flows from this ingress point, and thus from egress point 5. With this end in mind, `ISP1` should compute and notify both `AS1` and `AS2` of the higher costs incurred for carrying the current amount of traffic. In turn, IRCs from `AS1` and `AS2` pass the retrieved ISP feedbacks to the path utilities and adapt routing accordingly, so that some flows are shifted to alternative paths.

It should be noted that in our example we deliberately only focus on the load over egress point 5, because our objective is to design COOP as a distributed system, where each egress point can have its own autonomy to control traffic. For instance, the COOP mechanism at egress points 4 and 6, may react to congestion autonomously by advertising to IRCs, the appropriate costs to attract or deter traffic.

5.4.2 Rationale of the COOP Design

In this subsection, we set out eight key design principles for developing COOP.

5.4.2.1 Design Principles

To narrow the range of possible solutions, we made the design decisions attached to each principle, as outlined below.

Controlling the interaction of the middleboxes. Given the fact that the IRC and TE boxes are agnostic of each other's traffic objectives, they can interact and cause performance degradations arising from routing disputes. The proposed architecture should provide an effective means of tackling this problem, which is the main objective of this work.

Design decision: If a cooperative approach between both middleboxes was adopted, this would obviate the problems caused by the interaction. The projected design employs this approach and seeks to maintain the max-utilization in the ISP networks as low as possible. To achieve this goal, the TE boxes feeds some data (e.g., traffic measurements, resource constraints or preferences) back to the IRCs to bias the IRC routing decisions in favor of a better ISP network performance. More precisely, in our design this feedback aims to restrict the traffic demand fluctuations that are caused by the IRC routing decisions, while ensuring that the ISP network is exactly provisioned. This implies that current traffic demands over the ISP networks must be virtually the ones that were used as input in the previous optimization round. Optionally, the TE boxes could accept requests from the IRCs to bias the TE optimizations, and thus obviate any possibility of aggressive route changes caused by them, for instance, to use specific routes (similar to loose source routing) or to enforce certain routing policies, such as “my traffic should avoid AS y”.

Lightweightness. Rather than making significant changes to existing inter-domain traffic engineering platforms, the proposed design should follow a lightweight approach through which most of the current base mechanisms can be reused or lightly patched.

Design decision: Most of the proposals outlined in this chapter are improvements or extensions of conventional mechanisms of IRCs, TE boxes and routers. A new BGP-like protocol was designed for communications between the IRC and TE boxes, which implements `QUERY-UPDATE` transactions. An IRC sends a `QUERY` message requiring the ISP feedback. In turn, the ISP replies with an `UPDATE` message carrying out the requested feedback. If preferred, an IRC can advertise its performance goals within the `QUERY` message with the aim of biasing the traffic engineering optimizations in favor of some IRC objective.

Efficiency and fairness. The designers of routing mechanisms often forget these important requirements. The resources of inter-ISP peering links, including performance/QoS transport services that have been contracted to peers, should be efficiently utilized and fairly distributed between the IRC flows.

Design decision: The projected design can carry out these goals with the aid of a pricing scheme and a mechanism that inflates flow loads in case of unfairness. For the sake of flexibility and scalability, these mechanisms that aim at controlling efficiency and fairness, should be decoupled from each other.

Preserve the autonomy of the middleboxes. Selfishness characterizes both of the middleboxes but is a factor of under-performance. In a view of this, the projected design should involve both boxes in a cooperative process of path selection, while preserving the autonomy of each of them. This implies that when the routing decisions are biased in favor of the other mechanism, it does not mean that a box is losing all its control over local path selections.

Design decision: On the basis of a principle of separation of the timescales on which the mechanisms of both middleboxes operate, and by adopting a win-win approach, it can be preserved the autonomy of both middleboxes. Any improvements or extensions to existing middleboxes should take account of timescales of operation that are carefully spaced and offer some concessions to the routing control and/or traffic optimizations. With regard to this last point, as mentioned earlier, our proposal is that an ISP should provide a service interface that serves/feedbacks IRCs with the information they need to prevent over-utilization or under-utilization of the ISP network, and a TE set-up option to bias the traffic optimizations.

Common routing strategies. The operational independence of the IRCs is a fundamental principle. Although it can lead to inter-domain congestion and routing oscillation episodes caused by resource contention, it can simplify the development and deployment of the IRCs systems on a large scale. When taking the responsibility for preserving this principle within the proposed joint architecture, the challenge is to ensure that there is cooperation between the independent IRCs while these are competing for the same network resources.

Design decision: By employing common strategies to the routing adaptation that reflect changes in the network resources availability, the IRCs can cooperate and obviate potential oscillations caused by IRC disputes, while conforming to the operational independence principle. More specifically, the IRCs should react and adapt routing to congestion or the feedback information the ISPs provide in accordance with similar rules. In a situation that is analogous to well-known resource contention protocols such as CSMA/CD [176], it means that IRCs should cooperate in a similar way to the Ethernet hosts when they are accessing a transmission medium. In our design, this translates to use similar IRC traffic objectives, and utility functions to quantify the path quality and integrate ISP feedbacks. Otherwise, the overall result might be unpredictable.

Limited disclosure of information. ISPs are reluctant to share information with their peers or customers (e.g., internal topology, metrics or procedures). Their concern is that this kind of information, could give their peers a competitive business advantage. For example, if an ISP finds out that a peer is improving latency, it can adapt its routing to offer latency guarantees to its customers at the expense of that peer. Although there are systematic methods to discover internal information [177], we assume this operational constraint in our design.

Design decision: ISP feedbacks must be advertised through an opaque metric, as introduced in [169]. The exact function that is used to calculate this metric is not shared to the stub networks that are employing a cooperative IRC strategy. In this way, there is no chance that stub networks will uncover the goals or procedures employed by ISPs. In addition, the opaque metric values are bounded to $[0, 1]$ to enable IRCs of multi-homed stub networks to make a fair comparison between feedbacks sent by different ISPs and with different formulations. This also avoids the IRCs to make some normalization operation to combine the ISP feedbacks with the path performance measures.

Multi-service support. The solution should be easily expanded to control the traffic and resource utilization by distinct transport services, including Quality of Service (QoS) services.

Design decision: The choice of a suitable pricing scheme is a particularly effective way of achieving this goal. With differential pricing, the proposed feedback-based architecture can be extended to multi-class networks, and we can expect to get the same benefits as those obtained in Diffserv-based networks [178].

Predictability (and Stability). A key objective of the proposed feedback-based architecture is to ensure that the resources of the ISP networks are used efficiently, as well as the predictability of the traffic over them, while minimizing the subsequent number of route changes performed by the TE box to adapt the inter-domain routing. However, there is a risk that the dynamics of the ISP feedbacks sent to the cooperative IRCs may impair the overall effectiveness of the scheme.

Design decision: Even though other design options can be selected, it is assumed that the maximum values for the ISP feedbacks – the prices – are dynamically adapted in response to the deviations of the ISP network performance from a predefined target range, in accordance with a Multiplicative Increase Additive Decrease (MIAD) policy. When adapting the maximum ISP feedback values, the aim is to deter traffic from the ISP network quickly and, at the same time, attract traffic slowly to avoid IRC oscillations, especially when the network is under-utilized. MIAD policy should be ideally suited to this adaptive process, since it can combine an exponential growth of the maximum price values with a linear reduction whenever a network overload occurs.

5.4.2.2 Summary of the key design decisions

Bearing in mind the design principles that were referred to earlier, the remaining details of the devised framework will be outlined in Subsection 5.4.3. Nevertheless we will briefly describe the memorandum of the main design decisions here.

First, we assume that the transit network – the Internet Service Provider (ISP) – estimates its performance regime at regular intervals *and* computes the deviation from the optimal point (i.e., when the ISP network is exactly provisioned). On the basis of this deviation, the ISP computes an *aggregate* feedback that can enable it to return its network to optimal performance. At this stage, the ISP only needs to deal with the whole aggregate of traffic that is sharing each egress point of its network and not yet concerned about whom each flow belongs to. This decision allows a decoupling of the mechanisms that will be devised to achieve the twin goals of network efficiency and fairness among the IRC flows.

Following this, the ISP computes and advertises its *individual* feedbacks – the per-IRC flow prices – via `UPDATE` messages sent synchronously to IRCs (or asynchronously depending on the COOP protocol mode). The feedbacks notify the prices of carrying out the amount of traffic that belongs to each of the IRC flows, which play an important role in repulsing/attracting traffic from/to the ISP network. It is understood, however, that these prices are advertised on the basis of an opaque metric which allows it to adhere to the principle of a limited disclosure of the internal information of the ISP network. Moreover, the values of the metric are bounded to $[0, 1]$ to enable a multi-homed stub network to make fair comparisons between prices sent by different ISP networks. We also assume that the maximum value for a price is dynamically adapted

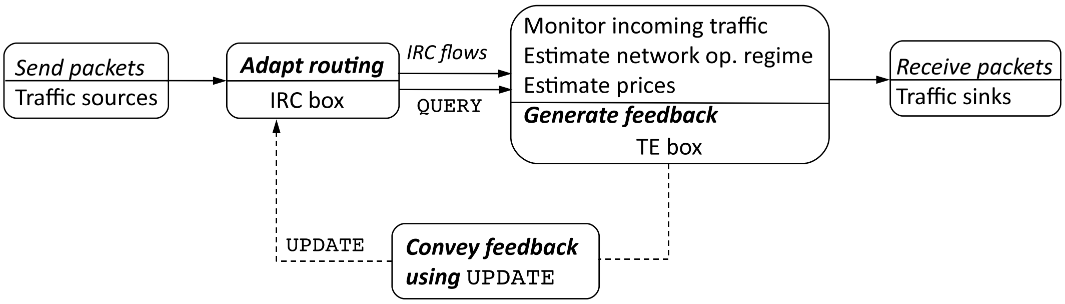


Figure 5.9: Feedback-based architecture of COOP.

by means of a MIAD policy and in this way improve the overall effectiveness and stability of the scheme.

Finally, we assume that the IRCs employ a path selection algorithm that employs a utility function to combine the transit network feedbacks – the prices – with the performance measures of paths. In this way, an IRC can more easily compare paths that are constrained to more than one metric, including their performance measures and the network transit feedbacks. An IRC may sends a QUERY message requiring an ISP feedback or receive it asynchronously from the ISP within a UPDATE message.

5.4.3 COOP Architecture and Algorithms

The feedback-based architecture of COOP is displayed in Figure 5.9. The main goal is to control incoming traffic demands, while the resources are allocated to customers in an efficient and fair manner. In addition, COOP architecture is based on a set of extensions to conventional inter-domain Traffic Engineering (TE) and Intelligent Route Control (IRC) boxes. This subsection provides this joint IRC-TE architecture for COOP, accompanied by its main algorithms. From now onwards, for the sake of simplicity, the descriptions below will regard the minimization of maximum (egress) link utilization (mMLU) as the traffic objective of the Internet Service Provider (ISP) network, although they can be easily adapted to other objectives.

5.4.3.1 COOP Extensions to Inter-domain Traffic Engineering

COOP design divides the inter-domain traffic engineering problem into two components. First, in every interval of $T_{offlineTE}$ seconds, a centralized off-line TE box takes as input the traffic demands, the resource and topology constraints, and outputs the optimal routing, as it was discussed in Subsection 5.2.1. Once the router configurations are applied accordingly, the centralized off-line TE box exports to each egress point of the network the corresponding target optimal utilization OU. This off-line traffic engineering box is basically a conventional box that has been extended with a COOP communication module.

Second, each egress point has an on-line TE agent – a COOP controller –, that obtains current utilizations of local resources from border routers or from a monitoring platform, and ensures that theses are used in an efficient and fair manner. More specifically, it employs a closed-loop feedback control to manage the utilization of local resources, before the off-line TE box has made any new routing decisions. Each COOP controller computes the price to carry the current traffic load, and sends individual feedbacks to IRCs, whose flows are sharing the local egress point. The computed price aims at signaling what utilization changes are required, so that the egress point resources can be used in an efficient and fair manner. This kind of feedback sent to the IRCs will coordinate their actions, so that they can manage collectively the over/under-utilization of the ISP network.

Following this, we detail the on-line TE mechanisms, beginning with the measurements that have been carried-out to compute the current ISP network regime.

Measuring the performance of the ISP network. A COOP controller must, first, keep track of the resources utilization at its egress point. To achieve this, it is assumed that each COOP controller is linked to the ISP monitoring box from which the required state information about the local load/utilization can be imported. This includes the individual contributions of the IRC flows at every adaptive period, denoted as the interval of sampling T_s (within a timescale $<<$ the $T_{offlineTE}$ timescale). Optionally, the COOP controller can estimate the outgoing traffic through its egress point. In effect, these data can be obtained in two distinct manners: indirectly by joining the traffic demands data with the routing information (if the ISP does not generate traffic) or more accurately, as was mentioned, by assuming the ISP monitoring box is also linked to the egress routers and can obtain the needed state information about the load/utilization over each egress link (or transport service contracted to upstream ISPs), including the individual contributions of the IRC flows to the egress point load/utilization for a time interval of sampling T_s.

Regarding the capturing of raw traffic data by the ISP monitoring box, instead of using a packet-based traffic capturing tool, such as SNMP (Simple Network Management Protocol), a flow-based tool such as Cisco NetFlow is enough [126, 179]. The per-flow traffic characteristics provided by NetFlow, to be more precise start and end times and the volume of traffic for a time interval T_s are then stored in a relational database. Subsequently, it can be combined a traffic predictor to estimate the traffic more in a more accurate way (e.g. by using an Exponential Moving Average (EMA) method or Last-Value sampling method).

Calculating ISP route prices and feedback. Pricing is usually a process of associating a number with a service. In this study, the use of prices is justified on two grounds. First, prices are used as a means/incentive to influence the behavior of IRCs [180]. Later on we will see that prices are dynamically adapted and sent to IRCs as feedback signals in response to deviations in the network performance from

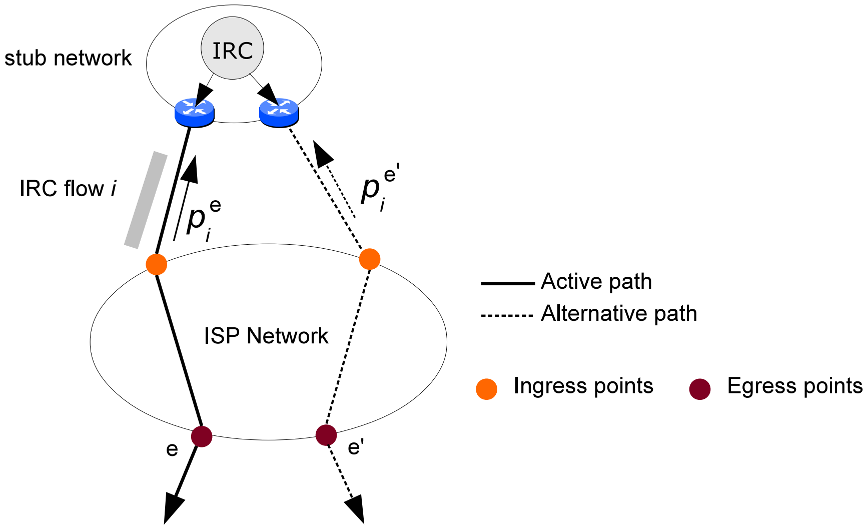


Figure 5.10: ISP price feedback.

a predefined target range. Second, prices can materialize the objective of limiting the disclosure of information about ISP networks, since the function of the mapping enables the exact ISP traffic objectives to be hidden.

At the same time, feedback is a process of advertising, within an `UPDATE` message, the raw route prices to IRCs, which implicitly provide the required corrections to the incoming traffic load, so as to keep the total traffic load within the expected state. More specifically, whenever each control interval of $T_{feedback}$ seconds is reached, a price, denoted as p_i^e, for a given IRC flow i is calculated for conducting the egress point e to a given target utilization, and hence collectively the network. Thus, p_i^e is a function of the current egress point utilization (U^e), the optimal utilization (OU), and the rate/traffic demands of the IRC flow i (d_i), as it is shown in Equation (5.8).

$$p_i^e = f(U^e, OU, d_i) \tag{5.8}$$

If an IRC flow has multiple entry points and thus possibly multiple egress points toward the destination, to avoid IRC oscillations caused by price under-estimations, the price $p_i^{e'}$, $e' \neq e$, advertised to the IRC through each alternative ingress router is calculated by taking into account the extra-load introduced by the IRC flow (d_i), as if the corresponding egress point e' was carrying that flow. Figure 5.10 displays an illustration of this scenario. In this manner, we are able to solve a problem that is similar to the self-load problem introduced to the network by IRCs [89].

For the sake of simplicity, we will adopt a price adaptation scheme based on the use of a linear mapping between the egress point utilization and the price in a range $[0, 1]$. This will enable the IRCs to make fair comparisons between the different prices formulations, as well as to avoid carrying out normalization operations that mix prices with path performance attributes. Figure 5.11 displays an illustration of how this mapping works.

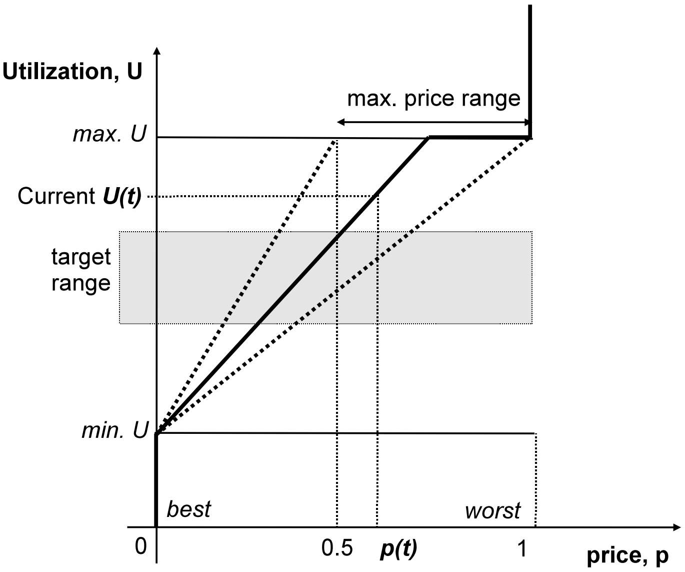


Figure 5.11: Mapping of the ISP performance measures to agnostic values.

Furthermore, the proposed price adaptation is based on a similar philosophy of Adaptive Random Early Detection (A-RED) for queue management, that is, of adapting the maximum dropping probability of packets (max_p) to keep the average queue size (avg) within an given target interval.

We employed thus a revised A-RED mechanism. However, the price adaptation differs in several ways from that of A-RED [181] (see Algorithm 5.2):

- The adaptive parameter maximum price (p_{max}) varies between 0.5 and 1 (rather than between 0.01 and 0.5). This lower bound of 0.5 on p_{max} was chosen to optimize the pricing algorithm in situations when the ISP network is over-utilized/over-subscribed;

- The adaptation function of p_{max} uses a MIAD (Multiplicative Increase Additive Decrease) policy (instead of an AIMD (Additive Increase Multiplicative Decrease) policy). This policy and the range for the adaptive parameter aims to deter traffic from the ISP quickly and attract traffic slowly to avoid IRC oscillations, and thus to contribute to the overall stability of inter-domain routing and traffic;

- The adaptation of p_{max} is performed slowly over timescales longer than the IRC path switching and TE feedback timescales, and in small steps; but over timescales shorter than the off-line TE timescale.

An egress point is over-provisioned – by definition – if $U_e > OU$, and is under-provisioned if $U_e < OU$, where U_e is the current egress point utilization and OU is the target utilization goal, which is generally the optimal utilization. In our evaluations, we consider the target range OU_{range} defined as in Equation (5.9), where the optimal utilization is OU and w is a small step (e.g., $w = 1.1$). For the sake of concreteness, this target range is illustrated in the shaded area of Figure 5.11. Thus, the parameter p_{max} is adapted, according to the Algorithm 5.2, depending on the current value of the utilization compared to the target range.

$$OU_{range} = \left[\frac{OU}{w}, OU * w\right] \tag{5.9}$$

Algoritmo 5.2 $p_{max}(\{U, OU_{range}\})$

```
**Increment:** $\alpha = 1.1$
**Decrement:** $\beta = 0.1$
**if** $(U > OU_{range})$ and $(p_{max} \leq 0.909)$ **then**
    Increase: $p_{max} = p_{max} * \alpha$
**end if**
**if** $(U < OU_{range})$ and $(p_{max} > 0.5)$ **then**
    Decrease: $p_{max} -= \beta$
**end if**
```

Improving fairness among the IRC flows. If a simple pricing mechanism is used, it might cause a problem of unfairness, as it does not consider the contribution of each IRC flow to the current utilization of the egress point. In other words, the current distribution of the IRC flows must also fit the optimal distribution of IRC flows that have been previously computed by the off-line TE algorithm. To achieve this, the current egress utilization U^e must consider the imbalance of the IRC flows against the optimal distribution. More precisely, every single IRC flow load used in the U^e calculation must be inflated whenever the IRC flow imbalances the flow distribution.

To describe the imbalance between K IRC flows assigned to an egress point, we consider a Flow Balance Index (FBI), I_e, defined by Equation (5.10), where $d^e(k)$ denotes the current traffic demand for the destination k, and $t^e(k)$ denotes the (target) traffic demand for destination k defined by the centralized off-line TE algorithm. D^e represents the current total traffic demands over the egress point e, and T^e the total traffic demands assigned by the off-line TE box to the egress point e in the interval of $T_{offlineTE}$ seconds.

$$I_e = \sum_{k=1}^{K} abs\left(\frac{d^e(k)}{D^e} - \frac{t^e(k)}{T^e}\right), \forall e \tag{5.10}$$

Ideally FBI is zero, which means that the flow distribution of the IRCs fits precisely the optimal mixture of the traffic. Whenever the ratio $\frac{d^e(k)}{D^e}$ approaches the ratio $\frac{t^e(k)}{T^e}$, I_e shrinks. Otherwise, it expands and thus the IRC flow imbalance increases.

Subsequently, an inflation load, denoted by β, is calculated from the difference between the current FBI, $I^e(current)$, and the estimated FBI, $I^e(shifted)$, if the new flow is shifted/admitted to the egress point, i.e., $\beta = I^e(current) - I^e(shifted)$.

Finally, the IRC flow load $L_{inflated}$ being reported to the price algorithm is given by Equation (5.11), where $L_{original}$ is the current flow load over the egress point e, and n is an inflation factor being tweaked by the operator of the ISP network to control the best way to exacerbate the inflation load.

$$\begin{aligned} L_{inflated} &= L_{original}(1 + n\beta) \\ \beta &= I^e(current) - I^e(shifted) \end{aligned} \tag{5.11}$$

To illustrate the calculation of the $L_{inflated}$, consider a scenario in which 20 IRC flows of 64Kbps for 10 different destinations must share an egress point e according to a Benford distribution [182] (i.e., $t_k^e \propto (log_{10}\left(\frac{k+1}{k}\right))$ (see Figure 5.12) and an IRC flow of 64Kbps to be admitted for destination $k = 3$. Consider two cases of imbalance in traffic for $k = 3$, these are when current d_3^e is higher (case 1) or lower (case 2) 64Kbps from the optimal distribution. Thus, in the first case, the fairness mechanism will report to the price mechanism a $L_{inflated}$ equal to 56.6Kbps to attract the IRC flow, because it will improve the overall flow distribution. On the other hand, if we consider d_3^e is higher 64Kbps from the optimal point, the $L_{inflated}$ being reported will be equal to 78.2Kbps, to repulse this flow because it will impair the IRC flow distribution.

5.4.3.2 COOP Intelligent Route Control

Upon receiving an `UPDATE` message from the ISP network, an IRC is responsible for performing the adaptation of the inter-domain routing at the stub with the aim of maximizing the benefit or *utility* to the end-user network or application, that is constrained to the price advertised by the ISP network. Below we describe the design for a delay-driven IRC, although it should be pointed that this is easily extended to other objectives and to multi-objective-driven IRCs.

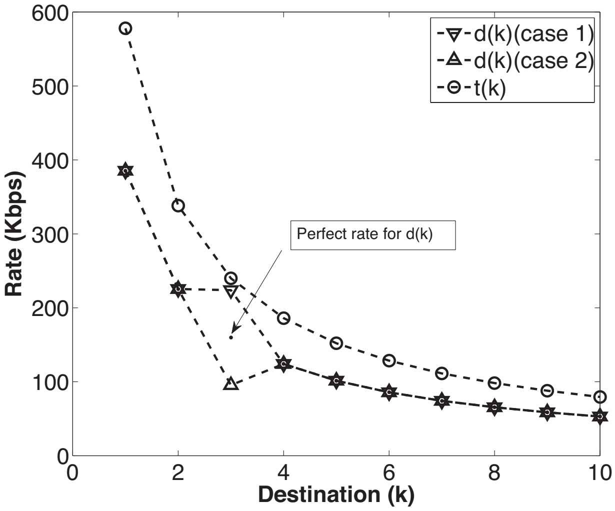


Figure 5.12: Examples of IRC flow mixtures against an optimal distribution $s(k)$.

We assume thus that an IRC runs a path switching algorithm that relies on the notion of utility function [183]. An utility $U(.)$ is assigned to each traffic flow for each path to define the perceived performance value provided by the target path. Following this, an IRC attempts to obtain the best degree of satisfaction for both networks – stub and transit network. In particular, it maintains an additional variable, that stores the degree of satisfaction, and is given by the surplus between the utility and the price of using that path/ISP. The goal of the routing adaptation is to maximize this surplus, subject to the maximum value of delay. In short, this strategy enables the objectives of both parties to be combined.

In this work, we consider the following constraints concerning the utility values $U(d)$, where d denotes the delay measured (OWD or RTT) throughout the path offered by the ISP_k:

1. $U(d) : [0, d_{max}] \rightarrow [0, 1]$, if the path offers a delay higher than the delay bound d_{max} its utility is zero;

2. $\lim_{d \to 0^+} U(d) = 1$, since the delay tends to zero, the utility tends to one;

3. $U'(d) \leq 0$, since the utility should decrease as the delay increases;

4. $U''(d) > 0$, to make the utility more sensitive as the delay approaches the delay bound.

In concrete terms, a modified sigmoid utility function was chosen for our design to overcome these constraints (see Equation (5.12), where α determines the steepness of the curve). The parameter α is set to 0.0125 to match the utility to the shape of the E-model Rating from ITU's G.107 recommendations [108].

$$U(d) = \begin{cases} 1 - \frac{2}{1-exp(-\alpha(d-d_{max}))} & \text{, if } d \in [0, d_{max}[\\ 0 & \text{, if } d \geq d_{max} \end{cases} \tag{5.12}$$

Algorithm 5.3 details the whole utility-based IRC path switching algorithm, where the IRC selects as best path for each traffic flow the one that has the the largest surplus. Obviously, with this incremental algorithm we are not able to obtain the collective optimal IRC routing immediately. We assume this strategy is the best to avoid a massive number of path switches that would result if the IRC would be able to optimize the whole routing at once (like it does a TE box).

Algoritmo 5.3 IRC($\{P, D, d_{max}\}$)

```
Require: {P} - vector of the set of AS-paths for a prefix $p$
         {D} - matrix of the set of path performance attributes
         {p} - vector of criteria representing the ISP route prices
         {$d_{max}$} - delay traffic bound
Ensure:  $P_x$ - the active path fits traffic goals toward prefix $p$
  Wait for changes in path performance attributes and/or ISP price
  /* Basic IRC path selection process */
  Compute the vector of Utilities $U$
  Identify the set of feasible paths $P'$
  $[P', U'] \leftarrow \{P_i \in P : U_i(d_i) \geq 0\}$
  if $\| P' \| \neq 0$ then
    /* Identify the highest rank path $P_i \in P'$, that gives the max. surplus in $S'$ */
    $x' = argmax\chi(\{P', S'\}) = U_i'(d_i) - p_i, \forall P_i \in P'$ /* the ranking function $\chi$ compares the surplus of all feasible paths in $P'$*/
    if $\| x' \| > 1$ then
      /* If there is more than one path equally good, apply the standard BGP process to break the ties */
      $P'' \leftarrow \{P_i \in P' : i = x'\}$
      $x' \leftarrow BGPTie(P'')$
    end if
    Switch traffic towards $p$ from $P_x$ to $P_{x'}$
    $P_x \leftarrow P_{x'}$
  end if
  /* End of IRC path selection process */
```

5.4.3.3 Discussion of Timescales

Similar timescales must be used to synchronize some of the mechanisms that have been described. On the other hand, to obviate the occurrence of interaction between the mechanisms, and resulting oscillations, the principle of the timescale separation must be observed. In this work, multiple timescales must be addressed as displayed in Figure 5.13 and discussed here.

(a) Since the IRC reaction, denoted as T_{irc}, can be anything from a few seconds up to tens of seconds, the TE feedback should be sent over a timescale that is no shorter than the IRC timescale (i.e., $T_{irc} \approxeq T_{feedback}$). With this proximity, the objective is to synchronize both mechanisms to make it possible to manage the over-utilization of the ISP network. The $T_{feedback}$, however, should be smaller than $T_{offlineTE}$. As a result of this separation, the IRCs (in conjunction with the COOP controller) are given the opportunity to manage this over-utilization. If they cannot manage the over-utilization during an interval of $T_{offlineTE}$ seconds, the offline TE box is activated and adapts the ISP routing to fix the problem of over-utilization. In short, the COOP controller operates over timescales faster than the off-line TE, and no greater than the IRC timescale (i.e., $T_{irc} \approxeq T_{feedback} << T_{offlineTE}$);

(b) The adaptation of the maximum price p_{max} between 0.5 and 1 is performed slowly over a timescale that is greater than the IRC path switching and ISP feedback timescales, but over a timescale that must be shorter than an interval of $T_{offlineTE}$ seconds (i.e., $T_{feedback} < T_{pmax} << T_{offlineTE}$). With this separation, the adaptation process of p_{max} only depends on the long term dynamics of egress point utilization (rather than short term dynamics) to avoid improper adaptation and the resulting instability.

5.4.4 Implementation Issues

This section describes some practical issues to bear in mind during the implementation of the COOP framework implementation.

5.4.4.1 Address discovery of the COOP controllers (or the COOP proxy)

To establish peering sessions between IRCs (supporting COOP) and ISP COOP controllers, a mechanism is needed to locate the ISP COOP controller(s) before establishing the cooperation. If preferred an ISP can implement a single COOP proxy, which plays the role to represent all ISP COOP controllers and relays the IRC COOP requests – `QUERY` messages – to the proper COOP controllers, and replies as a COOP controller to the IRC.

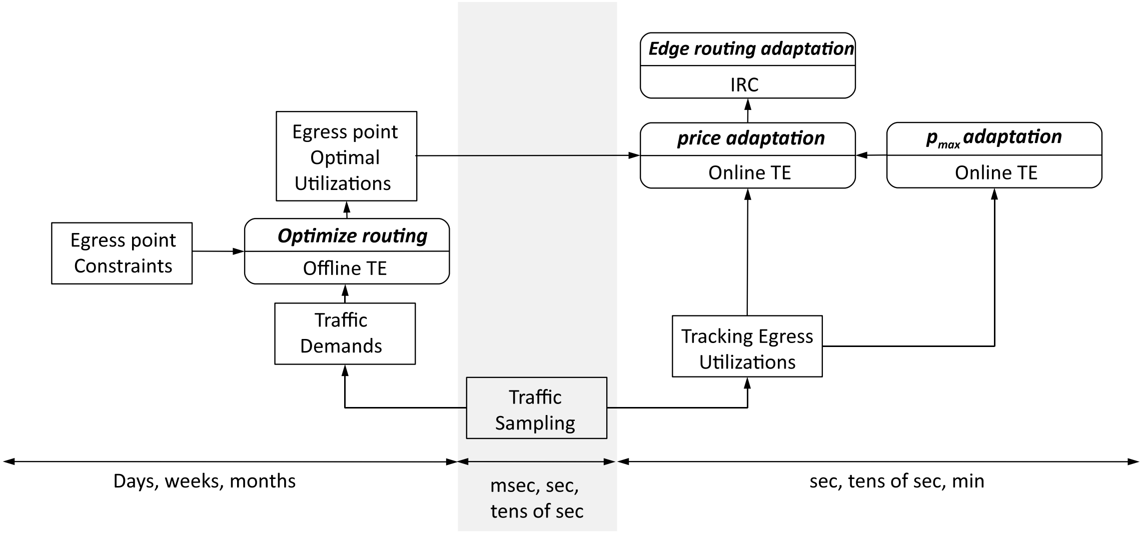


Figure 5.13: The multiple timescales of COOP, IRC and TE mechanisms.

An appealing option to discover a COOP device, is to rely on a BGP UPDATE, namely a COOP Advertisement (COOPA) can be used to advertise the IP address(es) of the ISP COOP controller(s) or proxy that serve(s) the IRC. For instance, a new type of the BGP community attribute can be defined for the effect, e.g., a COOP community. As a result, the routes originated by the ISP will also advertise the IP address x.x.x.x of COOP controllers within the COOP community attribute.

Another attractive option is to use a new Domain Name Service (DNS) Resource Record (RR) [184]. For instance, a new RR called COOP CONTROLLER can be added to the reverse DNS, and subsequently the IRC only needs to perform a reverse DNS query for the prefix, of the ISP and ask for the COOP CONTROLLER IP address. With regards to a COOP proxy, the discovery process is the same.

5.4.4.2 COOP peering sessions

Once the ISP COOP controller (or proxy) is located, a handshaking process starts. To undertake this process, some kind of initialization messages must be exchanged between each pair of IRC-ISP COOP devices to request the setting-up of a peering session. An initial solution is to rely on an eBGP mechanism and benefit from the robustness of its infrastructure, for instance, by setting-up a standard or multi-hop eBGP session between both mechanisms.

Unfortunately, the use of the eBGP mechanism restricts the IRC-ISP COOP devices to operating at the BGP timescale, which can impair the overall performance of the scheme and can serve to increase of the complexity of the BGP software. Instead, we suggest forming a new signaling protocol from scratch to establish the peerings. Even this approach can give rise to potential problems due to firewall blocking. In the past, this problem was responsible for the failure of several promising distributed systems technologies, such as CORBA [185].

In the light of this, we have outlined an out-of-BGP-band signaling protocol, called COOP that will be described in detail in Section 5.4.5. As can be observed, two types of COOP messages are exchanged to control a COOP peering session: an `OPEN` message and `CLOSE` message. An `OPEN` message is sent in order to request the setting-up of a COOP session and to change the parameters of an existing peering. On the other hand, a `CLOSE` message is sent to shutdown an existing peering. Two other messages, `QUERY` and `UPDATE`, are used for requesting and updating the pricing information, as well as a `NOTIFICATION` message is used to notify an error condition.

5.4.4.3 Deployment and Scalability

We can implement the proposed scheme at any provider network that makes a contract with its customer in the form of an SLA (Service Level Agreement) that specifies the reachable destinations, service parameters, and the need for cooperation.

ISP feedbacks sent to IRCs are generated at the egress points, that is by the on-line ISP COOP controllers that are attached to these. Since ISP COOP controllers are external processes to egress routers, to the ISP TE platform and to the ISP monitoring devices, with minor changes, the COOP scheme can be incorporated into the current ISP network infrastructure. In fact, ISP COOP controllers only need to interface with these devices. Moreover, COOP does not need any changes in core routers; it only requires a small amount of computational overhead in the edge routers, due to traffic tracking, using Cisco IOS NetFlow, SNMP or other monitoring infrastructure [126,179].

As was mentioned earlier, if preferred an ISP can implement a single COOP proxy that relays the IRC COOP requests to the target COOP controllers. This configuration can be useful between stubs and ISPs more densely connected, for instance when they maintain 3-4 mutual peering links. In this case, it would be necessary to establish a single peering session.

In short, most of the need for changes and greater effort stems from the implementation of the on-line ISP COOP controllers, which rely on a distributed approach. Consequently, the architecture of the COOP framework is incrementally deployable and highly scalable. In addition, the proposed scheme is practical, and constitutes a cost-effective means of solving the problem of ISP resource optimization and guarantees a fair sharing between the IRC flows, because it adjusts the feedbacks in a distributed manner.

5.4.5 COOP Protocol

A communication protocol is required by the COOP framework for exchanging the feedback information between the COOP controllers and IRC boxes. Before proceeding, let us note that the usage scenarios for the COOP protocol are not limited to IRC-COOP controller/proxy peering. The COOP protocol can also be used between two ISPs employing TE or between two remote IRCs for incoming traffic control.

The COOP protocol was designed as a BGP-like protocol. However, we removed the related issues with the setting-up of the TCP sessions. In short, the Finite State Machine (FSM) and messages of the COOP protocol are quite similar to the those of BGP. In other words, COOP is a revised BGP protocol, but let highlight only with regard to aspects of signaling.

5.4.5.1 Protocol Messages

All the COOP messages have a common header with a marker for synchronization (e.g., 16 bytes of “1s”), the length of message and its type. The COOP messages format is described below.

OPEN To open a COOP peering session, an IRC sends an `OPEN` message to challenge a COOP controller of an ISP to cooperate (after the TCP three-way handshake is completed). The `OPEN` message contains information about the COOP peer that initiates the session: COOP version, AS number, peer ID, optional parameters, and optional parameters length. When an `OPEN` message is received, all the fields are checked. If there are no errors, the TE box sends an `OPEN` message and `KEEPALIVE` message. Otherwise, the session is dropped and sends a `NOTIFICATION` message.

QUERY An IRC uses `QUERY` messages to request ISP feedback information for one or more IRC flows or services. For this reason, `QUERY` messages contain lists of destinations with information about each. Each entry in this list contains a flow or aggregate ID, the destination network or host of a flow or aggregate of traffic, and the transport service ID; and optional service parameters and optional service parameters length.

UPDATE Upon receiving a `QUERY` message, the COOP controller determines the ISP route price for each IRC flow or traffic aggregate, and sends back the corresponding `UPDATE` message with the list of service and prices pairs. Each entry in this list contains a flow or aggregate ID, the transport service ID, and the new price. In the case of employing a COOP proxy, the ID of the ingress border router associated with each entry must be transmitted as optional parameters to the IRC be aware of the IRC flow entry point. As described so far, we assume the existence of a mechanism that dynamically adapts the ISP route prices in response to ISP load changes and deviations from a predefined target range. This means that normally a pair of messages `QUERY-UPDATE` are exchanged synchronously through a reply-response, i.e., an ISP only replies with a `UPDATE` message after receiving a `QUERY` message. In addition, ISP COOP devices also can send out `UPDATE` messages asynchronously if there are important changes in the network operation regime.

KEEPALIVE If there are no ISP performance changes, all the ISP prices remain unchanged, and only `KEEPALIVE` messages are sent back from the ISP COOP devices to the IRCs.

NOTIFICATION Any COOP device – IRC, COOP controller or other – sends `NOTIFICATION` messages in response to error or special conditions.

CLOSE Any COOP device – IRC, COOP controller or other – sends a `CLOSE` message to tear down the COOP session.

5.4.5.2 Sequence of Messages

Figure 5.14 shows the messaging sequence for a COOP peering session, as follows:

1. Before starting a COOP session, the IRC discovers and establishes a TCP connection with an ISP COOP controller.
2. IRC and the COOP controller exchange COOP `OPEN` messages.
3. IRC and the COOP controller initiate the exchange of periodic health messages, i.e., COOP `KEEPALIVE` messages. After the initial handshake, the IRC and the COOP controller are ready to exchange pairs of COOP `QUERY-UPDATE` messages.
4. The IRC sends a `QUERY` message to the ISP. The COOP controller of the ISP determines a route price for each flow or aggregate listed in the `QUERY` message.
5. The COOP controller returns the `UPDATE` message to the IRC. After the first `QUERY` message, `UPDATE` messages can also be sent asynchronously.
6. To terminate a cooperative session, the IRC or the COOP controller sends a `CLOSE` message to its peer.

5.4.5.3 Finite State Machine (FSM)

There is a set of different messages that can be exchanged during a COOP session. The way that messages are handled depends on the triggering of a number of events and the state of the COOP speaker. Thus, each COOP speaker maintains a Finite State Machine (FSM) associated with each of its peers. The COOP FSM is shown in Figure 5.15 and includes the following basic states:

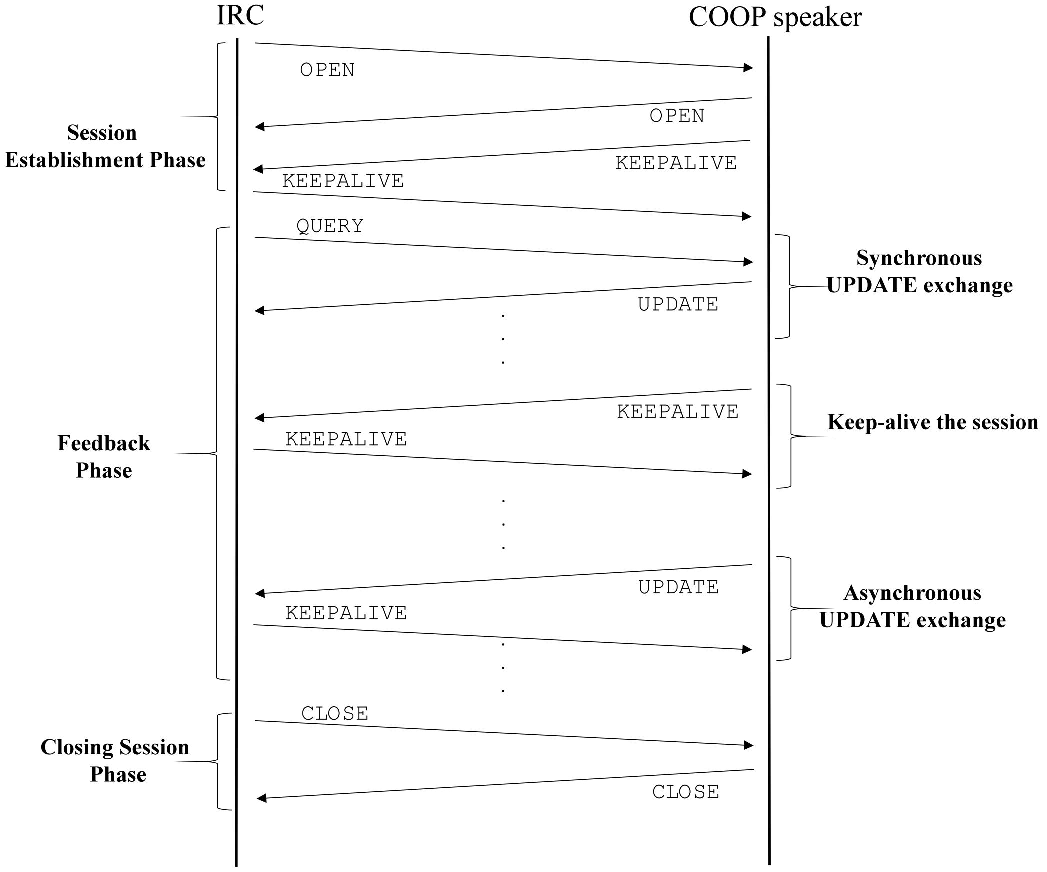


Figure 5.14: COOP messaging sequence between an IRC and a other COOP speaker.

IDLE: Initial state of a COOP speaker. In this state, the COOP speaker is not yet ready to start any connection or accept any incoming connection. When either a manual or an automatic start event occurs, a COOP speaker starts a TCP connection with its peer, and moves to the Connect state in order to start a COOP connection. Alternatively, a COOP speaker can play a passive role in so far as it first listens to the incoming COOP connections. Then it moves to the Listen state instead of the Connect state.

LISTEN: Once a TCP connection has been completed, a COOP speaker sends an `OPEN` message and moves to the OpenConfirm state.

CONNECT: In this state, it is assumed that the local COOP speaker plays an active role in so far as it first starts a COOP connection. Thus, the local COOP speaker sends an `OPEN` message to its peer and moves to the OpenSent state.

OPENSENT: In this state a COOP speaker waits for `OPEN` messages. If an `OPEN` message is received, it sends an `OPEN` message followed by a `KEEPALIVE` message, and starts a KeepAliveTimer and moves to the OpenConfirm state.

OPENCONFIRM: A COOP speaker waits for a `KEEPALIVE` message from its peer or generates a KEEPALIVE message. It generates a KEEPALIVE message when the KeepAliveTimer expires. A COOP speaker moves to the Established state when it receives a KEEPALIVE message from its peer.

ESTABLISHED: In this state, a COOP peering is established. In this state, each COOP speaker exchanges UPDATE, QUERY, KEEPALIVE and NOTIFICATION messages with a peering COOP speaker.

Finally, it should be added that all the events that operate over the COOP FSM are triggered by the COOP logic or by human intervention (e.g, by manual switching from selfish to cooperative behavior).

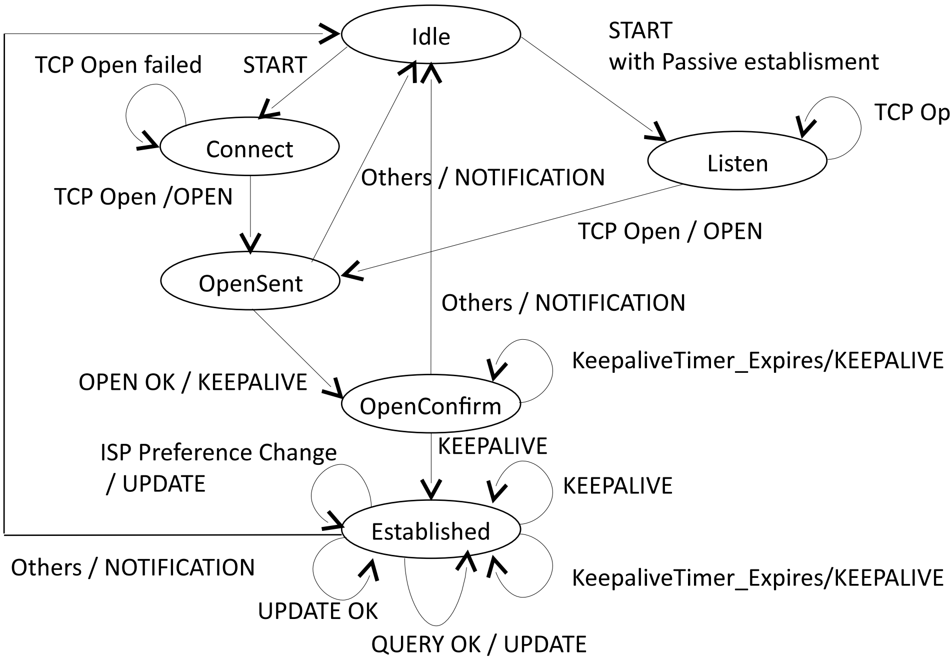


Figure 5.15: COOP finite state machine.

5.5 Evaluation

In this section, we investigate the benefits and feasibility of the COOP framework. First, we describe the data traces, the traffic demands (including an analysis), and the network topology, together with the link latency and available bandwidth functions used in the evaluation. Following this, we describe the evaluation methodology and its main objectives. Finally, we give the COOP approach compared with the selfish approach, and study the fairness of the traffic control achieved by COOP. A later section studies the tuning of COOP to reduce the number of route changes performed by the IRCs.

5.5.1 Trace

We use one set of data traces that were collected at the GÉANT pan-European academic network in 2005 [186]. The data traces were collected by means of the Cisco's IOS NetFlow feature [126]. The NetFlow measurements were made through a period of two weeks, as shown in Figure 5.16. Each sample in the figure represents the amount of bytes seen during each interval of 15 minutes, multiplied by 1000, because the NetFlow sampling was performed at a rate of 1/1000. Then, the traffic was divided into demands. A demand represents a given amount of traffic (in bytes) from GÉANT's users to a destination. Each demand models the traffic aggregation of several small traffic flows sent to that destination.

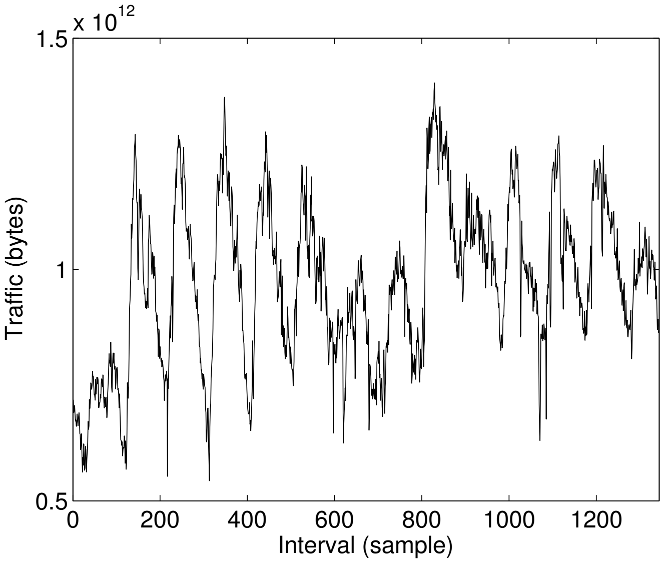


Figure 5.16: GÉANT Traffic: traffic evolution from Sunday (05-07-10) to Saturday (05-07-23).

5.5.2 Analysis of the Traffic Demands

A lack of scalability of inter-domain Traffic Engineering (TE) may limit its applicability to large networks due to high computational costs and overhead of routing changes, because of the potential growth of the routing tables of large networks. One suggested solution of reducing these overheads is to focus the TE optimization on the paths to the popular destinations (a.k.a. top receivers) and thus be able to shift a large volume of traffic with a small number of paths switches, rather than shifting the traffic for all prefixes [67].

One major issue is, thus, to determine how to select the top receivers for the TE optimization process, assuming the traffic demands are heterogeneous. One popular method adopted for tracking top receivers is to measure the total amount of traffic sent to each destination during an interval of time and compute the flow or aggregate rankings on the basis of the contribution of each flow to the total volume of traffic [187, 188]. A ranking R is an ordered set of flows or aggregates as found in definition (5.5). Both the flows and destinations, whose contribution to the total traffic volume is below a specified threshold T, i.e., $V_{norm}(.) < T$, are eliminated. The remaining flows or aggregates are considered to be top receivers.

Definition 5.5: *Given a set of k flows or aggregates represented by a vector* $F = (f_1, ..., f_k)$*, R is a ranking of flows iff* $\forall f_i, f_j \in F: \ r_i \succeq r_j \Leftrightarrow V_{norm}(f_i) \leq V_{norm}(f_j)$*, where* r_k *is the rank of flow* f_k*, and* $V_{norm}(f_k)$ *represents the traffic in terms of bytes for the flow* f_k *normalized by total volume of traffic.* $r_j = 1$ *is the highest rank of a flow, and therefore the flow that contributes the most volume of traffic.*

The Figures in 5.17 show the results of flow ranking for GÉANT data trace. One the basis of these results, it can be verified that the trace is composed of heterogeneous flows, and that the traffic volumes $V(i)$ are roughly consistent with Zipf law, i.e, $V(i) = c.r_i^{-\alpha}, \alpha > 0, c = constant$, with $\alpha = 1.5818$ and $c = 9.6634E^{10}$. It should be noted that to fit Zipf distribution, we first fit the traffic volumes for prefixes with a Power-law distribution using the method described in [189]. Then, we map the Power-law distribution that is found to a Zipf distribution [174]. This implies that a small fraction of prefixes contribute to nearly all the volume of traffic. This traffic volume concentration on a small fraction of prefixes means that there is no need to perform TE on a global scale.

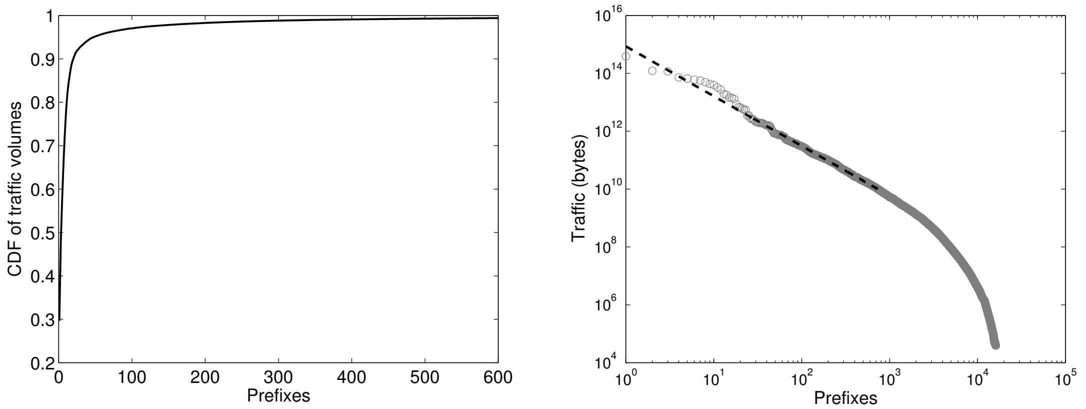


Figure 5.17: GÉANT Traffic: (left) Cumulative Distribution Function of traffic volumes; (right) Zipf's law fitting with $\alpha = 1.5818$ and $c = 9.6634E^{10}$.

The main issue is to define a proper value for the threshold T. The optimal value for T is hard to find as it depends on several factors, such as the trade-off between the overhead on the traffic engineering algorithm, the overhead of routing changes and the level of routing control.

The purpose of this section is not to deal with the issue of how to show the best choice for threshold T in a formal way, but to introduce a practical method for setting the threshold T. This method is based on the errors of common predictors for traffic tracking: (the simple) Last-Value (LV) predictor and the Moving Average (MA) predictor. A prediction error was defined by the difference between the value estimated for the next interval of time and the real value.

More complex predictors could also be employed in our study, by using methods such as AutoRegressive Integrated Moving-Average (ARIMA) (that combines linearly past traffic volumes and/or errors) [128] and Neuronal Networks (NN) (here the basic idea is to train a NN with past traffic volumes in order to predict future values) [129]. However, we only employed LV and MA because there is no advantage in using complex predictors when the performance achieved is almost the same as with simpler predictors [123, 130].

The Figures in 5.18 provide the mean prediction error of each flow in the trace for LV and MA predictors. These results show that the prediction errors depend on the granularity of the traffic flows. These errors grow sharply, (roughly) above the flow 300. This finding introduces a practical bound for the threshold T, since it only makes sense to track "popular destinations" with a bounded prediction error. Applying this prediction error-based criterion to our traffic demands with a maximum error of 100%, 296 destinations were identified (out of a total of 16150 prefixes), and the sum of their individual traffic represents 99% of the total traffic volume.

The interested reader can found further details about this study of a prediction-based criterion for selecting popular destinations in Appendix A or in [190].

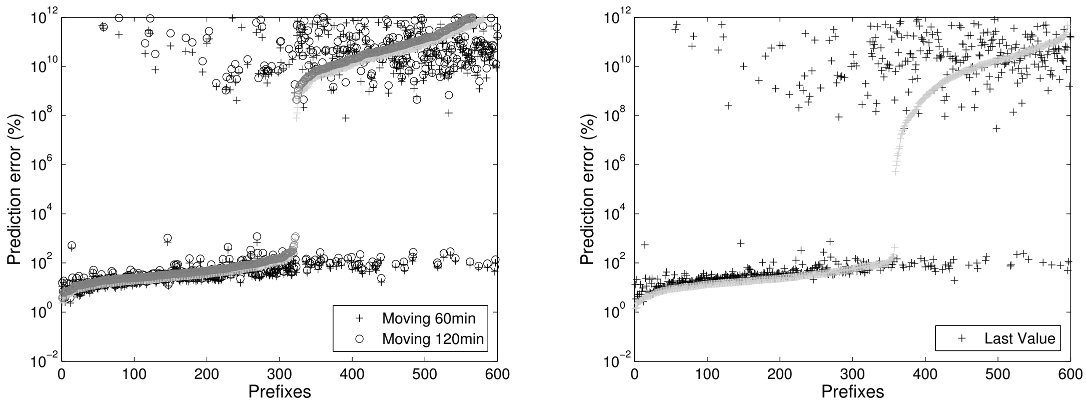


Figure 5.18: Analysis of the prediction errors for Moving Average (left) and Last Value predictors (right).

5.5.3 Network Topology

We use a realistic network model composed of an AS-level topology connected to the GÉANT pan-European network [191]. The AS-level topology aims to represent the whole Internet (viewed from GÉANT) and is modeled as a graph $G = (V, E)$, where

V is the set of ASs and E the set of peering links. The AS-level topology version used in our evaluations is composed of 731 ASs and corresponding peerings, and was constructed from the GÉANT BGP routing table dumps, with only the entries for popular destinations being taken into account. Figure 5.19 details the peering properties of the AS-level topology.

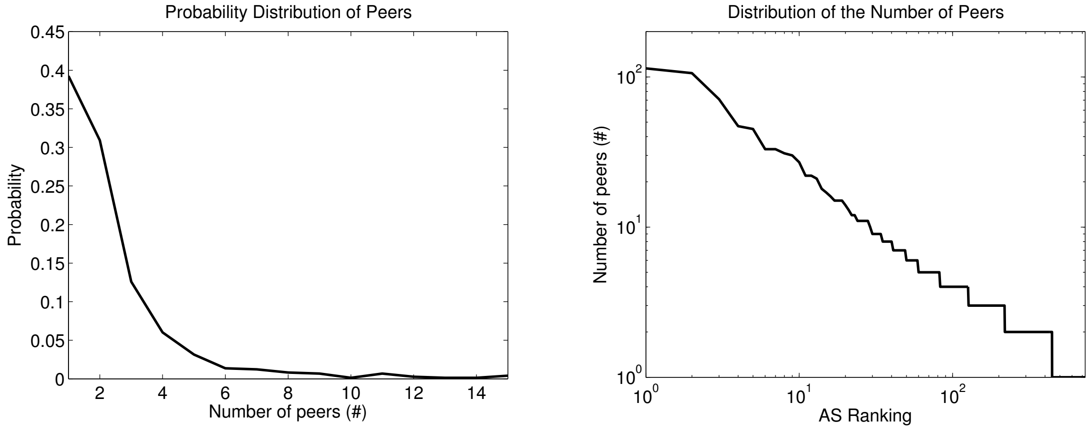


Figure 5.19: AS-Level topology: (left) Probability of a number of peers; (right) Distribution of the number of peers.

For link capacities, assuming splittable traffic, we use a setting obtained from an estimation based on the overlapping of routes in each link and the average traffic volume for each destination, as it is shown in Equation (5.13), where $I_{overprovisioning}$ is an over-provisioning index for the whole network (in our study equal to 1.25), $A_{capacity}$ is the whole network capacity needed to accommodate all the traffic, $x_{fraction}(k)$ is the contribution of flow k to the total traffic aggregate, $I_{maxoverlapping}(k)$ is the maximum number of overlapped paths for flow k (i.e., the number of routes connecting the GÉANT to the destination k), and $I_{overlapping}(k)$ is the number of overlapped paths for flow k in the local link i.

$$c_i = I_{overprovisioning}.A_{capacity}.\sum_{k=1}^{296}\left(x_{fraction}(k).\frac{I_{overlapping}(k)}{I_{max\ overlapping}(k)}\right) \tag{5.13}$$

In the next stage, we approximated the found capacities c_i to the standardized Optical Carrier transmission rate levels (OC) used in backbone Internet lines based on Synchronous Digital Hierarchy (SDH) (whose rates are given by OC-n = 51.84 Mbps x n, where 51.84 Mbps is the base OC rate level) [192]. The propagation delays and queue sizes were chosen heuristically.

Finally, we assume that the latency and available bandwidth of a link are a function of its load. Specifically, we use a M/M/1 link latency function, which belongs to the general class M/G/1 [193]. For a queue M/M/1, the queueing delay d_i at a link i of capacity c_i can be expressed as in Equation (5.14), where ϵ_i is small positive value and $prop_i$ is the propagation latency on the link i. To avoid discontinuity when the load

x_i approaches the capacity c_i, we assumed $l_i(x_i) = d_{max} = 1/\epsilon_i$ and d_{max} simulates a small finite buffer (where d_{max} is the maximum queueing delay).

$$l_i(x_i) = \begin{cases} \frac{1}{c_i - x_i} + prop_i, x_i \leq c_i - \epsilon_i \\ d_{max} + prop_i, x_i > c_i - \epsilon_i \end{cases} \tag{5.14}$$

On the other hand, to estimate the available bandwidth of a link i, A_i, we considered the function in Equation (5.15), where c_i is the link capacity, and x_i is the actual link load.

$$A(x_i) = \begin{cases} c_i - x_i, x_i \leq c_i \\ 0, x_i > c_i \end{cases} \tag{5.15}$$

5.5.4 Evaluation Methodology and Objectives

The performance of the COOP framework is compared against a provable near future scenario in the Internet, where selfish IRC and TE boxes are widely employed by stub and transit ASs, respectively. More specifically, three IRC-TE combinations were considered in the evaluations, COOP, Selfish, and Only TE, depending on whether the COOP feature and IRCs are switched ON/OFF.

The simulation tests were conducted with the aid of a simulation model coded in MATLAB, a high-level language [173]. In this model, GÉANT employs inter-domain TE, and ASs (representing European countries) connected to GÉANT, employ IRCs. In the results reported, IRC plays at a timescale of a quarter-hour; in turn TE plays at a timescale of 24 hours.

The traffic engineering algorithm employed to solve the Egress Router Selection (ERS) problem is based on a genetic single objective version of [106]. The coded traffic objective is the mMLU (to minimize the Maximum Link Utilization) of GÉANT egress links. However, this is well-suited to being extended to other objectives and multi-objective optimization [170]. In this problem, a gene is the assignment of a single aggregate of traffic flows to an egress point of the GÉANT transit network, and an individual – a chromosome – is a potential solution.

In the evaluations, we configure both boxes to focus on the traffic toward the 296 most popular destinations of GÉANT, that have been previously identified. Finally, we configure IRCs to observe the ITU-T's G.114 recommendation, which suggests – regardless of the type of application – a one-way delay (OWD) bound of 400ms [109].

In the evaluations, four main performance metrics are employed: the number of path switches, the latency (defined as the average of OWD for the traffic), the average of TE performance ratio (defined in Section 5.2), and the FBI index (introduced in Subsection

5.4.3). However, the performance ratio is based solely on the mMLU objective. To complement the analysis, we also use the fraction of traffic shifted by TE in each optimization round, as well as the fraction of traffic shifted by IRC in each window of 1/4 hour using a moving average of 24 hours (i.e., 96 samples).

The objectives of the evaluations are as follows:

(1) Assessment of the ability of COOP to achieve synergistic interactions: This entails evaluating the following requirements:

(i) The TE performance ratio with COOP is better than the one without COOP;

(ii) The latency measured with COOP is approximately the same or no worse than the one without COOP;

(iii) The total number of path switches with COOP is near or smaller than the one without COOP.

(2) Assessment of the performance of the COOP fairness mechanism: To study the ability of the fairness mechanism to distribute the available resources at egress links between the IRC flows, while keeping the TE solution valid. This entails comparing the FBI index obtained for the flows at each GÉANT egress point.

(3) Finding the best COOP tuning: This is undertaken by studying the sensitivity of the performance results of COOP to its input parameters, namely to the parameter α of the fairness mechanism and to the range $[U_{min}, U_{max}]$ of the price mechanism, to find the best tuning heuristically, while meeting a handful of objectives: stability of IRCs, low latency, fairness and obtaining a good TE performance ratio. Again, we compare the latency measured, the total number of IRC path switches, and the performance ratio and FBI indexes obtained for GÉANT.

5.5.5 Results

In the next subsection, we study the performance of the three schemes - COOP, Selfish and Only TE - and the COOP effects. Next we study the traffic fairness provided by COOP and the effects of the chosen range $[U_{min}, U_{max}]$ for the price mechanism.

5.5.5.1 Base comparison and COOP effects

We first contrast the three schemes using the performance ratio as the metric of performance. The left-hand Figure of 5.20 plots the average traffic engineering performance ratio for each optimization round. The time interval between consecutive optimization rounds spans 1 day. It can be observed that COOP out-performs the selfish scheme in all the rounds (after the first optimization). The performance ratio of COOP is in average less than 1.03, while under the selfish approach it is in average 1.09. Moreover, a performance ratio spike of about 1.12 in the 8th optimization round for the selfish approach can be observed.

The results indicate that COOP can reduce the effect of route changes performed by IRCs over the GÉANT network, and provide a performance trade-off between the two extreme scenarios. On the one hand, when the IRC boxes are switched OFF (i.e., for the Only TE case) the traffic engineering ratio is about 1, which shows the high effectiveness of the TE optimization when the traffic demand variations only include the component caused by the traffic sources. On the other hand, when the IRC boxes are switched ON, the traffic engineering performance ratio shows the poor ability of TE to accommodate traffic variations due to the route changes performed by the IRCs.

To supplement previous observation, the right-hand Figure of 5.20 plots the corresponding fraction of the traffic shifted in each optimization round. It can be observed that when the IRCs are switched ON (i.e., for both COOP and selfish schemes) a large fraction of traffic is shifted by the traffic engineering algorithm in each optimization round. In the selfish case, the TE has to shift about 51% of the total traffic volume; however, with COOP this fraction decreases by about 15%. This fact constitutes an extra reason for employed COOP. It leads to more predictable traffic demands over the downstream ASs of GÉANT network, and can enable operators of these ASs to make TE changes with greater confidence and less concern about complexity [67].

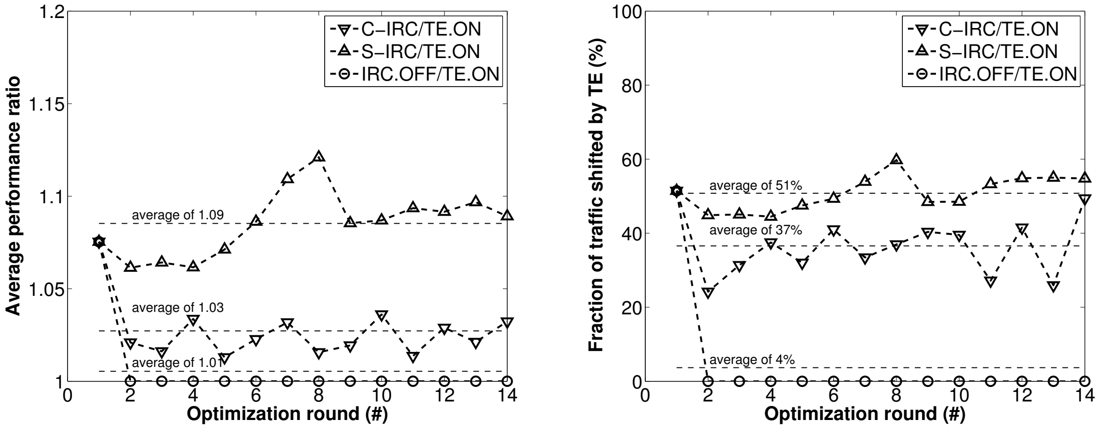


Figure 5.20: (left) Average traffic engineering performance ratio; (right) Fraction of the traffic shifted in each optimization round.

Next we study the performance of the IRC/customer side. The Figures in 5.21 plot (left) the average latency achieved for each destination, and (right) the overall average latency for each optimization round. These results reveal two significant observations. On the one hand, we observe, again, the effectiveness of the IRCs in improving traffic latency performance, by now using this new simulation environment. Almost all the destinations experience an average latency that is much smaller than the maximum tolerated latency of 400ms. In contrast, without the IRCs, a significant fraction of destinations (approximately 60%, i.e., 175 out of a total of 296 destinations) experience an average latency higher than the tolerated bound.

However, we observe that COOP can ensure a traffic performance similar to the one obtained by the selfish scheme. This suggests that the COOP design can reduce the dispute between the IRC and ISP TE traffic goals, and give approximately the same or no worse traffic performance than the one without COOP. When COOP is used, the IRC flows only pay a small performance penalty of about 28ms – COOP obtained an average latency of 68ms, while the selfish approach obtained an average of 40ms. The observed performance penalty is expected as the utility function of COOP-enabled IRCs takes into account the latency that has been measured and the price advertised by the ISP. In other words, this is what is expected in general whenever a win-win approach is employed.

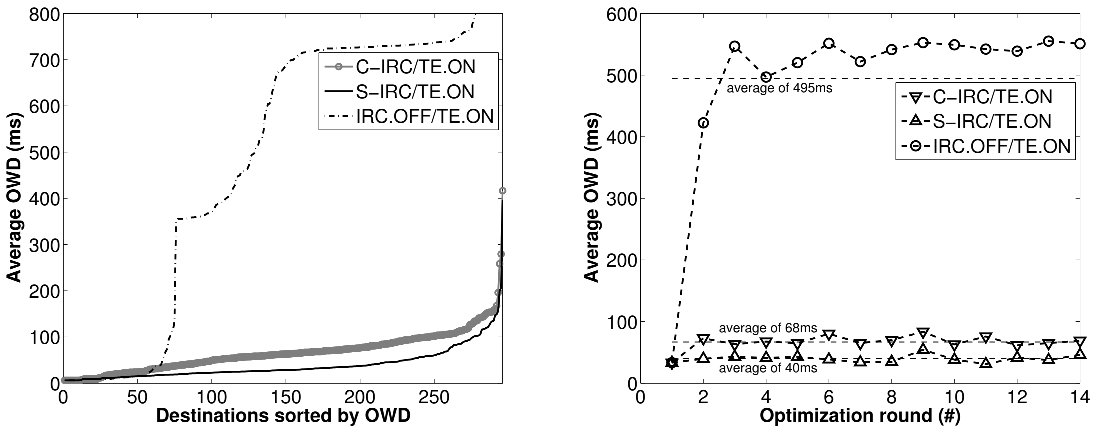


Figure 5.21: (left) Latency measured for each destination; (right) Overall latency measured for each optimization round.

The effects of COOP on IRC stability are discussed next. To evaluate these effects, we compare the number of path switches performed by the IRCs for both schemes (COOP, Selfish). The left-hand Figure of 5.22 plots the samples ordered by the overall number of path switches that have been performed (in a quarter-hour interval). Unfortunately, the actual COOP set-up clearly requires four times more then the number of path switches needed by the selfish scheme (S-IRC/TE.ON) to meet the same OWD constraint.

From previous result, it can be inferred that COOP has a potentially negative effect on the IRC stability apparently on account of the dynamics of the prices advertised by the GÉANT's COOP infrastructure. Regarding this issue, it is also worth noting that that COOP needs to shift roughly double of the average amount of traffic than the selfish approach, as can be observe in the right-hand Figure of 5.22. It needs to switch an average fraction of 7% of traffic, while the selfish approach needs only to shift 3%

of the traffic in each sample. This means that more predictable outgoing traffic at the ISP side, may translate into less predictable traffic demands.

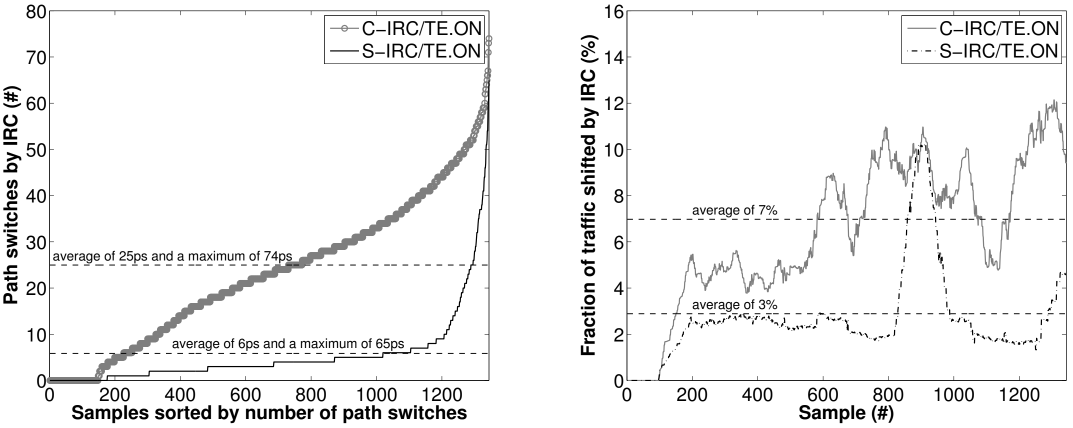


Figure 5.22: (left) Number of path switches performed by IRC in each optimization round; (right) Moving average of the fraction of the traffic shifted by IRC (window size of 96 samples).

To sum up, the results, as a first main conclusion, again substantiate the claim regarding the reduction of the effectiveness of inter-domain TE due to (selfish) IRC route control. The results have shown that inter-domain TE box is able to adapt the routing to traffic demands fluctuations in an efficient way caused by the traffic source dynamics; however, it might be unable to accommodate traffic fluctuations because of the IRCs route changes.

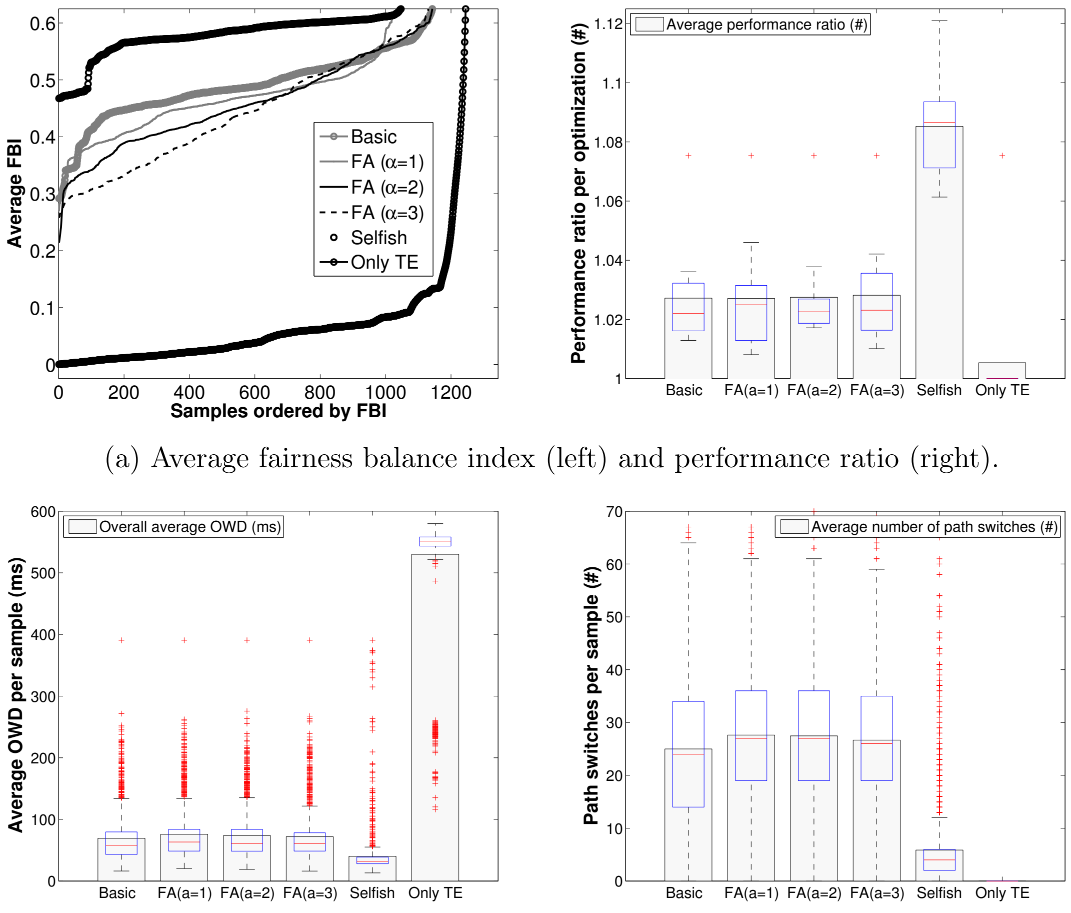
As a second main conclusion, the results have shown that the COOP scheme can meet the two main requirements for producing synergistic interactions between IRC and TE. With COOP, the performance ratio of the GÉANT network is improved and the low traffic latency is almost the same as that the observed for the selfish approach.

Unfortunately, we have observed that the base set-up of COOP is unable to meet the third recommended requirement for achieving synergistic interactions between both mechanisms. Effectively the flaw of COOP is that an IRC may have to pay a serious stability penalty. This result is a driving-force for further analysis in the next subsection, which arises from the need to find the best tuning to allow COOP to tackle this flaw.

5.5.5.2 Traffic fairness and COOP tuning

COOP is designed to provide fairness among traffic flows or aggregates that are sharing an egress point of a transit network. Section 5.4.3 introduced the COOP fairness mechanism, the goal of which is to ensure the (ideal) mixture of traffic as defined by the optimal TE solution. However, the fairness is only achieved gradually and depends on the α parameter, hence, COOP is not able to “jump” to the optimal mixture of traffic, immediately after the response of IRCs to the path performance changes originated by the last TE optimization.

Our attention is thus focused on the potential improvements indicated in the results provided by the proposed COOP fairness mechanism. More precisely, in the first step, we examine the Fairness Balance Index (FBI) improvements as a function of the α parameter. Top left-hand Figure 5.23 shows these results. In this figure, FA (Fairness-Aware) refers to COOP which is endowed with the fairness mechanism and Basic refers to the original version. It is evident that as the parameter α increases, the FBI values for FA COOP are improved. This analysis shows the effectiveness of the COOP fairness mechanism. The remaining metrics (performance ratio, latency and number of path switches) shown in the top right-hand figure and bottom Figures of 5.23 do not differ significantly. It is important to note, however, that when contrasting FA COOP with the Basic scheme, an extra penalty ($\sim$ 5%) is apparent in the stability of the IRCs, which exacerbates the stability problem identified previously.



(a) Average fairness balance index (left) and performance ratio (right).

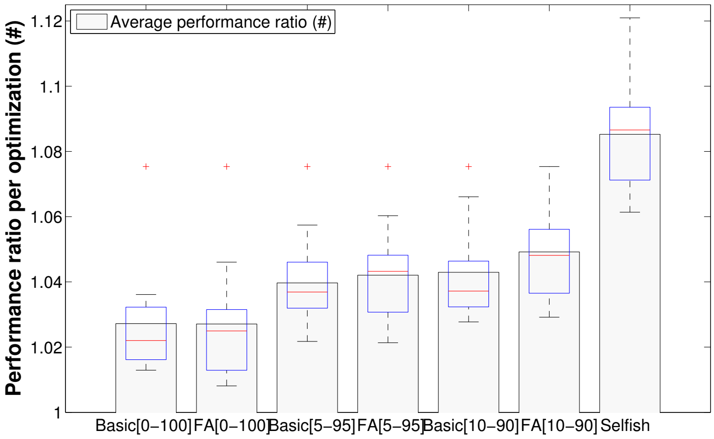
(b) Average OWD (left) and path switches (right).

Figure 5.23: Comparison of the FA COOP with the Basic COOP, Selfish and Only TE scenarios.

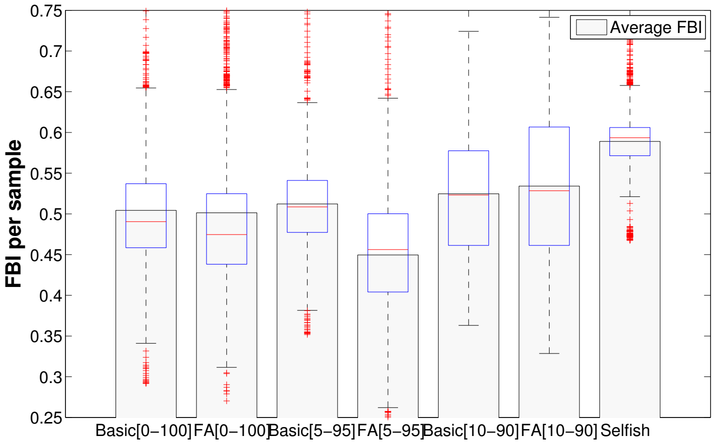
We argue that previous IRC stability problem can be explained by the dynamics of the prices advertised to the IRCs. For this reason, in a second step, we examine the sensitivity of the results to the utilization range $[U_{min}, U_{max}]$, used by the fairness mechanism to find the best tuning, while avoiding the excessive IRC oscillations caused by price dynamics. There is a trade-off in choosing between the different values of U_{min} and U_{max} for the price mechanism. When the value of U_{max} is high (i.e., $U_{max} \rightarrow 100$), and the value of U_{min} is low (i.e., $U_{min} \rightarrow 0$), the aggressiveness of the COOP price mechanism to deter/attract traffic for ensuring optimal utilization is weak. However, this behavior leaves room for price oscillation when the current utilization is very high or too low. At the same time, choosing a lower value for U_{max} and a higher value for U_{min} leaves less room for price oscillation, since the aggressiveness of the price mechanism is strong. However, this behavior can interact with the fairness mechanism whenever the target optimal utilization is high or low.

To evaluate the effects of the $[U_{min}, U_{max}]$ range, the three schemes are compared again. The Figures in 5.24 plot the performance ratios and FBIs. In turn, the Figures in 5.25 plot the results at the IRC side, namely the latency and number of path switches. These results suggest that by shrinking the $[U_{min}, U_{max}]$ range, the IRC stability is improved with no traffic or performance ratio penalties. This is particularly evident when the range $[5, 95]$ is employed for $[U_{min}, U_{max}]$. However, when observing the FBI index more closely, we found there is a penalty in the traffic fairness. Thus, we recommend the ISP operator so that some operational experience can be obtained to tune the range $[U_{min}, U_{max}]$ properly. Actually, it will be worth devising a mechanism that can dynamically adapt this range to traffic, since we believe the effects of changing this range depends on the traffic characterization of the network.

To conclude, on the basis of this analysis when choosing the proper range for $[U_{min}, U_{max}]$ *(in our evaluations the range* $[5, 95]$*), the COOP framework can definitively produce synergistic interactions between in IRC and inter-domain TE boxes, – as was pursued in this chapter.*

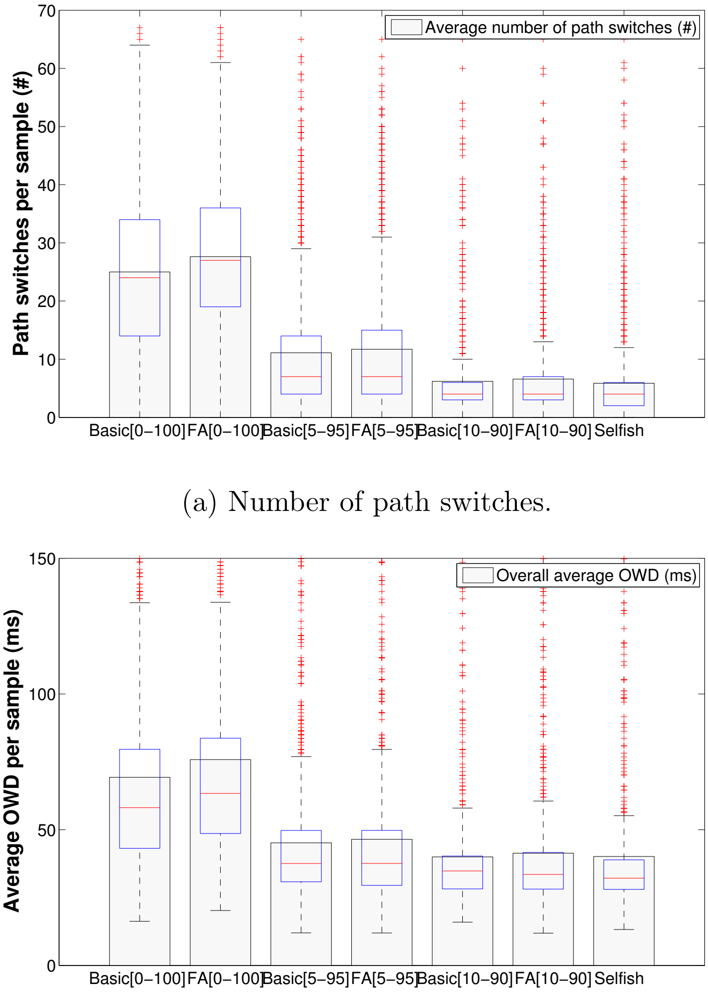


(a) Performance ratio.



(b) Fairness balance index.

Figure 5.24: Comparison of the FA COOP with the Basic COOP, Selfish and Only TE scenarios for different $[U_{min}, U_{max}]$ ranges [ISP side].



(a) Number of path switches.

(b) Average OWDs.

Figure 5.25: Comparison of the FA COOP with the Basic COOP, Selfish and Only TE scenarios for different $[U_{min}, U_{max}]$ ranges [IRC side].

5.6 Summary

The interaction that may occur between Intelligent Route Control (IRC) and inter-domain Traffic Engineering (TE) boxes is an important concern in routing control, since it can jeopardize the traffic performance achieved by either of them.

In this chapter, the problem of the interaction between these boxes was identified by carrying out extensive simulations, and a cooperative solution was proposed to minimize their disputes in the routing control. The devised solution is a novel ISP (Internet Service Provider)-friendly IRC Cooperative Framework (COOP), and its challenge is to combine both mechanisms with a view to producing synergistic interactions, while keeping the existing IRC and TE control platforms almost intact.

The novelty of the COOP framework is the employment of a feedback-based method where both boxes cooperate to control how traffic enters the transit network, while ensuring that the network performance approaches an optimum level. In particular, this mechanism supports the exchange of feedback signals between traffic engineering and intelligent route control, which notify the IRC controllers about route price modifications that result from traffic load changes over the ISP transit network.

By employing a utility-based algorithm, an IRC can integrate ISP feedback information into its routing decision process, and seek finding a *social* optimal routing, that respect both individual stub and transit traffic preferences. Significant features of COOP include the use of out-BGP(Border Gateway Protocol)-band signaling and the agnosticism of the feedback information to the ISP traffic objectives. This first feature avoids overloading the current routing system with extra functions. The second feature ensures that the common operational constraint of transit networks in limiting the disclosure of detailed information related to the networks and their traffic objectives can be fulfilled.

We designed and applied COOP for a representative transit network, the GÉANT pan-European academic network. The performance results were analyzed and demonstrated that COOP achieves synergistic interactions between IRC and inter-domain TE. With COOP, the performance ratio of the transit network is improved, and the traffic latency and IRC stability is almost the same as that the observed for the selfish approach.

Chapter 6

Conclusion and Future Work

This chapter summarizes the contributions of this thesis and discusses some of the key unresolved issues. This chapter is divided as follows. Section 6.1 provides a synthesis of the thesis. Section 6.2 describes the major contributions that resulted from the work done. Following this, Section 6.3 describes the limitations of this research. Finally, Section 6.4 contains some of the main issues that should be addressed in future work.

6.1 Synthesis of the Thesis

As the Internet is becoming an increasingly important part of the worldwide communications infrastructure, the interconnection of Autonomous Systems (AS) with Quality of Service (QoS) has emerged as a primary goal, where routing plays a key role. However, the classical approach to the problem of inter-domain QoS routing, which is based on the proposals of new extensions to Border Gateway Protocol (BGP), is no longer sustainable, in view of the numerous changes required by the routing infrastructure and the need for consensus on a global scale. This consideration led to the main objective of this thesis, which was the investigation of an alternative solution to problem of the inter-domain QoS routing.

The study entailed identifying of the main drawbacks of the BGP protocol, conducting a survey of the main research studies of inter-domain QoS routing and engaging in a discussion surrounding the complex issue of extending or replacing BGP for providing QoS routing across AS boundaries, as outlined in Chapter 2. This has laid the groundwork for addressing the need to explore and conceive a lightweight inter-domain route control strategy.

As well as dealing with these issues, the primary goal of Chapter 3 was to investigate matters related to the Intelligent Route Control (IRC) strategy. The full-design of an IRC system was provided, together with an analysis of the benefits and feasibility of intelligent route control. The strengths of the IRC strategy and the most important algorithms that could be employed to fully support intelligent route control functionality, were evaluated by means of extensive simulations.

In view of the fact that the conventional standalone and selfish IRC model may introduce instability in the network, especially in competitive environments, the aim of Chapter 4 was to move from this model to a standalone and social route control model. A Social Route Control (SRC) algorithm was provided to deal with this problem of instability, which is supported by an adaptive cost metric and a two-stage filtering process. The strengths of this strategy were evaluated by means of extensive simulations.

The analysis of the interactions between IRC and backbone Traffic Engineering (TE) led to the conception of an ISP (Internet Service Provider)-friendly IRC Cooperative Framework (COOP), which was designed to tackle this problem. COOP was outlined in Chapter 5 and was established to enable both IRC and TE middleboxes to cooperate in a closed-loop, while also ensuring that they fulfill the local traffic objectives. The COOP architecture, algorithms and a signaling protocol were also discussed. The ability of COOP to ensure synergistic interactions between both traffic control middleboxes was evaluated by means of extensive simulations with the aid of a realistic simulation model.

6.2 Major Contributions

This section describes the two major contributions of this thesis. First, we showed that Intelligent Route Control (IRC) provides an effective and practical means of improving Quality of Service (QoS) across Autonomous Systems (ASs) boundaries and then outlined the design of an IRC system from scratch. To the best of our knowledge, this is the first study that addresses a full range of design issues regarding the functional architecture of an IRC system, and includes path monitoring and path switching mechanisms.

At the architectural level, we decomposed an IRC system into two core components, a Monitoring and Measurement Module (MMM) and an intelligent Route Control Module (RCM), which had corresponding sub-components to leverage the modularity of the IRC system, that was not only aimed at reducing the complexity but also at allowing generality. For instance, new IRC path switching algorithms or path monitoring methods can be introduced to extend the IRC functionality without affecting other mechanisms or components. We evaluated most of the parts of our design by means of simulation models, and the results showed the benefits and feasibility of the IRC paradigm, and that the devised mechanisms for path monitoring and path switching yield good performance.

We also studied the stability of IRC middleboxes and proposed mechanisms to improve the stability of these boxes. The main mechanism is based on a self-adaptive cost metric. The performance results showed that this mechanism introduces a "kind" of social route control, as it reduces the penalties associated with frequent traffic relocations, while providing almost the same – and in several cases an even better – end-to-end traffic performance.

As the second major contribution, we established an ISP (Internet Service Provider)-friendly IRC COOPerative Framework (COOP), since we were aware that IRCs can have a negative impact on the performance of a transit network achieved by inter-domain Traffic Engineering (TE) with BGP (Border Gateway Protocol). The novelty of this framework is that it includes a cooperative scheme between IRC and backbone traffic engineering middleboxes. In addition, a feedback-based architecture for COOP was designed and evaluated. This is based on a cooperative strategy between direct neighbors, and is, as far as we know, the first strategy to tackle the problem of the interactions between stub networks employing IRC and transit networks employing an inter-domain traffic engineering technique. The performance results of COOP were analyzed and demonstrated the feasibility of this framework. A notable feature of the results was that they showed that this framework can produce synergistic interactions between IRC and TE middleboxes.

6.3 Limitations

One of the underlying reasons for carrying out this work is to come up with an engineering design that breaks away from the traditional approach (which involves proposing extensions to routing protocols), but has the potential to shape the currently used inter-domain routing infrastructure. We hope to encourage the use of Intelligent Route Control (IRC) – with a greater focus on end-user networks or applications – on a large scale, although whether the market will adopt our IRC design is, of course, beyond our control. Similar observations could be made about the ISP (Internet Service Provider)-friendly IRC COOPerative framework (COOP). However, although it is similar to the IRCs, in so far as it needs few changes in the BGP (Border Gateway Protocol) infrastructure, it requires cooperation between the IRCs and inter-domain TE (Traffic Engineering) middleboxes and some important changes to the traditional traffic engineering paradigm, and ISP monitoring platforms. The COOP framework introduces a hybrid approach that is dominated by the off-line traffic engineering, but blended with on-line components to respond to variations in traffic demands.

Yet, it can be argued that this work has proved to be a worthwhile experiment. It is a proof-of-concept that is "technically doable" since it provides the user with the ability to select best performing inter-domain paths. It is hoped that in the future, if the Internet market welcomes user choice, our work can serve as a valuable guide.

6.4 Future Work

This thesis attempts to address a full range of design problems that arise from supporting an Intelligent Route Control (IRC) paradigm and the ISP (Internet Service Provider)-friendly IRC COOPerative framework (COOP). Inevitably, some areas of this work are still in a provisional state, and require further study on the part of other researchers. Yet six key unresolved issues are worth highlighting.

The first aspect that needs further investigation is the path switching dynamics of IRCs, especially in competitive environments. Ideally, this should be sought in models without any dependence on the details of the IRCs. We believe that as a first step, the stochastic distribution of path switches will help to introduce refinements or new mechanisms for the IRCs. Moreover, further studies should attempt to gauge the accuracy of the probing mechanisms, and determine the path switching mechanisms on the basis of other Quality of Service (QoS) metrics (not limited to latency and available link bandwidth).

The second aspect is the control of incoming traffic with IRC. Our joint work with the Technical University of Catalonia in [152] was a first step in addressing this issue, and showed that the IRC model can improve the distribution of incoming traffic. Further research in this field is essential. It is foreseeable that a COOP version adapted to single pairs of remote IRCs will be used, since in some ways it provides a means of controlling incoming traffic. After adapting the COOP to remote IRCs, we plan to carry out extensive simulations in which the behavior of our cooperative framework is contrasted with the current incoming traffic control techniques (e.g., AS-path prepending). The results collected will enable an evaluation to made of the strengths and limitations of our framework.

The third aspect that requires further study concerns the modeling of interactions between the IRC and TE middleboxes. The use of Game Theory is envisaged as a promising tool to devise a valid model in a formal way, not only to analyze the interactions, but also to study the performance of the subsequent proposals [194]. In fact, routing can be modeled as a game between the IRC and TE users in which routing decisions are made solely on the basis of local traffic objectives, and thus their strategy.

The fourth aspect concerns with the evaluation of the IRC systems and the COOP framework. In Chapters 3 and 4, we conducted our IRC evaluations in a simulation environment that was based on an event-driven paradigm. In the future, we would like to test our full IRC design in the realistic environment provided in Chapter 5, removing the features previously associated with COOP. The COOP framework evaluations were conducted in this realistic environment, although it only gives a perspective of the performance of COOP from a single transit network. As well as this, a deeper study of the timescales separation is required.

The fifth aspect that needs improvements is the COOP fairness mechanism and the utilization range used within the price adjusting mechanism. We designed a mechanism for the COOP framework to improve the IRC flows fairness and then conducted experiments to demonstrate its effectiveness. However, we found that this mechanism may have an adverse effect on global achievements of the COOP framework. Similarly, we found that the chosen utilization range for the price adjusting mechanism may have an impact on the IRC stability. In short, even though we found a good tweaking for COOP, we believe the reported results can be significantly improved if other kinds of network models are used. Our current simulation set-up has the limitation that all IRC systems are located above the GÉANT pan-European network; in addition the distribution of the GÉANT traffic itself has most of the stable traffic volume condensed into 30-50 top destinations, resulting in a reasonable number of flows with relatively high dynamics.

Finally, the overall COOP performance should be contrasted with emerging traffic engineering frameworks such as those based on the concept of optimal oblivious routing, where the TE box seeks to balance the performance across all the possible traffic demands [180, 195]. We believe that COOP has the potential to outperform the latest approaches in competitive environments, because instead of just the dynamics of the traffic source, the route changes are the main factor that is responsible for the most significant changes in traffic demands in the Internet [118], and COOP was precisely designed to address this issue, in particular to sustain the aggressive route changes caused by stub ASs employing IRCs. As well as this, a deeper study of the potential performance gains of COOP is required, if different degrees of biasing the TE objectives in favor of end-user networks or applications traffic objectives are employed.

Appendix A

A Prediction Criterion for Selecting Popular Destinations

Inter-domain Traffic Engineering (TE) has become an important part of today's network management systems to provide best inter-domain routing, while ensuring a high level of global network performance [67,196]. This is particularly important to generate cost savings and to face several challenges posed by the competitive Internet market, such as the provision of value-added services.

The inputs to the inter-domain traffic engineering process are the traffic demands over the network, the available egress point choices for for each reachable prefix destination (provided by BGP (Border Gateway Protocol) [32]) and the egress point capacities. The output is the inter-domain routing so as the traffic objectives can be satisfied. Lastly, the optimal routing is translated to a careful tuning of the BGP routes attributes [63].

One of the common optimization problems that has to be dealt with by the network managers is, thus, the Egress Router Selection (ERS) [165]: *how should the traffic demands be assigned to multiple egress points, to ensure that the transit network's traffic objectives are fulfilled (e.g., minimizing the maximum link utilization (mMLU) or Load-balancing (LB))*? Studies, such as [166,197], provide some instances of algorithms which can solve the problem of ERS efficiently.

The biggest concern with ERS is that with regard to the number of traffic flows, objectives and egress point choices, the task of optimizing the routing can become computationally hard and entails a large number of route changes. One effective means of reducing these overheads is to focus the traffic engineering optimization on popular destinations (also known as top receivers), and thus being able to shift a large volume of traffic with a small number of path switches, rather than shifting the traffic for all the prefixes [67].

However, there are two major issues that have to be tackled; these are the selection of popular destinations, and the prediction of the amounts of traffic that will be sent to each destination during the next interval of time. The selection of popular destinations is of critical importance to alleviate the complexity of the traffic engineering process, which has an inherent problem of lacking scalability. Tracking the traffic accurately is a significant issue because the network traffic is bursty [198], which means that weak predictions may lead to spurious traffic changes or congestion over egress links or downstream Internet Service Providers (ISP).

This appendix correlates both issues and sets out a practical criterion to define a threshold for traffic volumes, the value of which is used to categorize the popularity of the traffic destinations. This is based on the analysis of errors of common predictors for traffic tracking, namely (the simple) Last-Value (LV), the Moving Average (MA), and the Low-pass Exponential Moving Average (LpEMA). The results show that by applying this criterion, we can reduce the number of target prefixes to a small fraction of the total number, while ensuring stability in the traffic engineering, which results from an effective predictability of the traffic headed toward these prefixes.

The rest of this appendix is divided as follows. Section A.1 outlines the whole inter-domain traffic engineering process, describes the concept of Zipf's Law and its significance for traffic engineering, and states our problem. Sections A.2 describes the data trace and analyzes the performance of different traffic predictors to postulate a criterion for selecting popular destinations. Finally, Section A.3 concludes this appendix.

Bibliographical Note. This appendix will appear as a paper in [190].

A.1 Background and Statement of the Problem

This section sets out by describing the traffic engineering process. We then underline the importance of the consistency of traffic demands with the Zipf's law for the traffic engineering. Lastly, we describe the problem of selecting the popular selections, and provide our proposal.

A.1.1 Basics of Inter-domain Traffic Engineering

The goal of inter-domain traffic engineering is to optimize inter-domain routing, subject to a given traffic objective and the constraints of the network. For the sake of illustration, we consider as traffic objective, the minimization of the maximum link utilization (mMLU) of the egress links. A transit network to be optimized is represented by a set of ingress points I and a set of egress points E.

The inputs to the TE algorithm are the incoming traffic demands (TD) over the transit network, the available egress point choices for each reachable prefix destination

$p \in P$ (provided by BGP) and the egress point capacities. The predicted TD are represented by the matrix $D = \{d_{ip} \mid i \in I, p \in P\}$, where each entry d_{ip} is the demand for the ingress point i - destination p pair. The set of egress point capacities are represented as C, where each entry $c(e)$ is the capacity at the egress point $e \in E$. An inter-domain routing is represented by ϵ, where each entry $\epsilon_{iep} \in \{0, 1\}$ is an indicator function that tells whether the d_{ip} is assigned to the egress point e.

The output of the inter-domain traffic engineering process is the set of optimal routes to achieve the optimal mixture of traffic. In turn, these results are translated to a careful tuning of BGP routes attributes. For the interested reader, the set of techniques (e.g., `LOCAL-PREFs` tuning) that can be used for egress traffic control are described in [63].

The TE problem is known as the egress router selection (ERS): *how to assign each entry of traffic demands d_{ip} to an egress point e, so as to optimize a certain traffic objective.* The traffic objective that is encoded in the ERS problem, is to minimize the whole MLU (mMLU) at egress links. To be more specific, we introduce the definitions (A.1) and (A.2).

Definition A.1: *The link utilization of e for a routing ϵ is defined as the traffic to capacity ratio as shown in Equation* (A.1.1).

$$U_e = \sum_i \sum_p \frac{\epsilon_{iep} d_{ip}}{c(e)}, \epsilon \textit{ is a routing.} \tag{A.1.1}$$

Definition A.2: *An optimal inter-domain routing for a given D, is the routing that minimizes the maximum link utilization (min-MLU), as shown in Equation (A.1.2), where OU is the optimal utilization.*

$$OU = min\ max\ U_e, \forall e \in E. \tag{A.1.2}$$

The traffic objective mMLU is subject to constraints (A.1.3) and (A.1.4). The capacity constraint (A.1.3) ensures that the total resource requirements of the traffic flows assigned to each egress point do not exceed the available/contracted capacity. The assignment constraint (A.1.4) guarantees that each traffic flow is assigned to exactly one egress point e.

$$\sum_i \sum_p \epsilon_{iep} d_{ip} \leq c(e), \forall e \in E \tag{A.1.3}$$

$$with, \sum_{e} \epsilon_{iep} = 1, \forall i \in I \tag{A.1.4}$$

A.1.2 Zipf's Law and its Significance for Traffic Engineering

Zipf's law is an empirical law introduced to describe the popularity of words in terms of rank and their frequency in use [174]. It states that if f_i is the frequency of the word i and r_i its rank order, then $f_i \simeq \frac{1}{r_i}$. This implies that the n-th word is used twice as often as the $2n$-th word. Visually Zipf's law can be easily observed by plotting the data set on a log-log scale, with the axes being $x = log(rank\, order)$ and $y = log(frequency)$. If the plot is (almost) linear, we say that the data set is consistent with Zipf's law. When using Zipf's law in general contexts, such as the popularity of web pages or traffic destinations as in this appendix, it can be re-formulated as: if f_i as a function of r_i is consistent with a power-law distribution it is referred to as Zipf's-like.

A typical backbone network has routes for more than 150000 prefixes [67]; as a result optimizing the routing to accommodate all the traffic demands that were detected by network monitors, can become computationally hard and entail a large number of route changes. The extrapolation of this concept to the field of traffic engineering is therefore significant.

When traffic demands of Internet traces (in terms of traffic volumes) are consistent with the Zipf's-like distribution, it implies that a small fraction of prefixes (i.e., about 5 - 10%, though in our trace about 2% of the total number of prefixes (see Sect. A.2)) of the total number of receivers contribute to most of the total volume of traffic. In practice, several studies have identified this phenomenon in Internet traffic [133, 188] which means that it is not necessary to consider all the destinations in the routing optimization process. It will be enough to change `LOCAL-PREFs` for a small handful of popular destinations, that would move a large fraction of traffic from one egress point to another, while minimizing the number of path changes. Moreover, these destinations tend to have more stable traffic volumes [133].

The process of verifying the consistency of traffic demands with Zipf's law proceeds as follows. First, the traffic demands are acquired from measurements taken from a network, i.e., the total amount of traffic sent to each destination is recorded during an interval of time [187, 188]. Subsequently, the flow rankings are computed on the basis of the contribution that each flow makes to the total volume of traffic. A flow ranking R is an ordered set of flows as shown in definition (A.3). The consistency of traffic demands with a Zipf's-like distribution is evaluated according to definition (A.4) or visually by plotting the data on a log-log scale.

Definition A.3: *Given a set of k flows or aggregates represented by a vector $F = (f_1, ..., f_k)$, R is a ranking of flows iff $\forall f_i, f_j \in F: \; r_i \succeq r_j \Leftrightarrow V_{norm}(f_i) \leq V_{norm}(f_j)$, where r_k is the rank of flow f_k, and $V_{norm}(f_k)$ represents the traffic in terms of bytes* *for the flow f_k normalized by total volume of traffic. $r_j = 1$ is the highest rank of a flow, and therefore the flow that contributes the most volume of traffic.*

Definition A.4: *Given a set of k flows $F = (f_1, ..., f_k)$, and a ranking R, these are consistent with a Zipf's-like distribution, iff $\forall f_i$ the corresponding traffic volume $V(i)$ satisfies the relation $V(i) = c.r_i^{-\alpha}, \alpha > 0, c = constant$.*

A.1.3 Problem Statement, and Challenge

As a network/ISP grows, a commonly-used traffic engineering practice is to select the popular destinations of the traffic and optimize the traffic performance mainly for them, as previously described. However, there remains an important issue to address which is, *how to define a proper value for a threshold (said T) that splits the whole set of destinations into popular and non-popular destinations*. This means that after a ranking R has been computed, both the flows and destinations, whose contribution to the total traffic volume is below some specified threshold T, i.e., $V_{norm}(.) < T$, should be pruned from the optimization process of the network. The remaining flows are considered as top receivers.

Unfortunately, the optimal value for T is hard to find as it depends on several factors, such as the trade-off between the overhead on the traffic engineering algorithm, the overhead of routing changes and the degree of routing control. The purpose of this appendix is not to deal with the issue of showing formally what is the best choice for threshold T, but rather to provide a practical method for setting the threshold T.

In this Appendix, we postulate that the threshold T can be empirically found through the analysis of the errors of the predictors that are used for traffic tracking. A prediction error is defined as in definition (A.5).

Definition A.5: *Given a flow a prediction error is the difference between the value of the estimated traffic volume e_i for the next time slot i, and the real value V_i, i.e., $error_i$=$|e_i - V_i|$.*

A.2 Prediction Criterion for Selecting Popular Destinations

Our thesis is that the analysis of the whole behavior of the prediction error in tracking traffic leads to a practical criterion for selecting popular destinations. In this section we, thus, study the performance of different predictors (LV, MA and LpEMA). In evaluating the performance of the predictors, their mean error was computed and analyzed.

A.2.1 Data Trace

We use one set of data traces that was collected at the GÉANT pan-European academic network in 2005 [186]. This was conducted by means of the Cisco Netflow feature [126]. The Netflow measurements were carried out over a period of two weeks, as shown in Figure A.1. Each sample in the figure represents the amount of bytes seen during each interval of 15 minutes, multiplied by 1000, because the Netflow sampling was performed at a rate of 1/1000. After this, the traffic was divided into demands. A demand represents a given amount of traffic (in bytes) from GÉANT's users to a destination.

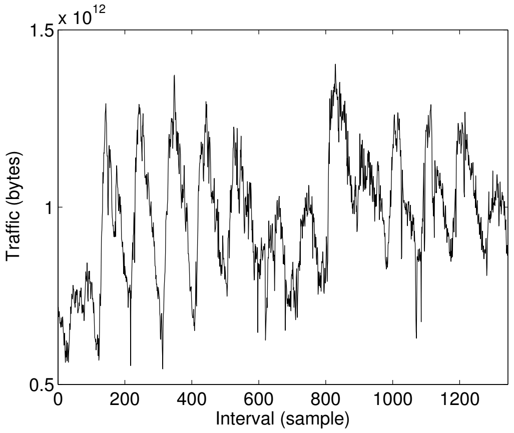


Figure A.1: GÉANT Traffic: (left) Traffic evolution from Sunday (05-07-10) to Saturday (05-07-16).

A.2.2 Tracking Traffic

In this study, we track traffic with three distinct predictors which vary in degrees of complexity: a very simple predictor, the Last Value (LV), the classical Moving Average (MA) , and an adaptive but more complex predictor, the LpEMA (Low pass Exponential Moving Average), defined below.

LV (Last Value). LV is the basic predictor. The actual traffic estimate e_i is equal to the traffic volume V_i measured in the last time slot i, i.e., $e_i = V_i$.

MA (Moving Average). MA is another very simple predictor. The actual traffic estimate e_i is equal to the arithmetic mean of the traffic volumes measured in the last n time slots, i.e., $e_i = \frac{\sum_{k=0}^{n-1} V_{i-k}}{n}$. This predictor requires a set of n traffic volume measures, and thus a window of size n. The biggest problem of this predictor is how to find the right size for the sliding window. Very large windows result in smoothing too much the real traffic changes. In contrast, small windows result in that fast traffic changes are not suppressed.

LpEMA (Low pass Exponential Moving Average). LpEMA is an extension of the classic EMA predictor, and is a more complex predictor than LV and MA. To compute the actual metric estimate e_i, the LpEMA combines the previous estimate e_{i-1} with the actual traffic volume measure V_i using an adaptive Exponential Moving Average, as shown in Equation (A.2.5), where α_i is an adaptive exponential weight, which is calculated by using the classical formula for low pass filter, m_i is the gradient between two metric samples (i.e., $\frac{V_i - V_{i-1}}{t_i - t_{i-1}}$), and m_{norm} is the normative gradient calculated over a given time window (e.g., 10 times the interval $t_i - t_{i-1}$). In contrast to the original EMA, it makes use of an adaptive exponential weight α, since with large weights the estimation follows the measurement exactly, but does not suppress fast traffic changes, whereas with small weights, the traffic changes are suppressed but the estimation follows the real changes too slowly [144].

$$\begin{cases} e_i = (1 - \alpha_i)e_{i-1} + \alpha_i V_i \\ \alpha_i = \alpha_{max} \frac{1}{1 + \frac{|m_i|}{m_{norm}}} \end{cases} \tag{A.2.5}$$

More complex predictors could also be employed in our study, such as Auto-Regressive Integrated Moving-Average (ARIMA) (that combines linearly past traffic volumes and/or errors) [128] and Neuronal Networks (NN) (here the basic idea is to train a NN with past traffic volumes to predict future values) [129]. We only employed LV, MA and LpEMA in the belief that there is no advantage in using complex predictors given the fact that the performance achieved is almost the same as with the simpler predictors [123, 130].

A.2.3 Analysis of Prediction Errors

We start by verifying the consistency of the GÉANT's data trace with Zipf's law using the procedure described in Section A.1.2. After the completion of the process, the right-hand graph of Figure A.2 shows that the trace is roughly consistent with the Zipf distribution with $\alpha = 1.5818$ and $c = 9.6634E^{10}$. This observation implies that a small fraction of the prefixes of the GÉANT data trace contribute with most of the total volume of traffic; this allows us to proceed with the analysis and select GÉANT's popular destinations. It should be noted that to fit the Zipf distribution, we first fit the traffic volumes with a Power-law distribution by using the method described in [189]. Then, we map the Power-law distribution that has been found in a Zipf distribution [174].

We next evaluate the performance of the LV, MA and LpEMA. Figures A.3 and A.4 provide the mean prediction error of each flow in the trace for different sizes of the sampling window. In the case of the LpEMA predictor, the results shown are for the best value of α_{max} that has been found for each flow, since its effectiveness

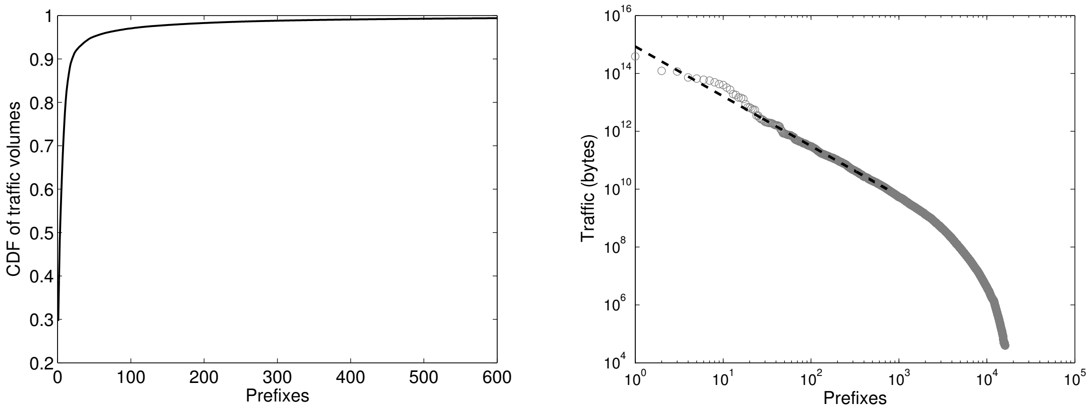


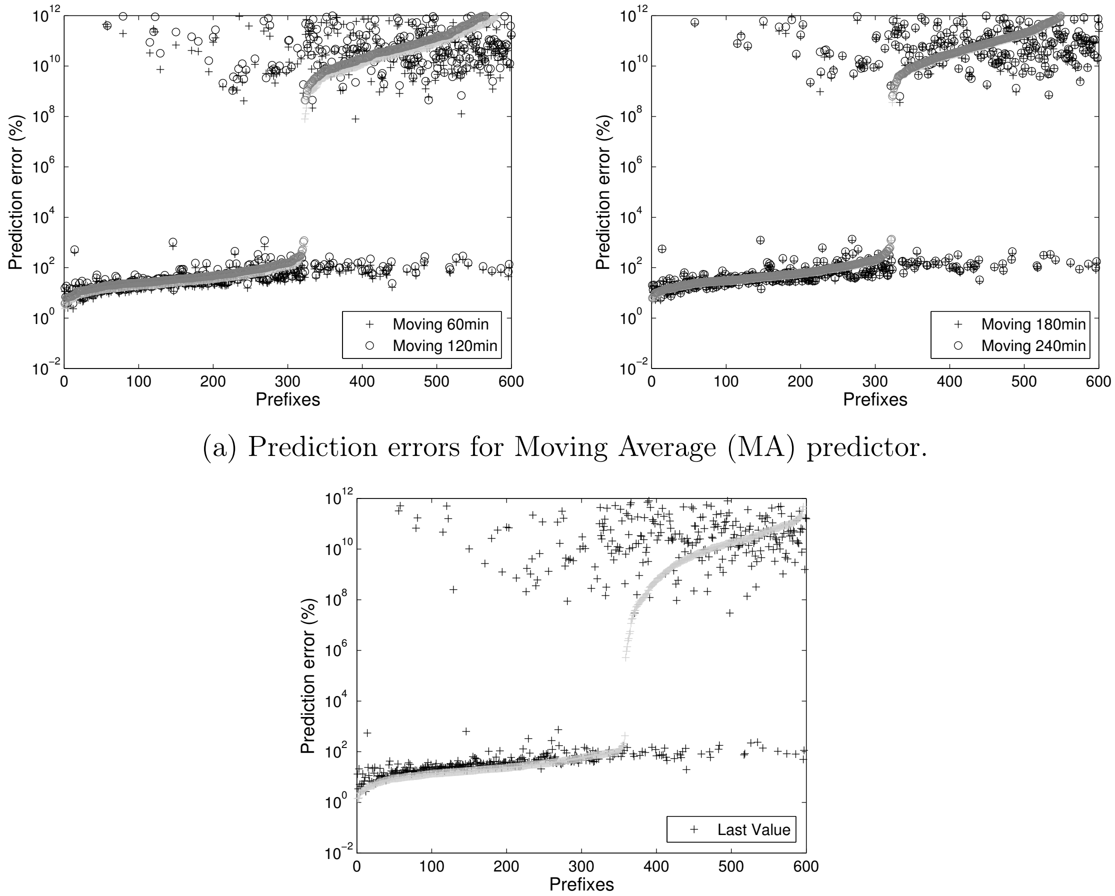
Figure A.2: GÉANT Traffic: (left) Cumulative Distribution Function of traffic volumes; (right) Zipf's law fitting giving $\alpha = 1.5818$ and $c = 9.6634E^{10}$.

depends on this parameter (as Figure A.4 suggests). The results of Figures A.3 and A.4 also confirm the previous finding, even when a different trace is used, that the performance of the different predictors does not differ very much in terms of complexity. Nevertheless, these results yield an extra finding, which is, that the errors of all the predictors depends on the granularity of the traffic flows. In fact, in both Figures A.3 and A.4, the prediction errors grow sharply, (roughly) above of the flow for the prefix number 300. Hence, *this suggests that the predictors are unable to track very small flows accurately.*

Previous finding introduces a practical bound that should be taken into account in manual or adaptive setting of the threshold T. This finding states that, regardless of other factors, it only makes sense to track "popular destinations" if its predictability is effective, i.e., those for which the mean prediction error is bounded. To be more specific, using this error-based prediction criterion with an approximate target maximum error of 100% (i.e., $E(f_i) = \frac{\sum_o^n error_k}{n} \leq 1$, where $E(f_i)$ is the mean prediction error for flow f_i, and $error_k$ is the prediction error in slot of time k), 296 destinations were identified (out of a total of 16150 prefixes), and the sum of their individual traffic represents 99% *of the total volume.*

Regarding the tuning of LpEMA, it should be noted that the prediction errors for a range of values for α_{max}, flows of different granularity and aggregation, and sizes of the sampling window were examined. Figure A.4 shows that the prediction error depends on all these factors. First, it shows that as long as α_{max} increases the prediction error decays significantly, and thus the predictor adapts to the variability of the traffic. When it is above some value for α_{max} the prediction error rises. Second, when the sum of all the traffic, the largest traffic flow, and the smallest traffic flow are compared, it can be observed that the aggregation favors the predictability of the traffic. This is evident for all the predictors.

Moreover, the best α_{max} that have been found to track each of the traffic flows is



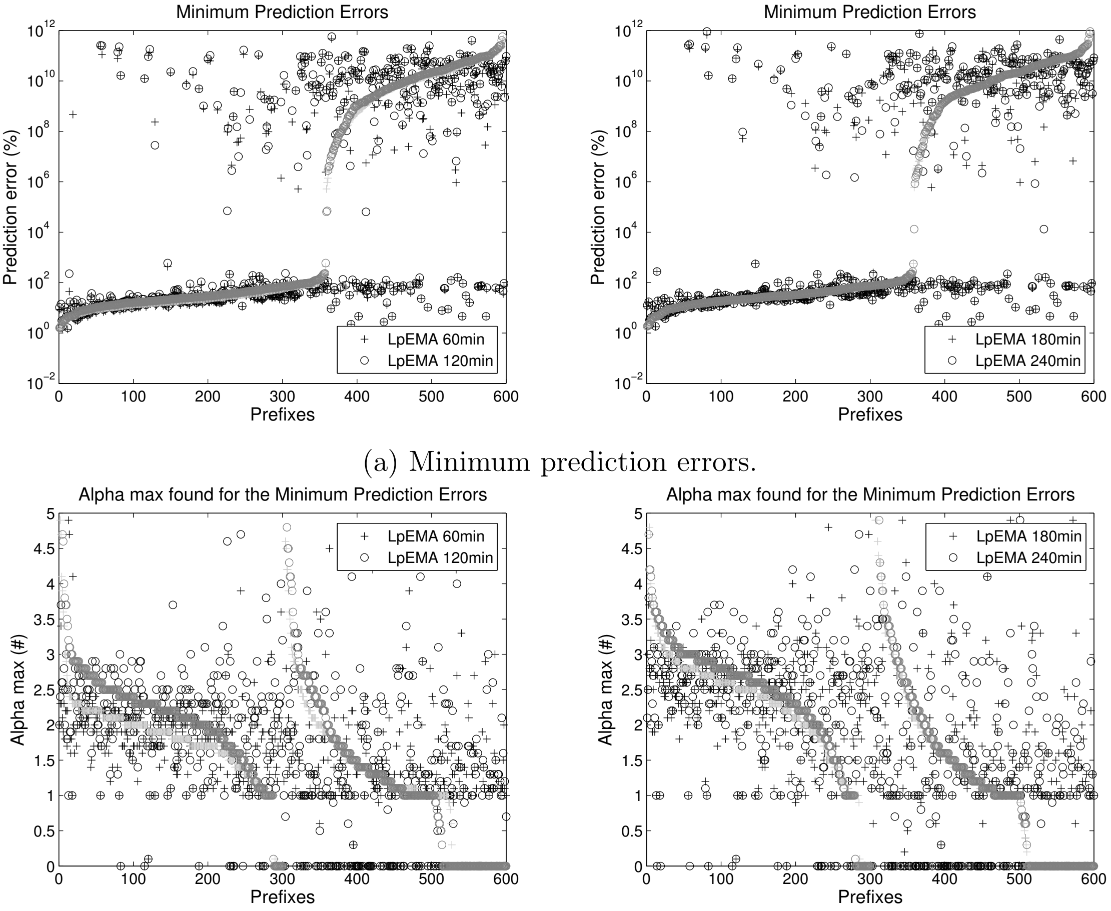
(a) Prediction errors for Moving Average (MA) predictor.

(b) Prediction errors for Last Value (LV) predictor.

Figure A.3: Analysis of the prediction errors for Moving Average (MA) and Last Value (LV) predictors.

highly variable, which means that a significant effort is required for tuning the LpEMA because there is no common value for all the flows. However, we found that it is more common to use values within the range [1.5, 3.0] or [2.0, 3.5] for bigger traffic flows, in the case of smaller and larger sizes of time intervals respectively. A possible method that can be used to define the α_{max} statistically, is to compute the median value.

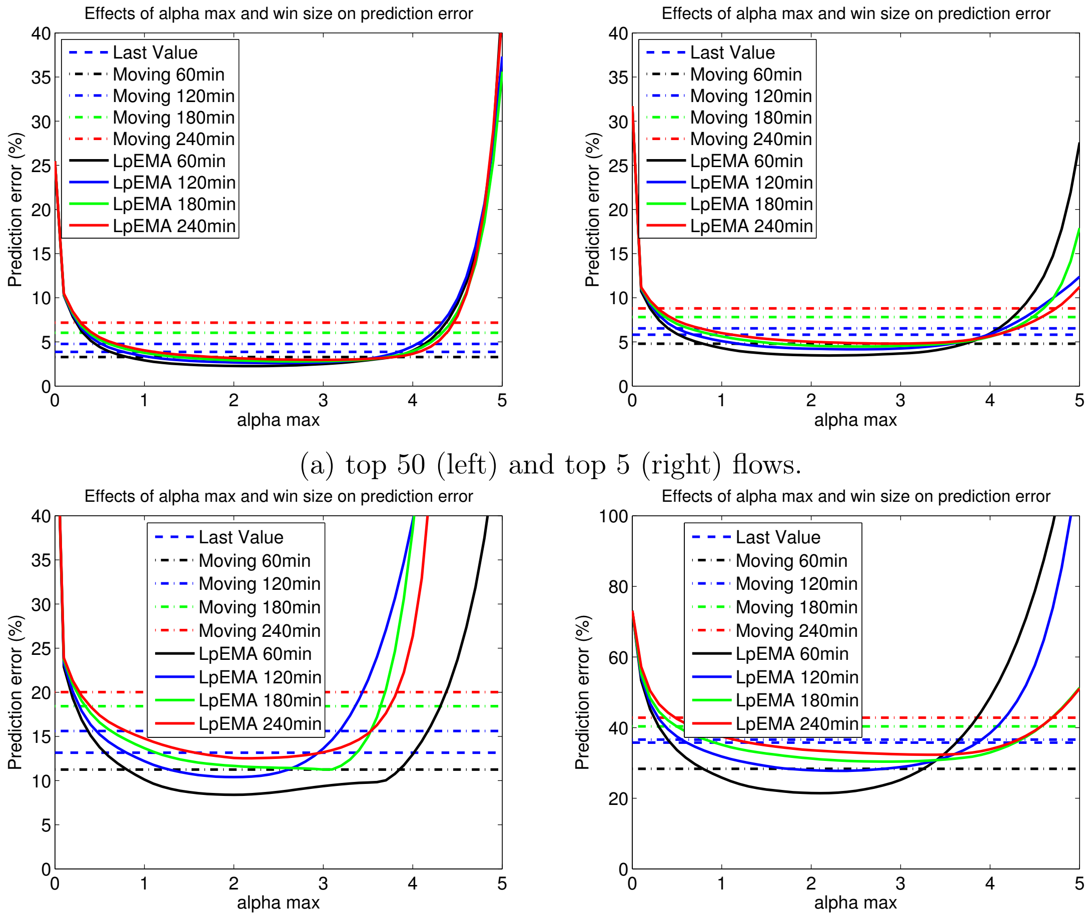
One final remark is that the choice of predictors depends on a trade-off between accuracy and complexity. In [123], the authors, together with their references, argue that the use of complex predictors, such as LpEMA, may not bring performance benefits because their “mean error is always larger than the one of the last value”. Figures A.5 and A.6 provide a counter-example. The results show that when making proper choices of α_{max}, LpEMA performs always better than LV, and no worse than MA, with regard to mean error. Moreover, the average gradient for LpEMA is always lower than that of LV. In short, the complex predictors may not lead to any benefits in terms of mean error, *but* despite this, may have a potential advantage in being able to ensure routing stability, while also following the traffic dynamics accurately.



(a) Minimum prediction errors.

(b) Best alpha max values found.

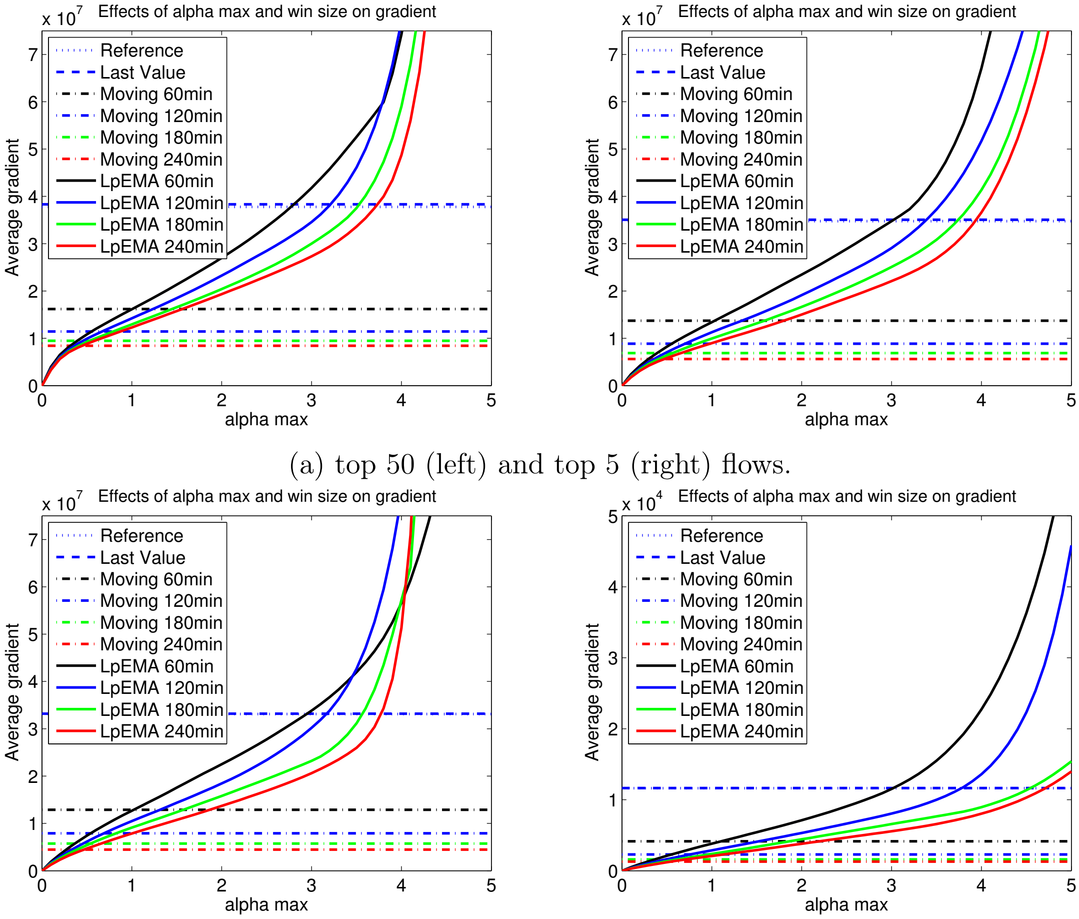
Figure A.4: Analysis of the prediction errors for the LpEMA predictor.



(a) top 50 (left) and top 5 (right) flows.

(b) biggest (left) and smallest (right) flows.

Figure A.5: Comparison of the prediction errors for different flow sizes and aggregation levels.



(a) top 50 (left) and top 5 (right) flows.

(b) biggest (left) and smallest (right) flows.

Figure A.6: Comparison of the average gradients of the samples for different flow sizes and aggregation levels.

A.3 Conclusion

This work has highlighted the importance of the consistency of traffic demands with the Zipf's law for the whole inter-domain traffic engineering process integrated in network management systems. This implies that it is enough to take account of only a small fraction of the total number of destinations (a.k.a. popular destinations) to control the routing of the majority of the traffic. However, there has been a lack of any simple and pragmatic method for selecting popular destinations.

In view of this, we drew up a practical criterion for the selection of popular destinations. The proposed criterion is based on the definition of a target bound for traffic volumes an relies on information about the behavior of the errors of the traffic predictors. The results showed that by applying this criterion, we were able to reduce the number of target prefixes to 2% of the total number, while ensuring routing stability due to the predictability of the traffic headed these prefixes.

Appendix B

Evaluation of IRC under Different Path Switching Policies

Dynamic path switching is the key technique used by the Intelligent Route Control (IRC) to obtain better end-to-end performance [87–89]. As described in Chapter 3, there are three path switching policies that can be observed by the IRCs, which are as follows:

- **Choose Best-Choose Best policy (CBCB)** – According to the CBCB policy (also denoted as CB policy), the IRC switches paths whenever it finds one that is better in terms of QoS. Thus, for each destination, the IRC picks the path that has the smallest value of the chosen metric, regardless of any QoS bound;
- **Choose Best-Choose Good policy (CBCG)** – In a similar way to the CBCB policy, the IRC switches paths whenever it finds better paths in terms of QoS. However, there is an important difference; here, the IRC is aware of the end-to-end quality bounds, so that if the quality of a path does not fit these bounds, it is enough to pick any alternative good path;
- **Choose Good policy (CG)** – According to the CG policy, the IRC only switches paths if their QoS characteristics are not sufficient to handle the QoS traffic requirements. If this is the case, it picks any suitable alternative path.

The purpose of this Appendix is to show the results of the original study that was carried out to compare the IRC strategy with BGP (Border Gateway Protocol), including a performance assessment of the IRC strategy, under different path switching policies.

B.1 Simulation Study

In this study, IRC was evaluated by means of the following path switching policies – Choose-Best-Choose-Best (CBCB), Choose-Best-Choose-Good (CBCG) and Choose-

Good (CG), when Differentiated Services (Diffserv) [6] was switched ON or OFF, and contrasted with the pure BGP model, resulting, thus, in four pairs of simulation scenarios:

1. CBCB IRCs and Diffserv enabled (CBCB-DSen) / CBCB IRCs and Diffserv disabled (CBCB-DSun);
2. CBCG IRCs and Diffserv enabled (CBCG-DSen / CBCG IRCs and Diffserv disabled (CBCG-DSun);
3. CG IRCs and Diffserv enabled (CG-DSen) / CG IRCs and Diffserv disabled (CG-DSun);
4. BGP and Diffserv enabled (BGP-DSen) / BGP and Diffserv disabled (BGP-DSun).

The simulations were performed with the aid of the J-Sim [145] simulator in which the functionalities of the IRCs were implemented.

Figure B.1 provides a simplified illustration of the network model that was used in the simulations. The simulated network aims at representing a multi-service part of the Internet composed of access Internet Service Providers (ISPs) that are able to provide some limited QoS services to some of their customers, and an over-provisioned Internet core.

The tests were conducted by means of a traffic mix consisting of Voice over IP (VoIP) calls, video calls, prioritized data, and Web traffic. When Diffserv is switched ON packets are classified and forwarded using the standard Expedited Forwarding (`EF`), Assured Forwarding (`AF11` and `AF21`), and Best-Effort (`BE`) traffic classes, respectively [148, 149].

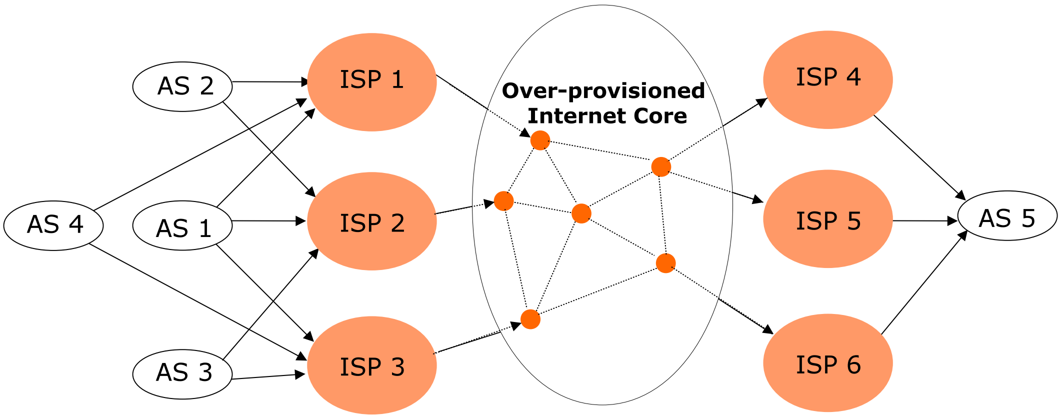


Figure B.1: Network model.

During all the experiments, it was assumed that a Service-Level Specification (SLS) had previously been exchanged between the remote multihomed stub networks, based on maximum One-Way Delay (OWDs) for each service. The maximum OWDs that can be tolerated by each service were chosen to represent reasonable but demanding values for the kinds of traffic sources that were employed during the simulations. Thus, the OWD bound for voice and video traffic was set at 100ms, and at 400ms for prioritized data and web traffic, which is in accordance with the E-model Rating from ITU's G.107/G.114 recommendations [108, 109].

B.1.1 Performance Metrics and Objectives

The first objective is to assess how far IRCs can improve end-to-end network quality. This is shown by evaluating:

(i) the average end-to-end OWD (or latency), denoted as owd;

(ii) and the traffic transfer efficiency for the different traffic flows, denoted as Ef. The efficiency for a traffic class is defined by $Ef = \frac{F_D}{C_O}$, where F_D is the total throughput at destinations D, and C_O is the corresponding total throughput sent by source domains O.

The second objective is to study how the IRCs can contribute to the overall network stability under variable performance/QoS dynamics. The performance indicator that is employed is the total number of path switches that is needed to meet the latency constraint for each kind of traffic.

Finally, third objective is to determine the overall efficiency of BGP and IRCs under the different path switching policies. The overall efficiency is given by an efficiency index, which represents the relative number of path switches that were needed to obtain a given latency and traffic transfers efficiency, as shown in Equation (B.1.1). A Lower value for this index indicate better overall efficiency, i.e., that there is a better trade-off between the number of path switches, the averaged OWD and the traffic transfer efficiency; this means that a given IRC path switching policy is more efficient.

$$\text{Efficiency index} = 100 * PSindex\left(\frac{d}{d_{max}}\right)\left(\frac{100}{Ef}\right) \tag{B.1.1}$$

, where $PSindex$ is a path switch index computed, as follows.

$$PSindex = \begin{cases} \frac{PS}{PS_{ref}} & \text{, if } PS \geq PS_{ref} \\ \frac{abs\left(PS-PS_{ref}\right)+PS_{ref}}{PS_{ref}} & \text{, if } PS < PS_{ref} \end{cases} \tag{B.1.2}$$

, where PS_{ref} is the number of path switches performed by the IRC adopting a given reference policy.

In the evaluations, the CG policy was used as the reference policy for calculating of the overall efficiency. Thus, the $PSindex$ is given by the number of path switches performed by other policies, when normalized by the number of path switches performed by the IRC CG policy. Otherwise, the $PSindex$ is given by a penalty that depends on the number of path switches that would be carried out with the IRC policy to achieve the same performance with the IRC CG policy.

B.1.2 Results

The first results concern the latency obtained from the different IRC path switching policies and BGP. The cluster of Figures B.2 illustrates the Complementary Cumulative Distribution Functions (CCDF) of the traffic latencies. If the probability of the OWD that is greater than or equal to x, is high (i.e., $P(OWD \geq x)$ is high), it means that there is a high likelihood that the traffic will suffer a latency greater than or equal to x. In turn, Figure B.3.(a) shows the averaged latencies for each traffic.

Two main conclusions can be drawn from the analysis of the latency results. First, the IRC architecture substantially enhances end-to-end QoS when compared with a pure BGP model. On the one hand, when Diffserv is not active and the IRCs are switched OFF, under stressful traffic load, BGP is not capable of supporting the ITU performance bounds. In particular, Figure B.2 shows that with the BGP-based scheme, there is a likelihood of violation of all the OWD bounds greater than 95%. On the other hand, when the IRCs are switched ON, Figure B.2 shows that with all the IRC policies, there is a greater than 50% likelihood of violation of the maximum bounds for all the traffic classes. This is not necessarily unsatisfactory given the fact that the simulations were carried out under stressful traffic load, where the number of feasible paths available to carry traffic is more limited. Another observation is that the CG policy has longer tails, so that for less important traffic classes (i.e., data prioritized and Web traffic), there is a greater than 10% probability that the latency exceeds 500 ms. This is what is expected, since the CG policy only reacts to QoS violations.

Second, Diffserv clearly shows its effectiveness in protecting the most important traffic classes, since all the traffic classes have OWDs within the ranges permitted, except for web traffic, and sometimes, for the interactive data prioritized traffic when BGP is enabled. However surprising this may seem, this is only possible on account of the strong efficiency penalization of the prioritized data traffic and web traffic. Figure B.3.(b) clearly shows that, especially in the case of BGP, more than 60% of the web traffic is dropped by the packet scheduling mechanisms of Diffserv.

Following this, there is a discussion of the IRC path switching policies used for controlling the number of path switches. Figure B.3.(c) shows the total number of path switches registered for each policy. As can be expected, the IRC path switching algorithm based on the CBCB policy shows the largest number of path switches, since according to the CBCB policy, an IRC switches paths whenever a better path is found. This is especially evident when Diffserv is disabled, and particularly for the less important traffic classes. Moreover, the IRC path switching algorithm that is based on the CG policy, shows the best performance in terms of the number of path switches. However, this is not necessarily a positive feature given the fact that this policy only reacts when there is a clear violation of the performance bound. As mentioned earlier, longer tails in the cluster of Figures B.3.(a) clearly show that this policy might allow unacceptable latencies.

Finally, Figure B.3.(d) displays the overall performance index results. As would be expected, BGP shows the weakest overall efficiency. On the other hand, a detailed observation leads us to conclude that the CBCG policy appears to be the most balanced scheme in terms of overall efficiency. This means that any additional path switch performed by the IRC using a CG policy, rather than other policies, results in traffic improvements in terms of latency and efficiency. The results also show that a larger number of path switches might not always be offset by improved latency and efficiency.

B.2 Summary

In this Appendix, we assessed the performance of IRCs under different path switching policies. The simulation study carried out showed that the CBCG policy is the most effective IRC path switching policy. In other words, this policy is the one that offers the best trade-off between the number of path switches, the averaged OWD and the traffic transfers efficiency.

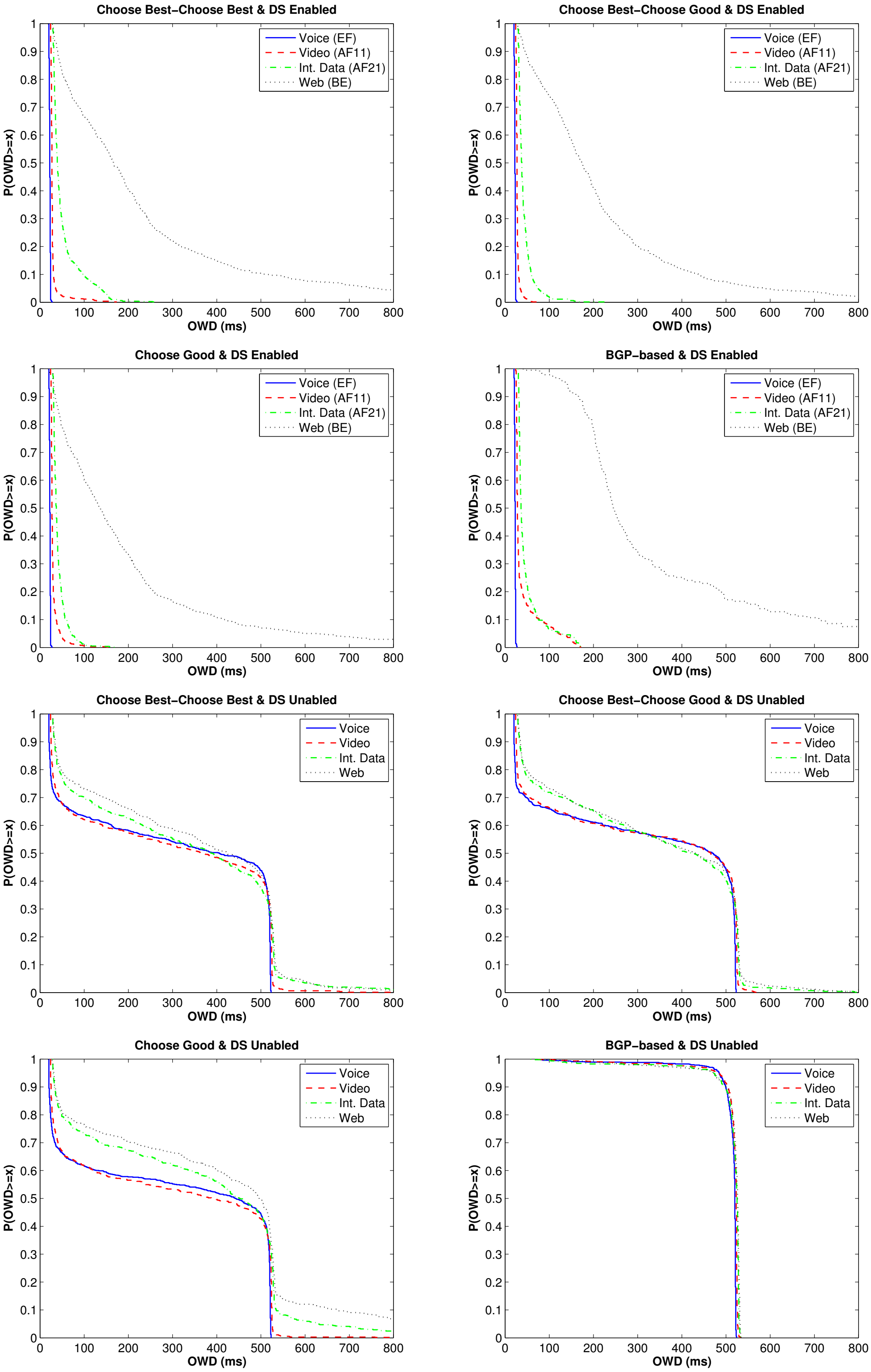
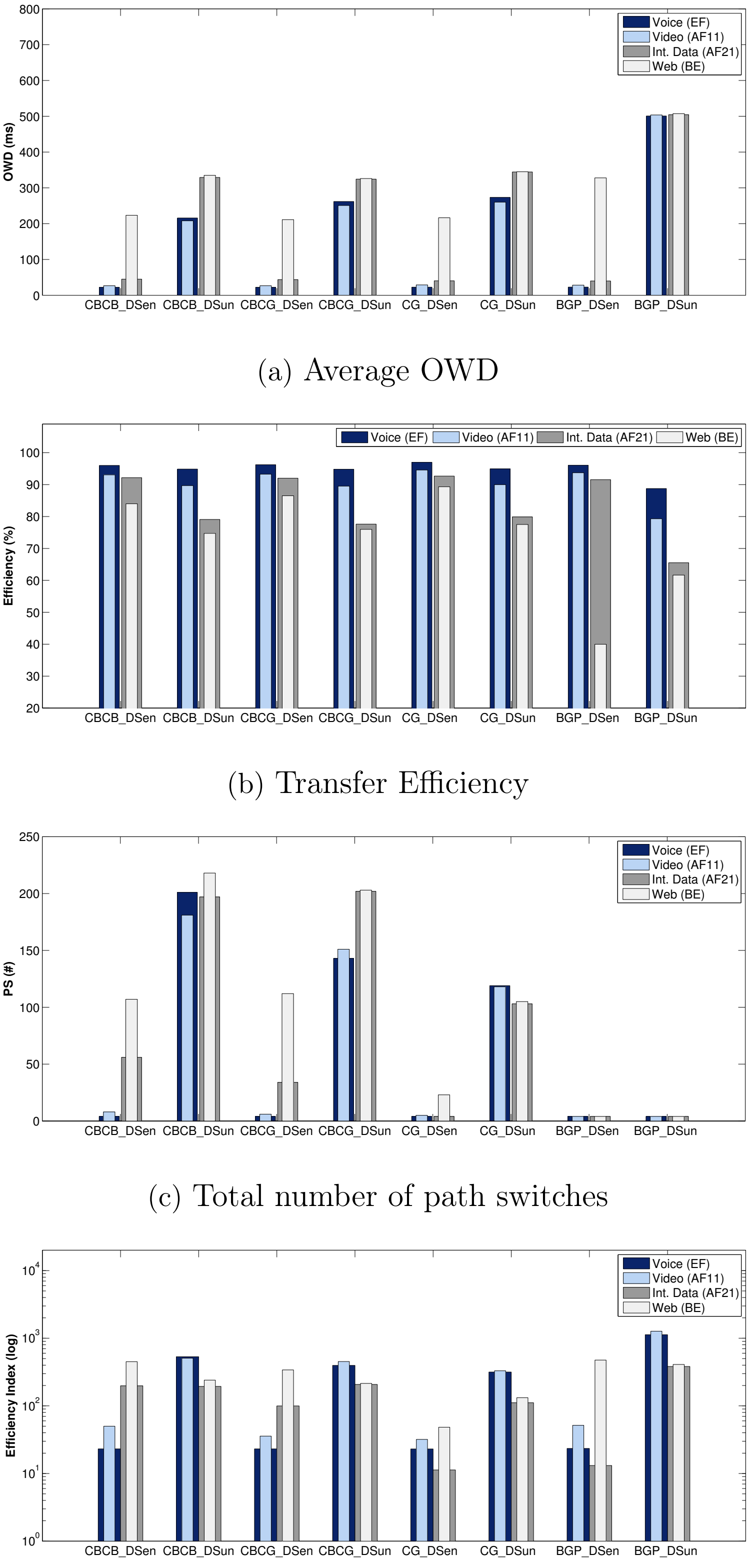


Figure B.2: Complementary Cumulative Distribution Function (CCDF) of OWDs for the competing IRC flows, whether (top) Diffserv feature is enabled or (bottom) Diffserv feature is disabled.



(a) Average OWD

(b) Transfer Efficiency

(c) Total number of path switches

(d) Global Efficiency Index

Figure B.3: Performance of IRC against BGP, under different path switching policies.

Bibliography

[1] *CIDR Report*, accessed September 2011. [Online]. Available: http://www.cidr-report.org

[2] J. Postel, "Internet Protocol," RFC 791 (Standard), Internet Engineering Task Force, September 1981, updated by RFC 1349. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[3] P. Almquist, "Type of Service in the Internet Protocol Suite," RFC 1349 (Proposed Standard), Internet Engineering Task Force, July 1992, obsoleted by RFC 2474. [Online]. Available: http://www.ietf.org/rfc/rfc1349.txt

[4] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460 (Draft Standard), Internet Engineering Task Force, December 1998, updated by RFCs 5095, 5722, 5871. [Online]. Available: http://www.ietf.org/rfc/rfc2460.txt

[5] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633 (Informational), Internet Engineering Task Force, June 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1633.txt

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Service," RFC 2475 (Informational), Internet Engineering Task Force, December 1998, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[7] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational IP backbone network," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 749–762, August 2008. [Online]. Available: http://dx.doi.org/10.1109/TNET.2007.902727

[8] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot, "Characterization of Failures in an IP backbone," in *Proceedings of the INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, Hong Kong, China, March 2004, pp. 2307–2317.

[9] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," in *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '03)*. San Diego, CA, USA: ACM, 2003, pp. 316–317.

[10] *IETF QoS Routing Working Group (WG)*, IETF, accessed December 2010. [Online]. Available: http://datatracker.ietf.org/wg/qosr/charter/

[11] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-based Routing in the Internet," RFC 2386 (Informational), Internet Engineering Task Force, August 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2386.txt

[12] M. Curado and E. Monteiro, "Quality of service routing," *Encyclopedia of Internet Technologies and Applications, M. Freire, M. Pereira (Eds.), Information Science Reference, Idea Group*, 2006.

[13] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," in *Proceedings of the IEEE INFOCOM 2001, The Conference on Computer Communications, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Anchorage, Alaska, USA, April 2001, pp. 834–843.

[14] H. Neve and P. Mieghem, "TAMCRA: a tunable accuracy multiple constraints routing algorithm," *Computer Communications*, vol. 23, no. 7, pp. 667–679, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0140-3664(99)00225-X

[15] F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Communications Magazine*, vol. 40, no. 12, pp. 50–55, December 2002.

[16] Q. Ma and P. Steenkiste, "Routing Traffic with Quality-of-Service Guarantees in Integrated Services Networks," in *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, New Hall College, Cambridge UK, July 1998.

[17] ——, "On path selection for traffic with bandwidth guarantees," in *Proceedings of the 1997 International Conference on Network Protocols (ICNP '97)*, Atlanta, Georgia, October 1997, pp. 191–202.

[18] G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, and T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions," RFC 2676 (Experimental), Internet Engineering Task Force, August 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2676.txt

[19] L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 541–550, 2002.

[20] S. Reeves and F. Salama, "A distributed algorithm for delay-constrained unicast routing," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 239–250, 2000.

[21] G. Cheng and N. Ansari, "A new heuristics for finding the delay constrained least cost path," in *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM '03)*, vol. 7, San Francisco, USA, December 2003, pp. 3711–3715.

[22] A. Juttner, B. Szviatovski, I. Mecs, and Z. Rajko, "Lagrange relaxation based method for the QoS routing problem," in *Proceedings of the IEEE INFOCOM 2001, The Conference on Computer Communications, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Anchorage, Alaska, USA, 2001, pp. 859–868.

[23] Y. Cui, K. Xu, and J. Wu, "Precomputation for Multi-constrained QoS routing in High-speed Networks," in *Proceedings of the IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, San Franciso, CA, USA, March-3 April 2003.

[24] P. V. Mieghem, O. A. Kuipers, and S. Member, "Concepts of Exact Qos Routing Algorithms," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 851–864, 2004.

[25] P. Van Mieghem, H. De Neve, and F. Kuipers, "Hop-by-hop quality of service routing," *Computer Networks*, vol. 37, no. 3-4, pp. 407–423, 2001.

[26] T. Korkmaz, M. Krunz, and S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints," in *Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '00).* Santa Clara, California, United States: ACM, 2000, pp. 318–327.

[27] L. Ford, "Network flow theory," The RAND Corporation, Santa Moncia, California, Paper P-923, August 1956. [Online]. Available: http://www.rand.org/cgi-bin/Abstracts/ordi/getabbydoc.pl?doc=P-923

[28] R. Bellman, "On a Routing Problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958. [Online]. Available: http://wisl.ece.cornell.edu/ECE794/Jan29/bellman1958.pdf

[29] E. W. Dijkstra, “A Note on Two Problems in Connection with Graphs,” *Numerical Mathematics*, vol. 1, pp. 269–271, 1959, http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf.

[30] J. Moy, “OSPF Version 2,” RFC 2328 (Standard), Internet Engineering Task Force, April 1998, updated by RFC 5709. [Online]. Available: http://www.ietf.org/rfc/rfc2328.txt

[31] G. Malkin, “RIP Version 2,” RFC 2453 (Standard), Internet Engineering Task Force, November 1998, updated by RFC 4822. [Online]. Available: http://www.ietf.org/rfc/rfc2453.txt

[32] Y. Rekhter, T. Li, and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” RFC 4271 (Draft Standard), Internet Engineering Task Force, January 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4271.txt

[33] X. Masip-Bruin, M. Yannuzzi, J. Domingo-Pascual, A. Fonte, M. Curado, E. Monteiro, F. Kuipers, P. V. Mieghem, S. Avallone, G. Ventre, P. Aranda-Gutierrez, M. Hollick, R. Steinmetz, L. Iannone, and K. Salamatian, “Research challenges in QoS routing,” *Computer Communications*, vol. 29, no. 5, pp. 563 – 581, 2006.

[34] X. Masip-Bruin, S. Sánchez-López, J. Solé-Pareta, and J. Domingo-Pascual, “A QoS Routing Mechanism for Reducing the Routing Inaccuracy Effects,” in *Proceedings of the Second International Workshop on Quality of Service in Multiservice IP Networks*, ser. QoS-IP 2003. London, UK: Springer-Verlag, 2003, pp. 90–102.

[35] M. Claypool, M. Claypool, G. Kannan, and G. Kannan, “Selective flooding for improved quality-of-service routing,” in *Proceedings of SPIE Quality of Service over Next-Generation Data Networks*, Denver, Colorado, USA, August 2001.

[36] S. Uludag, K.-S. Lui, K. Nahrstedt, and G. Brewster, “Analysis of Topology Aggregation techniques for QoS routing,” *ACM Computing Surveys (CSUR)*, vol. 39, September 2007. [Online]. Available: http://doi.acm.org/10.1145/1267070.1267071

[37] M. Pitkanen and M. Luoma, “OSPF flooding process optimization,” in *Proceedings of the 2005 Workshop on High Performance Switching and Routing (HPSR 2005)*, Hong Kong, China, May 2005, pp. 448 – 52.

[38] G. Apostolopoulos, R. Guerin, and S. Kamat, “Implementation and performance measurements of QoS routing extensions to OSPF,” in *Proceedings of the IEEE*

INFOCOM '99, The Conference on Computer Communications, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, The Future Is Now, vol. 2, New York, USA, March 1999, pp. 680–688.

[39] P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, D. Griffin, P. Georgatsos, D. Goderis, Y. T'Joens, L. Georgiadis, C. Jacquenet, and R. Egan, "A management and control architecture for providing IP differentiated services in MPLS-based networks," *IEEE Communications Magazine*, vol. 39, no. 5, pp. 80–88, May 2001.

[40] D. Goderis *et al.*, "Service Level Specification Semantics and Parameters," in *draft-tequila-sls-02.txt, IETF draft*, Internet Engineering Task Force. United States: IETF, August 2002.

[41] H. Pouyllau and R. Douville, "End-to-end QoS negotiation in network federations," in *Proceedings of Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, April 2010, pp. 173 –176.

[42] M. Howarth, P. Flegkas, G. Pavlou, N. Wang, P. Trimintzios, D. Griffin, J. Griem, M. Boucadair, P. Morand, A. Asgari, and P. Georgatsos, "Provisioning for inter-domain quality of service: the MESCAL approach," *Communications Magazine, IEEE*, vol. 43, no. 6, pp. 129–137, June 2005.

[43] X. Masip-Bruin, M. Yannuzzi, R. Serral-Gracia, J. Domingo-Pascual, J. Enriquez-Gabeiras, M. A. Callejo, M. Diaz, F. Racaru, G. Stea, E. Mingozzi, A. Beben, W. Burakowski, E. Monteiro, and L. Cordeiro, "The EuQoS system: a solution for QoS routing in heterogeneous networks [quality of service based routing algorithms for heterogeneous networks]," *Communications Magazine, IEEE*, vol. 45, no. 2, pp. 96–103, February 2007.

[44] "Specification of Business Models and a Functional Architecture for Inter-domain QoS Delivery," in *Deliverable D1.1 of the IST MESCAL project Interdomain QoS for the Internet, (www.mescal.org)*, 2003.

[45] R. Chandra and J. Scudder, "Capabilities Advertisement with BGP-4," RFC 3392 (Draft Standard), Internet Engineering Task Force, November 2002, obsoleted by RFC 5492. [Online]. Available: http://www.ietf.org/rfc/rfc3392.txt

[46] O. Bonaventure, "Using BGP to distribute flexible QoS information," in *draft-bonaventure-bgp-qos-00, IETF draft*, Internet Engineering Task Force. United States: IETF, February 2001.

[47] G. C. Christian and C. Jacquenet, "An approach to inter-domain Traffic Engineering," in *Proceedings of XVIII World Telecommunications Congress (WTC2002)*, Paris, France, September 2002, pp. 563–581.

[48] G. Cristallo and C. Jacquenet, "Providing Quality of Service Indication by BGP-4 protocol: the QOS-NLRI attribute," in *draft-jacquenet-qos-nlri-05.txt, IETF draft*, Internet Engineering Task Force. United States: IETF, June 2003.

[49] A. Fonte, M. Curado, and E. Monteiro, "Interdomain quality of service routing: setting the grounds for the way ahead," *Annals of Telecommunications*, vol. 63, pp. 683–695, 2008, 10.1007/s12243-008-0065-y. [Online]. Available: http://dx.doi.org/10.1007/s12243-008-0065-y

[50] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The impact of Internet policy and topology on delayed routing convergence," in *Proceedings of the IEEE INFOCOM 2001, The Conference on Computer Communications, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, Anchorage, Alaska, USA, April 2001, pp. 537–546 vol.1.

[51] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 293–306, 2001.

[52] N. Feamster and J. Rexford, "Network-wide prediction of BGP routes," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 253–266, 2007.

[53] K. hon Ho, N. Wang, P. Trimintzios, and G. Pavlou, "Multi-objective egress router selection policies for inter-domain traffic with bandwidth guarantees," in *Proceedings of the Third International IFIP-TC6 Networking Conference (NETWORKING 2004*, Greece, May 2004, pp. 271–283.

[54] F. Guo, J. Chen, W. Li, and T. cker Chiueh, "Experiences in building a multihoming load balancing system," in *Proceedings of the IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Hong Kong, China, March 2004, pp. 1241–1251.

[55] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute," RFC 1997 (Proposed Standard), Internet Engineering Task Force, August 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1997.txt

[56] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4," RFC 4760 (Draft Standard), Internet Engineering Task Force, January 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4760.txt

[57] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," RFC 4632 (Best Current Practice), Internet Engineering Task Force, August 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4632.txt

[58] Q. Vohra and E. Chen, “BGP Support for Four-octet AS Number Space,” RFC 4893 (Proposed Standard), Internet Engineering Task Force, May 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4893.txt

[59] J. Postel, “Internet Control Message Protocol,” RFC 792 (Standard), Internet Engineering Task Force, September 1981, updated by RFCs 950, 4884. [Online]. Available: http://www.ietf.org/rfc/rfc792.txt

[60] *Cisco IOS IP Configuration Guide, Release 12.2*, Cisco Systems, accessed December 2010. [Online]. Available: http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbook.pdf

[61] S. Deshpande and B. Sikdar, “On the impact of route processing and MRAI timers on BGP convergence times,” in *Proceedings of the 47th annual IEEE Global Telecommunications Conference, 2004 (IEEE GLOBECOM ’04)*, vol. 2, November-3 December 2004, pp. 1147–1151.

[62] *JUNOSe Internet Software for E-series Routing Platforms, BGP and MPLS Configuration Guide*, Juniper, accessed December 2010. [Online]. Available: http://www.juniper.net/techpubs/software/erx/junose73/swconfig-bgp-mpls/html/title-swconfig-bgp-mpls.html

[63] M. Caesar and J. Rexford, “BGP routing policies in ISP networks,” *IEEE Network*, vol. 19, no. 6, pp. 5–11, November-December 2005.

[64] L. Gao, “On inferring autonomous system relationships in the Internet,” *IEEE/ACM Transactions on Networking*, vol. 9, pp. 733–745, December 2001. [Online]. Available: http://dx.doi.org/10.1109/90.974527

[65] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz, “OverQoS: offering Internet qoS using overlays,” *Computer Communication Review*, vol. 33, no. 1, pp. 11–16, 2003. [Online]. Available: http://doi.acm.org/10.1145/774763.774764

[66] G. D. Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, and T. Schank, “Computing the Types of the Relationships Between Autonomous Systems,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 267 –280, April 2007.

[67] N. Feamster, J. Borkenhagen, and J. Rexford, “Guidelines for interdomain traffic engineering,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 19–30, October 2003. [Online]. Available: http://doi.acm.org/10.1145/963985.963988

[68] R. Chang and M. Lo, “Inbound traffic engineering for multihomed ASs using AS path prepending,” *IEEE Network*, vol. 19, no. 2, pp. 18–25, March-April 2005.

[69] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, vol. 2, 2000, pp. 519–528.

[70] B. Quoitin and O. Bonaventure, "A cooperative approach to interdomain traffic engineering," in *Proceedings of 1st Conference on Next Generation Internet Networks (NGI'05)*, April 2005.

[71] B. Huffaker, M. Fomenkov, D. J. Plummer, D. Moore, k claffy, B. H. M. Fomenkov, and A. Background, "Distance Metrics in the Internet," in *Proceedings of IEEE International Telecommunications Symposium (ITS)*, 2002, pp. 200–202.

[72] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '02)*. Pittsburgh, Pennsylvania, USA: ACM, 2002, pp. 3–16.

[73] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "In search of path diversity in isp networks," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, ser. IMC '03. New York, NY, USA: ACM, 2003, pp. 313–318. [Online]. Available: http://doi.acm.org/10.1145/948205.948247

[74] C. de Launois, B. Quoitin, and O. Bonaventure, "Leveraging network performance with IPv6 multihoming and multiple provider-dependent aggregatable prefixes," *Computer Networks*, vol. 50, pp. 1145–1157, June 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1143164.1143173

[75] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks*, vol. 32, no. 1, pp. 1–16, 2000. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(99)00108-5

[76] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 232–243, 2002.

[77] K. Donnelly, A. Kfoury, and A. Lapets, "The complexity of restricted variants of the stable paths problem," *Fundam. Inf.*, vol. 103, pp. 69–87, January 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1922521.1922526

[78] L. Xiao, J. Wang, K.-S. Lui, and K. Nahrstedt, "Advertising interdomain QoS routing information," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 10, pp. 1949–1964, December 2004.

[79] L. Xiao, K.-S. Lui, J. Wang, and K. Nahrstedt, "QoS extension to BGP," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP2002)*, Paris, France, November 2002, pp. 100–109.

[80] M. Boucadair, "QoS-Enhanced Border Gateway Protocol," in *draft-boucadair-qos-bgp-spec-01.txt, IETF draft*, Internet Engineering Task Force. United States: IETF, July 2005.

[81] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, January 2004.

[82] S. Agarwal, C.-N. Chuah, and R. Katz, "OPCA: robust interdomain policy routing and traffic control," in *Proceedings of the 2003 IEEE Conference on Open Architectures and Network Programming (IEEE Openarch 2003)*, San Francisco, California, USA, April 2003, pp. 55–64.

[83] Y. Liu, H. Zhang, W. Gong, and D. Towsley, "On the interaction between overlay routing and underlay routing," in *Proceedings of the IEEE INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, Miami, FL, USA, March 2005, pp. 2543–2553.

[84] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of the eighteenth ACM symposium on Operating systems principles (SOSP '01).* Banff, Alberta, Canada: ACM, October 2001, pp. 131–145.

[85] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03).* Karlsruhe, Germany: ACM, 2003, pp. 353–364.

[86] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics," RFC 2330 (Informational), Internet Engineering Task Force, May 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2330.txt

[87] *Cisco Systems, Inc., Performance Routing (PfR)*, accessed June 2011. [Online]. Available: http://www.cisco.com/en/US/products

[88] *Internap Networks Inc., Flow Control Platform*, accessed June 2011. [Online]. Available: http://www.internap.com/business-internet-connectivity-services/

[89] R. Gao, C. Dovrolis, and E. W. Zegura, "Avoiding Oscillations Due to Intelligent Route Control Systems," in *Proceedings of the IEEE INFOCOM 2006, 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies*, Barcelona, Catalunya, April 2006, pp. 1–12.

[90] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in internet-like environments," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 725–738, 2006.

[91] G. Shrimali, A. Akella, and A. Mutapcic, "Cooperative Interdomain Traffic Engineering Using Nash Bargaining and Decomposition," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 341 –352, 2010.

[92] ——, "Cooperative interdomain traffic engineering using nash bargaining and decomposition," in *Proceedings of IEEE INFOCOM 2007, 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, Alaska, USA, May 2007.

[93] D. Meyer, "Operational Concerns and Considerations for Routing Protocol Design – Risk, Interference, and Fit (RIFT)," in *draft-ietf-grow-rift-01.txt, IETF draft*, Internet Engineering Task Force. United States: IETF, 2004.

[94] D. Meyer and K. Patel, "BGP-4 Protocol Analysis," RFC 4274 (Informational), Internet Engineering Task Force, January 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4274.txt

[95] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "Impact of BGP dynamics on router CPU utilization," in *Proceedings of Passive and Active Network Measurement, 5th International Workshop (PAM 2004)*, ser. Lecture Notes in Computer Science, C. Barakat and I. Pratt, Eds., vol. 3015. Antibes Juan-les-Pins, France: Springer, April 2004, pp. 278–288.

[96] X. Zhao and D. Massey, "ON/OFF Model: A New Tool to Understand BGP Update burst," *tech. rep. 04-819, USC-CSD*, August 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.9538

[97] *Network Operating System Evolution*, Juniper, accessed December 2010. [Online]. Available: http://www.juniper.net/us/en/local/pdf/whitepapers/2000264-en.pdf

[98] J. Scudder and C. Appanna, "Multisession BGP," in *draft-ietf-idr-bgp-multisession-03.txt, IETF draft*, Internet Engineering Task Force. United States: IETF, 2007.

[99] C. Estan, A. Akella, and S. Banerjee, "A la carte: An Economic Framework for Multi-ISP Service Quality," in *Tech. Report ID 1591*. United States: University of Wisconsin-Madison, 2007. [Online]. Available: http://pages.cs.wisc.edu/ estan/publications/alacarte-tr.html

[100] A. Yahaya and T. Suda, "iREX: inter-domain QoS automation using economics," in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC 2006)*, vol. 1, Paris, France, January 2006, pp. 96–101.

[101] *FixedOrbit*, accessed August 2009. [Online]. Available: http://fixedorbit.com/

[102] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," RFC 2638 (Informational), Internet Engineering Task Force, Jul. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2638.txt

[103] A. Fonte, M. Pedro, M. Curado, E. Monteiro, and F. Boavida, "Combining Intelligent Route Control with Backbone Traffic Engineering to Deliver Global QoS-enabled Services," in *Recent Advances in Providing QoS and Reliability in the Future Internet Backbone.* Nova Science Publisher, 2011 1st quarter, pp. 149–172.

[104] A. Fonte, J. Martins, M. Curado, and E. Monteiro, "Stabilizing Intelligent Route Control: Randomized Path Monitoring, Randomized Path Switching or History-Aware Path Switching?" in *Proceedings of Management of Converged Multimedia Networks and Services (MMNS)*, ser. Lecture Notes in Computer Science, G. Pavlou, T. Ahmed, and T. Dagiuklas, Eds., vol. 5274. Springer Berlin / Heidelberg, 2008, pp. 151–156, 10.1007/978-3-540-87359-4_14. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87359-4_14

[105] K. Ho, N. Wang, P. Trimintzios, and G. Pavlou, "Traffic Engineering for Interdomain Quality of Service," in *Proceedings of the London Communications Symposium (LCS)*, September 2003.

[106] M. Pedro, E. Monteiro, and F. Boavida, "An approach to off-line inter-domain QoS-aware resource optimization," in *Proceedings of the Networking 2006: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems.* Springer. Lecture Notes in Computer Science Vol. 3976, 2006, pp. 247–255.

[107] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205 (Proposed Standard), Internet Engineering Task Force, September 1997, updated by RFCs 2750, 3936, 4495, 5946. [Online]. Available: http://www.ietf.org/rfc/rfc2205.txt

[108] "ITU-T Recommendation G.107: The E-Model, a computational model for use in transmission planning," 2003.

[109] "ITU-T Recommendation G.114: One-way transmission time," 2003.

[110] "ITU-T Recommendation P.862: Perceptual Evaluation of Speech quality (PESQ): An Objective Method for End-To-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs," 2001.

[111] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The case for separating routing from routers," in *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture (FDNA '04)*. Portland, Oregon, USA: ACM, 2004, pp. 5–12.

[112] M. Yannuzzi, A. Fonte, X. Masip-Bruin, E. Monteiro, S. Sanchez-Lopez, M. Curado, and J. Domingo-Pascual, "A Proposal for Inter-domain QoS Routing Based on Distributed Overlay Entities and QBGP," in *Proceedings of Quality of Service in the Emerging Networking Panorama*, ser. Lecture Notes in Computer Science, vol. 3266. Springer Berlin / Heidelberg, 2004, pp. 257–267. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30193-6_26

[113] *BGP Reports*, accessed September 2011. [Online]. Available: http://bgp.potaroo.net/index-bgp.html

[114] A. Bremler-Barr, Y. Afek, and S. Schwarz, "Improved BGP convergence via ghost flushing," in *Proceedings of the IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, March-3 April 2003, pp. 927–937 vol.2.

[115] W. Sun, Z. Mao, and K. Shin, "Differentiated BGP Update Processing for Improved Routing Convergence," in *Proceedings of the 2006 IEEE International Conference on Network Protocols (ICNP '06)*. Santa Barbara, California, USA: IEEE Computer Society, November 2006, pp. 280–289.

[116] D. Pei, M. Azuma, D. Massey, and L. Zhang, "BGP-RCN: improving BGP convergence through root cause notification," *Computer Networks*, vol. 49, pp. 175–194, 2005.

[117] C. Villamizar, R. Chandra, and R. Govindan, "BGP Route Flap Damping," RFC 2439 (Proposed Standard), Internet Engineering Task Force, November 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2439.txt

[118] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan, "Traffic Matrix Reloaded: Impact of Routing Changes," in *Proceedings of Passive and Active Measurement Workshop*, March/April 2005, pp. 251–264.

[119] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, NY, USA: Cambridge University Press, 2004.

[120] H. Hindi, “A tutorial on convex optimization,” in *Proceedings of American Control Conference*, Boston, USA, June 2004.

[121] ——, “A tutorial on convex optimization II: Duality and interior point methods,” in *Proceedings of American Control Conference*, Minneapolis, Minnesota, USA, June 2006.

[122] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, “A comparison of overlay routing and multihoming route control,” in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '04).* Portland, Oregon, USA: ACM, 2004, pp. 93–106.

[123] S. Uhlig and O. Bonaventure, “Designing BGP-based outbound traffic engineering techniques for stub ASes,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 5, pp. 89–106, 2004.

[124] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, “CapProbe: a simple and accurate capacity estimation technique,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 67–78, 2004.

[125] L.-J. Chen, T. Sun, B.-C. Wang, M. Y. Sanadidi, and M. Gerla, “PBProbe: A capacity estimation tool for high speed networks,” *Computer Communications*, vol. 31, no. 17, pp. 3883–3893, 2008.

[126] *Cisco IOS Netflow*, accessed June 2011. [Online]. Available: www.cisco.com/web/go/netflow

[127] S. Waldbusser, “Remote Network Monitoring Management Information Base Version 2 using SMIv2,” RFC 2021 (Proposed Standard), Internet Engineering Task Force, January 1997, obsoleted by RFC 4502. [Online]. Available: http://www.ietf.org/rfc/rfc2021.txt

[128] G. Box, P. Edward, and G. Jenkins, *Time Series Analysis, Forecasting and Control.* Holden-Day, Incorporated, 1990.

[129] P. Cortez, M. Rio, M. Rocha, and P. Sousa, “Internet Traffic Forecasting using Neural Networks,” in *Proceedings of International Joint Conference on Neural Networks (IJCNN '06)*, Sheraton Vancouver Wall Centre, Vancouver, BC, Canada, July 16-21 2006, pp. 2635 –2642.

[130] Q. He, C. Dovrolis, and M. Ammar, “On the predictability of large transfer TCP throughput,” *Computer Networks*, vol. 51, no. 14, pp. 3959–3977, 2007.

[131] G. Almes, S. Kalidindi, and M. Zekauskas, “A One-way Delay Metric for IPPM,” RFC 2679 (Proposed Standard), Internet Engineering Task Force, September 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2679.txt

[132] ——, "A One-way Packet Loss Metric for IPPM," RFC 2680 (Proposed Standard), Internet Engineering Task Force, September 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2680.txt

[133] W. Fang and L. Peterson, "Inter-AS traffic patterns and their implications," in *Proceedings of the Global Telecommunications Conference (GLOBECOM '99)*, vol. 3, Rio de Janeiro, Brazil, 1999, pp. 1859–1868.

[134] V. Jacobson, "Congestion avoidance and control," in *Symposium proceedings on Communications architectures and protocols (SIGCOMM '88)*. Stanford, California, United States: ACM, 1988, pp. 314–329.

[135] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[136] R. A. P. David F. Williamson and J. S. Kendrick, "The box plot: A simple visual method to interpret data," *Annals of Intemal Medicine*, vol. 110, no. 1, pp. 916–921, June 1989.

[137] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer," RFC 2988 (Proposed Standard), Internet Engineering Task Force, November 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2988.txt

[138] M. Menth, R. Martin, and J. Charzinski, "Capacity overprovisioning for networks with resilience requirements," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 87–98, 2006.

[139] R. Martin, M. Menth, and J. Charzinski, "Comparison of border-to-border budget based network admission control and capacity overprovisioning," in *Proceedings of 2005 IFIP Networking Conference*, ser. Lecture Notes in Computer Science, R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. Black, Eds. Springer Berlin/Heidelberg, 2005, vol. 3462, pp. 1056–1068. [Online]. Available: http://dx.doi.org/10.1007/11422778_85

[140] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory algorithms and applications*. Prentice Hall, Inc., 1993.

[141] M. R. Garey and D. S. Johnson, *Computers and intractability; a guide to the theory of NP-completeness*. W.H. Freeman, 1979.

[142] M. Curado, "Quality of Service Routing for Class-Based Networks," Ph.D. dissertation, University of Coimbra, 2005.

[143] J. Figueira *et al.*, *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer Science+Business Media, 2005.

[144] L. Burgstahler and M. Neubauer, "New modifications of the exponential moving average algorithm for bandwidth estimation," in *Proceedings of the 15th ITC Specialist Seminar*, Wurzburg, Germany, 2002, pp. 210–219.

[145] *J-Sim Homepage*, accessed September 2011. [Online]. Available: http://sites.google.com/site/jsimofficial/

[146] *Infonet Suite*, accessed September 2011. [Online]. Available: https://sites.google.com/site/jsimofficial/3rd-party-contributions–add-ons

[147] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771 (Draft Standard), Internet Engineering Task Force, March 1995, obsoleted by RFC 4271. [Online]. Available: http://www.ietf.org/rfc/rfc1771.txt

[148] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597 (Proposed Standard), Internet Engineering Task Force, June 1999, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2597.txt

[149] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598 (Proposed Standard), Internet Engineering Task Force, June 1999, obsoleted by RFC 3246. [Online]. Available: http://www.ietf.org/rfc/rfc2598.txt

[150] H. Alshaer and E. Horlait, "Expedited Forwarding Delay Budget Through a Novel Call Admission Control," in *Proceedings of Universal Multiservice Networks: Third European Conference (ECUMN 2004)*, ser. Lecture Notes in Computer Science, vol. 3262. Porto, Portugal: Springer, October 2004, pp. 50–59.

[151] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: an approach to universal topology generation," in *Proceedings of the Ninth International Symposium on-Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Cincinnati, OH, USA, August 2001, pp. 346–353.

[152] M. Yannuzzi, X. Masip-Bruin, A. Fonte, M. Curado, and E. Monteiro, "On the Advantages of Cooperative and Social Smart Route Control," in *Proceedings of the 15th International Conference on Computer Communications and Networks, 2006 (ICCCN 2006)*, October 2006, pp. 197–203.

[153] M. Yannuzzi, X. Masip-Bruin, E. Marin-Tordera, J. Domingo-Pascual, A. Fonte, and E. Monteiro, "Improving the Performance of Route Control Middleboxes in a Competitive Environment," *IEEE Network*, vol. 22, no. 5, pp. 56 –64, September-October 2008.

[154] *Avaya, Inc., Converged Network Analyser*, accessed June 2011. [Online]. Available: http://www.avaya.com/usa/

[155] D. K. Goldenberg, L. Qiuy, H. Xie, Y. R. Yang, and Y. Zhang, “Optimizing cost and performance for multihoming,” in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM ’04)*. Portland, Oregon, USA: ACM, 2004, pp. 79–92.

[156] A. Akella, S. Seshan, and A. Shaikh, “Multihoming performance benefits: an experiment evaluation of practical enterprise strategies,” in *Proceedings of the annual conference on USENIX Annual Technical Conference (ATEC ’04)*. Boston, MA, USA: USENIX Association, 2004, pp. 9–9.

[157] M. Yannuzzi, “Strategies for Internet Route Control: Past, Present, and Future,” Ph.D. dissertation, Technical University of Catalonia, Barcelona, Spain, 2007.

[158] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, “Open issues in interdomain routing: a survey,” *IEEE Network*, vol. 19, no. 6, pp. 49 – 56, November/December 2005.

[159] B. M. Waxman, “Routing of multipoint connections,” *Broadband switching: architectures, protocols, design, and analysis*, pp. 347–352, 1991.

[160] X. M.-B. E. M. S. S.-L. M. C. M. Yannuzzi, A. Fonte and J. Domingo-Pascual, “A self-adaptive interdomain traffic engineering scheme,upc-dac-rr-cba-2005-8,” Department of Computer Architecture, Technical University of Catalonia (UPC), Barcelona, Spain, Paper, December 2005. [Online]. Available: http://www.rand.org/cgi-bin/Abstracts/ordi/getabbydoc.pl?doc=P-923

[161] A. Fonte, M. Pedro, E. Monteiro, and F. Boavida, “Analysis of Interdomain Smart Routing and Traffic Engineering Interactions,” in *Proceedings of Global Telecommunications Conference, 2007 (IEEE GLOBECOM ’07)*, nov. 2007, pp. 1828 –1833.

[162] ——, “Improving Inter-AS Quality of Service through Multi-homing Smart Routing: Tackling the Interaction with Traffic Engineering,” in *Tutorial and Ph.D. Student Workshop of Med Hoc Net 2007*, Ionian University, Corfu, June 2007.

[163] A. Fonte, E. Monteiro, M. Yannuzzi, X. Masip-Bruin, and J. Domingo-Pascual, “A framework for Cooperative Inter-Domain QoS Routing,” in *EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World*, ser. IFIP International Federation for Information Processing, C. Kloos, A. Marin, and D. Larrabeiti, Eds. Springer Boston, 2006, vol. 196, pp. 91–104.

[164] ——, “A cooperative approach for coordinated inter-domain QoSR decisions,” in *11th Open European Summer School and IFIP WG6.6, WG6.4, WG6.9 Workshop on Networked Applications (EUNICE 2005: Networked Applications)*, University Carlos III of Madrid, Colmenarejo, Spain, July 2005, pp. 133–137.

[165] A. Dhamdhere and C. Dovrolis, "ISP and Egress Path Selection for Multihomed Networks," in *Proceedings of the IEEE INFOCOM 2006, 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies*, Barcelona, Spain, November/December 2006.

[166] T. Bressoud, R. Rastogi, and M. Smith, "Optimal configuration for BGP route selection," in *Proceedings of the IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, San Franciso, CA, USA, March-3 April 2003, pp. 916–926.

[167] T. Roughgarden and E. Tardos, "How bad is selfish routing?" *Journal of the ACM (JACM)*, vol. 49, no. 2, pp. 236–259, 2002.

[168] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031 (Proposed Standard), Internet Engineering Task Force, January 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3031.txt

[169] R. Mahajan, D. Wetherall, and T. Anderson, "Negotiation-based routing between neighboring ISPs," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation (NSDI'05)*. Berkeley, CA, USA: USENIX Association, 2005, pp. 29–42.

[170] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

[171] A. O. Dantzig, G.B. and P. Wolfe, "Generalized simplex method for minimizing a linear form under linear inequality restraints," *Pacific Journal Math.*, vol. 5, pp. 183–195, 1955.

[172] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182 –197, April 2002.

[173] *MATLAB*, MathWorks Inc., accessed September 2011. [Online]. Available: http://www.mathworks.com/products/matlab/

[174] L. A. Adamic and B. A. Huberman, "Zipf's law and the Internet," *Glottometrics*, vol. 3, pp. 143–150, 2002. [Online]. Available: http://www.hpl.hp.com/research/idl/papers/ranking/adamicglottometrics.pdf

[175] A. Kuzmanovic, A. Mondal, S. Floyd, and K. Ramakrishnan, "Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets," RFC 5562 (Experimental), Internet Engineering Task Force, June 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5562.txt

[176] "IEEE 802.3: CSMA/CD (ethernet) ACCESS METHOD," 2008. [Online]. Available: http://standards.ieee.org/about/get/802/802.3.html

[177] D. E. Shiraev, D. N. Ramakrishnan, C. chair Dr, S. Varadarajan, D. Calvin, J. Ribbens, and D. Shiraev, "Inverse Reinforcement Learning and Routing Metric Discovery," 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.107.2061

[178] C. Bouras and A. Sevasti, "SLA-based QoS pricing in Diffserv networks," *Computer Communications*, vol. 27, no. 18, pp. 1868 – 1880, 2004, performance and Control of Next Generation Communications Networks. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYP-4CY5JYJ-1/2/8d372a5e24066cc82b41829df2e868c8

[179] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157 (Historic), Internet Engineering Task Force, May 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1157.txt

[180] X. Wang and H. Schulzrinne, "Pricing network resources for adaptive applications," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 506–519, 2006.

[181] S. Floyd, R. Gummadi, and S. Shenker, *Adaptive RED: an algorithm for increasing the robustness of RED's Active Queue Management*, August 2001. [Online]. Available: http://www.icir.org/f̃loyd

[182] F. Benford, "The Law of Anomalous Numbers," in *Proceedings of the American Philosophical Society*, vol. 78, no. 4, 1938, pp. 551–572.

[183] H. R. Varian, *Microeconomics Analysis, third edition.* W. W. Norton and Company, Inc., 1992.

[184] D. Eastlake 3rd, "Domain Name System (DNS) IANA Considerations," RFC 5395 (Best Current Practice), Internet Engineering Task Force, November 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5395.txt

[185] O. M. Group, "The Common Object Request Broker: Architecture and Specification, 3.0.2 edition," 2002.

[186] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 83–86, 2006.

[187] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: methodology and experience," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 265–280, 2001.

[188] S. Bhattacharyya, C. Diot, and J. Jetcheva, "Pop-level and access-link-level traffic dynamics in a tier-1 POP," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IMW '01).* ACM, 2001, pp. 39–53.

[189] A. Clauset, C. Shalizi, and M. Newman, "Power-law distributions in empirical data," *SIAM Review, available in http://www.santafe.edu/ aaronc/powerlaws/*, 2009.

[190] A. Fonte, M. Curado, and E. Monteiro, "Prediction error-based criterion for selecting popular destinations," in *Proceedings of (to appear) the 3rd IEEE International Workshop on Management of Emerging Networks and Services (IEEE MENS 2011) in conjunction with IEEE GLOBECOM 2011*, Houston, Texas, USA, 2011.

[191] *GEANT network*, accessed September 2011. [Online]. Available: http://www.geant.net/Network/NetworkTopology/pages/home.aspx

[192] "ITU-T Recommendation G.803: Achitecture of transport networks based on the synchronous digital hierarchy (SDH)," 1997.

[193] P. V. Mieghem, *Performance Analysis of Communications Networks and Systems.* New York, NY, USA: Cambridge University Press, 2005.

[194] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, ser. MIT Press Books. The MIT Press, June 1994, vol. 1, no. 0262650401.

[195] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke, "Optimal oblivious routing in polynomial time," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, ser. STOC '03. New York, NY, USA: ACM, 2003, pp. 383–388. [Online]. Available: http://doi.acm.org/10.1145/780542.780599

[196] M. Amin, K. hon Ho, M. P. Howarth, and G. Pavlou, "An Integrated Network Management Framework for Inter-domain Outbound Traffic Engineering," in *Proceedings of International Conference on Management of Multimedia Networks and Services*, 2006, pp. 208–222.

[197] S. Uhlig, "A multiple-objectives evolutionary perspective to interdomain traffic engineering," *International Journal of Computational Intelligence and Applications*, 2005.

[198] K. Park and W. Willinger, *Self-similar network traffic and performance evaluation.* Wiley-Interscience, 2000.