UNIVERSIDADE NOVA DE LISBOA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores

MULTI-REGION ROUTING

Por:
Francisco José Dinis de Sousa Fernandes Ganhão

Dissertação apresentada na Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa para a obtenção do grau
de Mestre em Engenharia Electrotécnica e de Computadores

Orientador: Doutor Luís Filipe Lourenço Bernardo
Co-Orientador: Doutor Paulo da Costa Luis da Fonseca Pinto

LISBOA
(2009)

To my Mother
and Grandmother

Abstract

This thesis proposes a new inter-domain routing protocol. The Internet's inter-domain routing protocol Border Gateway Protocol (BGP) provides a reachability solution for all domains; however it is also used for purposes outside of routing. In terms of routing BGP suffers from serious problems, such as slow routing convergence and limited scalability.

The proposed architecture takes into consideration the current Internet business model and structure. It benefits from a massively multi-homed Internet to perform multipath routing. The main foundation of this thesis was based on the Dynamic Topological Information Architecture (DTIA). We propose a division of the Internet in regions to contain the network scale where DTIA's routing algorithm is applied. An inter-region routing solution was devised to connect regions; formal proofs were made in order to demonstrate the routing convergence of the protocol.

An implementation of the proposed solution was made in the network simulator 2 (ns-2). Results showed that the proposed architecture achieves faster convergence than BGP. Moreover, this thesis' solution improves the algorithm's scalability at the inter-region level, compared to the single region case.

Resumo

Nesta tese é proposto um novo protocolo de encaminhamento para inter-domínio. O protocolo usado na Internet para realizar o encaminhamento inter-domínio é o Border Gateway Protocol (BGP). O BGP fornece uma solução baseada na alcançabilidade dos domínios; contudo o seu uso é estendido a vários propósitos não relacionados com o encaminhamento. No que diz respeito a encaminhamento, este protocolo sofre de problemas graves, como convergência lenta de encaminhamento e escalabilidade limitada.

A arquitectura proposta toma em consideração o actual modelo de negócios da Internet e a sua estrutura. Beneficia do facto da Internet estar massivamente *multi-homed* (cada nó tem múltiplos fornecedores) para oferecer encaminhamento multi-caminho, isto é, aproveita múltiplos caminhos no encaminhamento para um destino. A estrutura fundamental desta tese assenta no protocolo Dynamic Topological Information Architecture (DTIA). É proposta uma divisão da Internet em regiões para conter a escala da rede onde o algoritmo de encaminhamento do protocolo DTIA é usado. Uma solução de encaminhamento inter-região foi desenvolvida para ligar regiões; foram feitas provas formais para demonstrar a convergência de encaminhamento do protocolo.

A implementação da solução proposta foi feita no simulador network simulator 2 (ns-2). Os resultados mostraram que a arquitectura converge mais rapidamente que o BGP. Além do mais, a solução desta tese melhora a escalabilidade do algoritmo ao nível inter-região face a uma única região.

Acknowledgements

I would like to acknowledge several people that helped me through the development and writing of my thesis.

First I would like to show my gratitude towards Prof. Luís Bernardo for its constant guiding, patience and companionship throughout my dissertation. I would also like to thank him for his selfless attitude, whilst on a busy schedule; otherwise this dissertation's quality would be inferior.

Prof. Pedro Amaral was also of valuable help during the development of my thesis. I would like to thank him for his thorough knowledge and method, otherwise I would not accomplish this lengthy milestone.

I would like to thank Prof. Paulo Pinto for his great experience and tact towards scientific knowledge. All meetings I had with him were very instructive, it made me realize a different perspective of knowledge itself.

Cláudio Assunção's companionship was also of valuable help during the development of my dissertation. His knowledge of the Internet and of the ns-2 simulator were a great help for the initial steps of this thesis.

I would like to thank all of my colleagues from FCT-UNL that gave me support since the beginning of my thesis: Ricardo Gomes, José Luzio, Hugo Lopes, António Rocha, João Melo, Michel Rodrigues, Miguel Pereira, Miguel Luís, Michael Figueiredo, José Custódio, Edinei Santini, Tiago Oliveira, Pedro Magalhães, Gonçalo Luís, Nuno Pereira, João Antunes, José Belo and *NEEC* 08/09.

***

I would like to thank my closest friends for their support: Filipe Dias and Dalila Forte. Finally, I would like to show the most kind and special gratitude to my family: my Mother and Grandmother. Even with the illness they face everyday, they have always supported me during the most difficult times of my life.

To all of you, thank you!

Almada, September 2009
Francisco Ganhão

Acronyms

AS Autonomous System

ASes Autonomous Systems

BGP Border Gateway Protocol

C2P Customer to Provider

CAIDA Cooperative Association for Internet Data Analysis

CDN Content Delivery Network

CIDR Classless Inter Domain Routing

DNS Domain Name System

DTIA Dynamic Topological Information Architecture

DV Distance Vector

eBGP External Border Gateway Protocol

EID Endpoint IDentifier

EIDs Endpoint IDentifiers

ETR Egress Tunnel Router

FCP Failure Carrying Protocol

FPV Fragmented Path-Vector

G-ISP Global Internet Service Provider

GNU GNU is Not Unix

HLP Hybrid Link-State Protocol

iBGP Internal Border Gateway Protocol

ICMP Internet Control Message Protocol

IEEE Institute of Electrical and Electronics Engineers

IGP Internal Gateway Protocol

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IRR Internet Routing Registry

IS-IS Intermediate System - Intermediate System

ISP Internet Service Provider

ISPs Internet Service Providers

ITR Ingress Tunnel Router

LISP Locator ID Separation Protocol

LSPV Local Simulated Path Vector

MED Multi Exit Discriminator

MRAI Minimum Route Advertisement Interval

NIRA New Inter-domain Routing Architecture

NRLS Name-to-Route Lookup Service

ns-2 network simulator 2

OSI Open System Interconnection

OSPF Open Shortest Path First

P2C Provider to Customer

P2P Peer to Peer

P2Patt Peer to Peer allowing transit traffic

P2Pbkp Peer to Peer allowing backup

PoP Point of Presence

QoS Quality of Service

RFC Request for Comments

RIP Routing Information Protocol

RIPE Réseaux IP Européens

RLOC Routing Locator

RLOCs Routing Locators

RPSL Routing Policy Specification Language

RTT Round Trip Time

S2S Sibling to Sibling

SLA Service Level Agreement

TIPP Topology Information Propagation Protocol

TCP Transport Control Protocol

UDP User Datagram Protocol

VoIP Voice over IP

VPN Virtual Private Network

VPNs Virtual Private Networks

XL Approximate Link-State

Contents

Abstract **v**

Resumo **vii**

Acknowledgements **ix**

Acronyms **xi**

1 Introduction **1**

1.1 Current context 2

1.2 Hypothesis 2

1.3 Objectives and Contributions 3

1.4 Dissertation's structure 3

2 Related Work **5**

2.1 Internet Business Model 5

2.1.1 Autonomous System 5

2.1.2 Common policies 6

2.2 Internet Characterization 7

2.2.1 Topology information sources 8

2.2.2 Inferring the Internet, an overview 9

2.3 *Traditional* Routing Protocols 11

2.3.1 Intra-Domain Protocols 11

2.3.2 Border Gateway Protocol (BGP): An Inter-Domain Protocol 13

2.4 Academical Solutions 16

2.4.1 Routing Independent Solutions 16

2.4.2 Intra-domain protocol extensions 19

2.4.3 New Inter-domain protocols 20

2.5 Summary 22

3 An approach to multi-region routing 25

3.1 Introduction 25

3.2 Dynamic Topological Information Architecture (DTIA) - Basic Rationale . 26

3.2.1 Model Assumptions 26

3.2.2 Design Choices 27

3.3 Architecture 29

3.3.1 Reachability 30

3.3.2 Routing 33

3.3.3 Failure Management 39

3.4 Multi-region Routing 43

3.4.1 Region definition 44

3.4.2 Inter-Region Routing 45

3.4.3 Inter-Region Routing Correctness 48

3.4.4 Failure Management 50

3.4.5 Deployment 51

4 Protocol's Implementation 53

4.1 Introduction 53

4.2 Protocol's Considerations 53

4.3 Algorithms 54

4.3.1 Path Exploration: Intra-region 54

4.3.2 Path Exploration: Inter-Region 64

4.3.3 Failure Management Algorithm 67

4.4 The network simulator 2 (ns-2) 69

5 Performance Analysis 75

5.1 Introduction 75

5.2 Topology Characteristics 75

5.3 Experiments 83

5.3.1 Routing table analysis 83

5.3.2 Packet signalling analysis 86

5.3.3 Packet delay analysis 89

6 Conclusions **91**

6.1 Synthesis 91

6.2 Conclusions 92

6.3 Future work 93

A DTIA: Routing at the Interdomain level **95**

Bibliography **111**

List of Tables

2.1 BGP's rules for tie-break decision about which route to choose 14

2.2 Protocols Summary 24

3.1 Path Validation for $D = 1$ 32

3.2 Path Validation for $D = 0$ 32

3.3 $\oplus$ binary operation, at the leftmost column we have the appending link and at the upmost row we have the path's signature 35

3.4 Ranking Order 35

3.5 $\oplus$ binary operation for Multi-Region, at the leftmost column we have the appending link and at the upmost row we have the path's signature. 48

3.6 Order of preference for inter-region traffic. 48

4.1 Path P's general structure. 55

4.2 Path P_1's tuples ordered by index. 55

4.3 Path P_2's tuple. 55

4.4 Paths that traversed X_j with a certain direction D_w, signature S_w and link label l_w. 56

4.5 Best First Hop tuples that reach AS X_j 57

4.6 Best first hop tuples that arrive destination X_j. 63

4.7 Ranking of the first hop tuples that arrived X_j. 64

4.8 Best Region First Hops at AS X_j. This table can also be used for a region R. 65

4.9 Restrictions table from AS X. 72

5.1 Nodes identification from figure 5.1 78

5.2 Number of Inter-region Links of each type. 82

5.3 Mean value of destinations that are reached remotely from border routers that have strict reachability. 84

List of Figures

2.1 Network example of three Autonomous Systems (ASes) 6

2.2 Network example of Provider to Customer (P2C) policy 7

2.3 The *count to infinity problem* on a loop topology. 12

2.4 Applying business relationships with communities. 15

3.1 Path exploration with backup links 33

3.2 Signature calculation by walking in the forward direction. 39

3.3 Network example of a routing loop if link A-B fails. 40

4.1 Figure exemplifying the data structures at each paths' index. 56

4.2 Path Initialization procedure 58

4.3 Path Exploration General Algorithm. 59

4.4 Flowchart that illustrates the processing of path P_w. 60

4.5 Merge of a path P_w Algorithm. 62

4.6 Decision process to disseminate a control packet. 68

4.7 Packet forwarding issues using a normal classifier. 72

5.1 ns/nam topology. Links in cyan are peering links; grey links represent provider to customer relationships from top to bottom. Nodes in yellow belong to region R_0 and nodes in white belong to region R_1. 77

5.2 ASes' degree distribution in percentage. 79

5.3 Comparison of ASes' degree for intra-region links between regions R_0 and R_1. 81

5.4 Comparison of ASes' degree for inter-region links between regions R_0 and R_1. 82

5.5 Comparison of the number of routing entries for both intra-region and multi-region. 84

5.6 Comparison of the number of first hops that allow full and strict reachability for multi-region routing. 85
5.7 Comparison of available intra-region paths for a single region in blue and for two regions in red. 86
5.8 Cumulative percentage of affected ASes. DTIA 1 presents the results for a single-region; for two regions DTIA $2i$ presents the results for intra-region links and DTIA $2b$ for inter-region links. 88
5.9 Cumulative percentage of affected ASes. The blue curve presents the results for a single-region, as for the red curve it presents the results for two-regions. 89
5.10 Cumulative percentage of the number of hops. The blue curve presents the results for a single-region, as for the red curve it presents the results for two-regions. 90

Chapter 1

Introduction

The Internet has grown at an *explosive* rate since its birth; such growth has pressured Internet Service Providers (ISPs) to carefully plan their networks against insufficient bandwidth and lack of routing capacity. Back in 1995, the common press *crucified* the Internet, announcing that the Internet would disrupt if this growth continued[met].

Today the Internet is still *alive*, but suffers from serious routing instability. The instability was measured for Voice over IP (VoIP) applications, which are more sensitive to routing changes[KKK07]. The Border Gateway Protocol (BGP) is currently used on the Internet as a solution for inter-domain routing. ISPs also use it also for other purposes different than routing, such as load balancing or prefix-based Virtual Private Networks (VPNs). Furthermore, the current Internet's structure is massively multi-homed, where customers are connected to more than one provider; however BGP cannot take full advantage of the multiplicity of connections to *communicate* with a given destination.

In this thesis, it is discussed the problematic of developing an architecture for inter-domain routing that solves the aforementioned issues without disrupting the current Internet. A routing protocol should be modular and not *monolithic* to separate different functionalities. It is also essential that routing capacity is not undermined with the growth of the Internet. Besides, new methods should be devised to improve routing stability and to take full advantage of multi-homing.

The main goal of this dissertation is to address the aforementioned requirements and devise a new architecture for inter-domain routing.

1.1 Current context

Recently, there have been academical solutions that tried to cope with the current Internet. Hybrid Link-State Protocol (HLP)[SCE$^+$05] and New Inter-domain Routing Architecture (NIRA)[YCB07] brought an *innovative* idea of partitioning the Internet's topology in regions. Both intended to contain the network's scale where the routing protocol is applied, however they were unsuccessful to cope with the current Internet business model. Besides, routers needed to hold and compute the topology information for more than one region; such *overload* of routing computation should be avoided.

Shortly after came an interesting academical proposal, Dynamic Topological Information Architecture (DTIA)[ABP08]. This protocol has some interesting features:

1. The protocol separates reachability, routing and traffic engineering as separate functionalities;
2. The protocol takes full advantage of multi-homing;
3. A robust failure management algorithm that eases routing stability;
4. The protocol is adapted to the current Internet business model.

The first three features are a response to BGP's flaws; regarding the fourth it covers the gap that NIRA and HLP did not cover. However, DTIA fails to scale as the network size grows.

The proposed solution of this dissertation is largely based on the DTIA protocol; HLP and NIRA were also an *inspiration* as they use regions to contain the network's scale with regions. As an important note, the proposed implementation works around the computational *overload* of NIRA and HLP.

1.2 Hypothesis

From the current context of the Internet in section 1.1, the following hypothesis are formulated: If a routing architecture is capable of handling the current Internet's structure and business model in a modular manner then we should take full advantage of multi-path routing through multi-homing. Furthermore, if the architecture can scale with the Internet's growth then we can expect better routing stability.

1.3 Objectives and Contributions

The main goal of this thesis is to prove the feasibility of implementing an inter-domain routing protocol that complies with the hypothesis of section 1.2.

The main contributions of this dissertation are the definition of a new inter-region protocol for the DTIA architecture, supported by a formal model, and its full implementation on the network simulator 2 (ns-2). The theoretical model and its respective results of the ns-2 simulator in chapter 5 were published at IEEE *Globecom'09* conference[AGA$^+$09].

1.4 Dissertation's structure

The dissertation is structured in six chapters and one appendix, each one is enumerated on the following paragraphs.

Chapter 2 presents the Internet's business model and a chronological vision of the Internet's structure. Furthermore, a comparison is made between currently used protocols and academical proposals. At the end it is discussed a brief overview of the state of the art. Chapter 3 overviews the DTIA protocol and presents this dissertation's theoretical model and respective assumptions.

Chapter 4 adds some considerations to the DTIA architecture and explains the proposed implementation on the ns-2 simulator through the visual aid of flowcharts.

Chapter 5 does a performance analysis of the ns-2 implementation of the proposed model. Based on the results, this chapter tries to correlate the data with chapters 3 and 4.

Chapter 6 does a global analysis of this dissertation, based on the hypothesis. At the end, this chapter enumerates a few topics that need further work based on this thesis.

Appendix A presents a technical report of the DTIA architecture.

Chapter 2

Related Work

2.1 Internet Business Model

Nowadays the Internet is a decentralized arrangement of networks that evolved over the years as a consequence of the increasing number of consumers and demanding services. It brought an unavoidable need for local providers to comply with this demand and to serve a broader audience than just corporate enterprises.

Due to the nature of services, local providers cannot offer a complete solution to their customers and must rely on other providers to relay traffic. A similar structure to the one of the telephone network has been created with long-haul providers. Therefore service contracts were established between these entities and policies were used to control the routing decisions of traffic.

A brief overview of the Internet Business Model is made, introducing the concept of Autonomous System and the inherent common policies that manage most of the contracts.

2.1.1 Autonomous System

Internet consumers seek the services of their local providers to meet their needs. The providers may use the services of other higher rank providers, or connect directly to the remote providers in a peer-to-peer approach. These entities have under their responsibility a collection of sub-domains or networks to maintain, also known as Autonomous System (AS).

An Autonomous System has a crucial role in terms of intra-domain and inter-domain networking, since its structure and configuration influence how downstream/upstream traffic is handled, and this is reflected on the policies at the Inter-domain level with other Autonomous Systems (ASes).

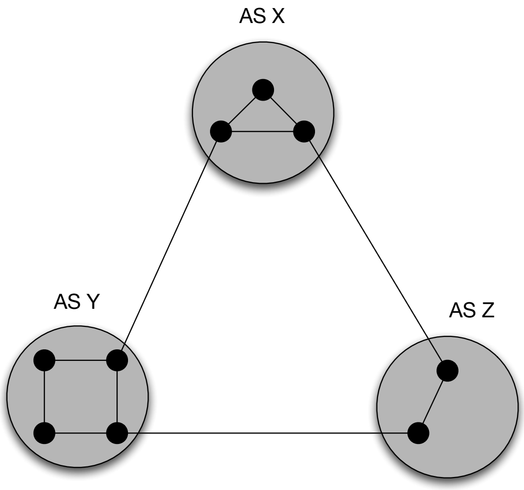

Figure 2.1: Network example of three ASes

In figure 2.1 we have an example of three ASes depicted as grey circles, while the black dots are the ASes routers. Routers that *bridge* ASes are known as border-routers.

Acquainted with the notion of Autonomous System, section 2.1.2 describes thoroughly the policy agreements between ASes.

2.1.2 Common policies

Gao's work [Gao01] identified a small set of relationships between ASes that covers 99% of the interconnections between ASes; these relationships are also known as common policies and are mainly of three types:

1. Provider-Customer,
2. Peer to Peer (P2P),
3. Sibling to Sibling (S2S).

The Provider-Customer relationship defines a paid service rendered by a provider to carry traffic from a customer domain. Assuming a link failure between ASes A and D in figure

2.2, they can use provider C to reach each other. However a customer AS with more than one provider does not carry traffic between its providers. For example in figure 2.2, AS A does not route traffic between providers C and B.

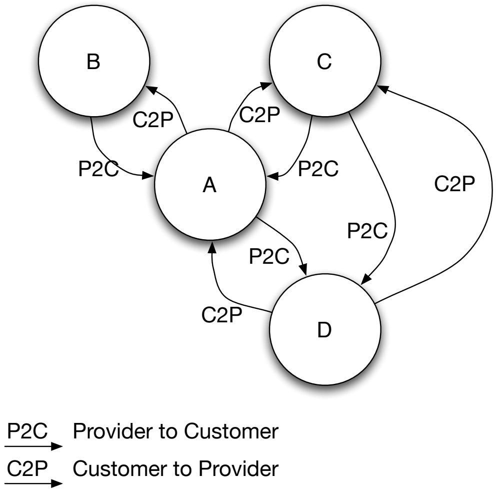

Figure 2.2: Network example of P2C policy

With respect to the P2P relationship, it defines an agreement between two domains to carry traffic from each other and from their customers exclusively. As an example two ASes A and B that exchange traffic through a provider C, could establish a P2P relationship to diminish their costs (if possible).

Finally, a Sibling to Sibling (S2S) relationship defines a mutual agreement between domains to provide connectivity to each other. However, this relationship could be used as a backup, assuring that a *sibling* does not loose connectivity from the rest of the Internet.

2.2 Internet Characterization

Drawing a picture of the Internet's topology can be a daunting task because the relationships are not explicitly stated. Therefore we need to ensure that we have the appropriate data sources to infer the relationships. Section 2.2.1 describes how viable are these information sources, followed by a chronological discussion of the Internet's structure on section 2.2.2.

2.2.1 Topology information sources

Several information sources can be used to infer characteristics from the Internet, albeit each one has a different level of abstraction. This section describes three levels of abstraction: IP level, Router level and AS level.

IP level

Starting at the IP level of abstraction, we can use the traceroute tool to probe several nodes in the Internet, by tracing the path of an UDP or ICMP *echo* for a given destination IP. CAIDA, for instance, developed a tool by means of *tracerouting* called *skitter* [ea02a] that collects information from multiple vantage points spread around the world probing the same destination set of IPv4 addresses.

Although *tracerouting* may sound simple, it has its own limitations, since it only discovers *forward* paths and *reverse* paths may differ from these[DF07].

Router level

Moving up one level of abstraction we have the router level. Since a router has multiple network interfaces, each one is *abstracted* as a distinct node at the IP level. Probing techniques are therefore employed to differentiate routers; these techniques map a set of address *aliases* with a router's interface.

For example, the *Mercator* [GT00] tool uses ICMP packets[Bra89] to probe network interfaces, thus attempting to infer the network's topology. Opposed to this methodology *Ally* [NSW02] looks for similarities from routers' host names using DNS information.

Besides increasing the network load due to the probing of network interfaces, routers might not respond to ICMP packets. Nevertheless, it is possible to infer the network's topology with *offline* data [DF07].

AS level

At the AS level, there are two possible information sources: Routing Registries Information and BGP Routing Information.

Regional Internet Registries [rir09] are an example of Routing Registries, that expose Inter-domain information through the *WHOIS* protocol [Dai04]. It is also possible to obtain *normalized* data in the Routing Policy Specification Language (RPSL) [ea99] using an Internet Routing Registry (IRR) [Dat09]. Despite the accessibility of Routing Registries, their content does not reveal the lack of temporary network flaws [NCC09].

Alternatively, BGP Routing Information can be found on *looking glass servers*, or *route servers* from ASes. *BGP table dumps* are also a viable alternative to these servers; the *RouteViews* [oO09] project or the *RIPE NCC* are perfect examples of routing information collected from BGP routers around the world.

BGP Routing Information holds an advantage over Routing Registries, since each BGP router locally portrays the actual state of the network, although it is hard to make any statements about ASes relationships[ea02b].

2.2.2 Inferring the Internet, an overview

This section exposes a chronological overview of research studies that inferred the Internet's structure.

Power Law: The Beginning

In 1999 *Faloutsos et al.* [FFF99] reported that the Internet's structure follows a power-law distribution. A power-law distribution c is described as:

$$c \propto A^t, \tag{2.1}$$

with A being the metric following a power-law, according to a characteristic value t. The topology generator *BRITE* [bri09] is based on this assumption although its design is only appropriate for large scale networks. Despite the power-law assumption, *Chen et al.* [ea02c] suggests that the data collected from the *RouteViews* project follows a heavy-tailed distribution for the nodes' degree.

Inferring the Internet as an Hierarchy

The following studies identify the Internet as a hierarchy. *Gao*'s work[Gao01] inferred ASes relationships based on this assumption through the *RouteViews* data; this work assumes that forwarded routing information follows a valley-free path. A valley-free path means that routing information coming from a peer or provider is not forwarded to another peer or provider. Nevertheless she found data inconsistencies on her research.

Gao's work was further continued in 2002 by *Subramanian et al.*[SARK02], inferring ASes relationships from multiple vantage points. This research also presented a mechanism to classify ASes at different levels of an hierarchy. At the same year, *Vazquez et al.*[VPSV02] studied the hierarchical properties of the Internet and its correlation with the nodes' connectivity, presenting a scale-free distribution that follows a power-law distribution with t varying between 2 and 3.

Examples of existent hierarchical topology generators are: *GT-ITM* [gti09] and *IGen* [ige09].

Latest conclusions

On 2006, CAIDA members [MKF$^+$06] found that Internet metrics following a power-law distribution are not correlated with hierarchical layers. They also discovered that hierarchical generators are not suited to create synthetic topologies. This work remarks that not all Internet metrics follow a pure power-law distribution but instead a scale-free distribution.

Later the authors argued [DKF$^+$07] if the existent common policies are capable of modelling/inferring a topology. To answer this question, CAIDA's members contacted *small* ASes administrators, and verified that established relationships with other domains might be *hybrid*, *e.g* an AS that has a P2P relationship, might use it as a sibling relationship for backup purposes.

As a note, the data [cai09b] inferred from CAIDA could be also useful for simulation purposes.

2.3 *Traditional* Routing Protocols

At the network layer, routing protocols perform an important role: to route packets between remote hosts. These protocols, through the exchange of routing messages, build a set of local routes to *any* destination. This section presents currently used protocols at two distinct levels: Intra-domain and Inter-domain.

2.3.1 Intra-Domain Protocols

The current section presents the most common intra-domain protocols: Distance Vector (DV) and Link-State, along with their limitations and advantages.

Distance Vector (DV)

Routers in the old days of the *ARPANET* [MW77] used a Distance Vector (DV) protocol based on the Bellman-Ford Algorithm[tan02], called Routing Information Protocol (RIP) v1 [Mal98].

The RIP protocol maintains a routing table with a 3-tuple information to reach each destination D_i with cost C_i, (D_i, C_i, N_i), with N_i as the next hop to forward. To ensure reachability to all nodes in a network, each node sends a table to its neighbours in a timely manner, containing the 2-tuple (D_i, C_i) .

Once a node receives a table from a neighbour D_k, it compares the pairs: (D_i, C_i) and $(D_i, C_{ki} + C_k)$ for each destination D_i, with C_{ki} as the cost from neighbour D_k to destination D_i.

If the cost C_i is higher than $(C_{ki} + C_k)$, the routing table will be altered with the latter cost and neighbour D_k as the next hop for destination D_i. Although the concept is quite simple and scalable, it has the known problem of *Count to Infinity*[CRK89]. To better illustrate this problem, figure 2.3 has a loop topology, where link $A - C$ fails. Since B is unaware of this change it announces to C that it can reach A with a metric delay of 2. Node C is whatsoever oblivious of B's next node, so C updates its cost to A with a metric delay of 3. This behaviour repeats back and forth until the cost of reaching A at B's routing table reaches infinity. Similarly this example would also feedback itself to

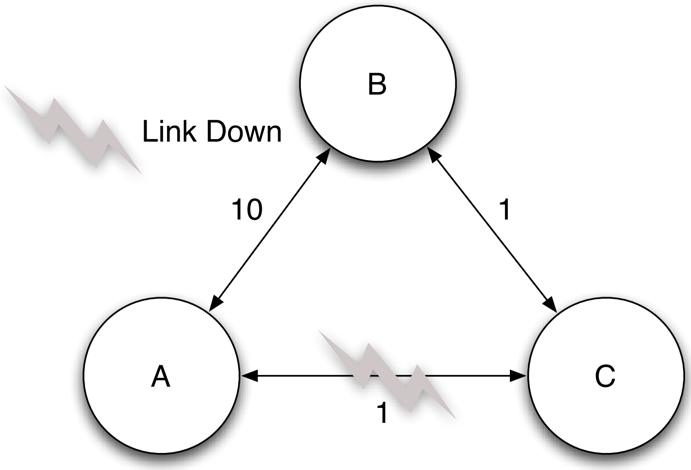

Figure 2.3: The *count to infinity problem* on a loop topology.

infinity if link $B - A$ was down.

Fortunately the *Count to Infinity* problem was reduced by means of the split horizon with reverse poison and the hold down techniques [eig09]. If a node sends his table of tuples (D_i, C_i) to a neighbour D_k, this technique marks all the message tuples that use neighbour D_k as an invalid entry. Despite the improvements the protocol still suffers from slow convergence time.

Link-State

Some years after using a Distance Vector protocol on the ARPANET, a new protocol class appeared - Link-State. This class gave birth to two intra-domain protocols: Open Shortest Path First (OSPF)[Moy98] and Intermediate System - Intermediate System (IS-IS)[Ora]. These protocols have as a main concern the maintenance of a consistent view of the network with a low convergence time, thus contrasting with DV's slow reaction to bad news, *i.e.* link failures.

Link-State's short convergence time owes greatly to the fact that each node floods periodically a list of the state of its links to its neighbours. These install the information and flood recursively the received message. To prevent over-flooding, a node checks if the received message has a recent sequence number, the message is dropped otherwise.

Since each node possesses a global view of the network, a shortest path tree is processed each time a topology change occurs, triggered by link-state message (or link failure/activation). This tree is computed with the Dijkstra algorithm [Dij59].

Despite being faster to converge than DV, it fails to scale well since each node must store the network's topology and compute the shortest path tree each time a new message is received, thus consuming more resources than its fellow DV [JI92]. This issue also brings another issue, more specifically route flapping, *i.e.* a link that changes its state constantly. Nodes have to notify each occurring flap [OBOM03].

2.3.2 Border Gateway Protocol (BGP): An Inter-Domain Protocol

Inter-domain routing is pretty similar to intra-domain routing but the basic element is now an AS instead of a router. The Border Gateway Protocol (BGP) [RLH06], currently on its fourth version, is the standard protocol for inter-domain routing.

The protocol's core behaves as a path vector algorithm exporting reachable paths of *visible* destinations. A router receiving a valid path, *i.e.* without loops, prepends his identification to the path and forwards the message to valid neighbours. Therefore, it is possible to avoid routing loops.

Two BGP routers need to establish a TCP session to exchange routes. There are two types of sessions: Internal Border Gateway Protocol (iBGP) and External Border Gateway Protocol (eBGP). The former is used to learn routes inside an AS, while the latter is used to learn routes between domains. Exchanged routes on these sessions are structured as a 3-tuple: the destination's IP-prefix, the path and the path's attributes. The path describes an ordered list of traversed ASes to reach the destination, while its attributes are used for routing decisions.

Once received a routing message, it is compared with a group of installed routes for the same prefix; a *best* path, is then selected from a set of rules. These rules, shown on table 2.1, are processed in an orderly way for tie-break decisions. Rules 1 and 3 use the attributes LOCAL_PREF and MED respectively.

#	Rule	Who defines the value?
1	Highest LOCAL_PREF attribute	Local Router
2	Lowest AS Path length	Neighbour
3	Lowest Multi Exit Discriminator (MED) attribute	Neighbour
4	eBGP over iBGP	Neither
5	Lowest Internal Gateway Protocol (IGP) cost	Local Router
6	Lowest router ID	Neither

Table 2.1: BGP's rules for tie-break decision about which route to choose

The LOCAL_PREF attribute, locally set at the BGP router, defines the preference of the path. ASes administrators can set this value to control outbound traffic. The MED attribute, set by neighbouring routers, defines how much an exported path should be discriminated; it is helpful to configure inbound traffic. However, the ability of the MED attribute could be overruled by the LOCAL_PREF attribute.

Once the route selection ends, if the received path is selected as the best path it will be used in a message exported to other routers. When exporting routes, routers might manipulate the path's attributes for various purposes. Three manipulation techniques are explained: AS-path prepend, route aggregation and the community attribute.

AS-path prepend consists on prepending an AS identifier one or several times to a path. As a result of this action, it *forces* an exported path to be less desirable, since BGP's decision process prefers shorter paths. However, using this method to control inbound traffic could be nullified by the LOCAL_PREF attribute.

Route aggregation is used to aggregate routes from different prefixes and announce a *generic* one. This technique helps to improve route scalability, since only one route is installed at receiving routers. For example, a provider with a /16 prefix that assigns a /24 prefix to a customer, could just advertise its prefix instead of both. At the level of routing decisions, BGP prefers to forward packets to prefixes that are as accurate as possible. This has consequences for multi-homed customers: They will receive traffic from non-aggregated routes, instead of the aggregated ones.

Regarding the community attribute, it is used to tag exported routes with a recognizable identifier. Upon the reception of a tagged message, routers apply a defined set of actions for the respective identifier. As an example, business relationships can be achieved with this attribute.

Figure 2.4 shows three ASes: A,B and C with border-routers $R1$, $R2$ and $R3$ respectively. Let us assume that AS C is a customer of both providers A and B. If $R1$ announces

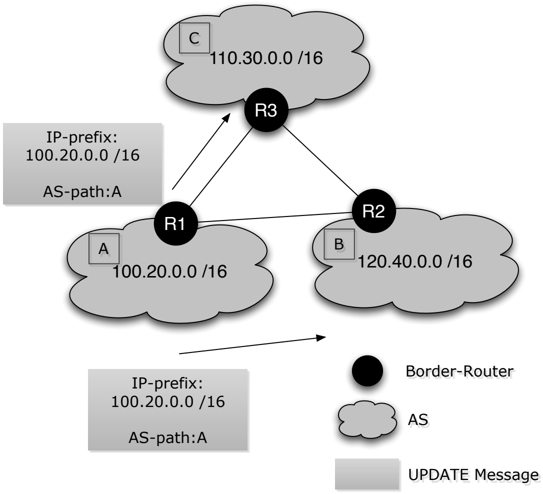

Figure 2.4: Applying business relationships with communities.

a route to A's prefix, $R3$ will tag this message with a community identifier to warn all intra-domain routers to only export this message to C's customers. As an advantage, policy violations can be avoided. However the use of community identifiers between ASes can lead to anomalous decisions, if misconfigured. In addition, *Donnet et al.* [DB08] also found that the use of this attribute has led to an increase on the number of routing entries.

Acquainted with BGP's route manipulation techniques, let us move further to BGP's route-flapping mechanisms: the Minimum Route Advertisement Interval (MRAI) and route-flap dampening. The MRAI [RLH06] functions as a *watchdog* for each prefix update, that only allows the announcement/retraction of updates after a minimum interval of time. If a given inter-domain prefix starts to flap, the MRAI timer will *hold* temporarily the announcement/retraction of all routes using this link, therefore improving route stability. However the MRAI might delay the announcement of important updates [GP01].

The route-flap dampening mechanism ignores routes that change too often, avoiding announcements from flapping routers [VCG98]. In the same *fashion* as the MRAI, it might also delay a network's convergence [ea02d].

After tackling BGP's basic features, along with their faults, let us remind ourselves that BGP was thought as a reachability protocol and has not changed as the network requirements became more stringent. Besides, its use extended to various purposes besides routing, such as traffic engineering.

Since the protocol is only concerned with reachability, a sensitive matter as Quality of Service (QoS) is hard to achieve. For example, route stability mechanisms might introduce *vast* packet loss and *jitter*, since these mechanisms delay a network's convergence [SKM09].

Besides route convergence, route scalability is also required. Given the fact that multihomed ASes might receive different routes for the same prefix, the number of routing entries increases considerably [YMBB05]. In addition a multi-homed AS cannot take advantage of multipath routing, since BGP only selects and exports the *best path* for forwarding purposes.

After recognizing today's problems on inter-domain routing, the next section introduces academical proposals that intend to solve intra-domain/inter-domain issues.

2.4 Academical Solutions

Section 2.3 gave a brief overview on today's routing protocols. This section presents a set of proposals that address their flaws. Nonetheless they are not perfect, thereby a summary is presented to point out what to expect from the future of Inter-domain routing.

2.4.1 Routing Independent Solutions

First let's begin with two routing independent solutions: Locator ID Separation Protocol (LISP) and G-ISP, both addressing distinct flaws.

Locator ID Separation Protocol (LISP)

On today's routing we use an unique numeration space to both identify and *locate* a network node. Separating both functions and relying on a distributed mapping service should alleviate scalability problems.

The Locator ID Separation Protocol (LISP) [FFO07a] attempts to separate both functions, through an IP-over-UDP tunnelling approach. It assumes that border-routers act as Routing Locators (RLOCs) between end-systems using IP addresses; these end-systems are named as Endpoint IDentifiers (EIDs). As expected on an Autonomous System, EIDs could have several RLOCs on their domain.

The basic concept of LISP is to tunnel packets through the Internet's core, from the RLOC of the source EID to the RLOC of the target EID. Packets that exit the source domain, are prepended with a LISP header at the Ingress Tunnel Router (ITR). Routing is therefore based on the target RLOC. Once these packets reach the Egress Tunnel Router (ETR) at the target domain, the LISP header is removed. This *paradigm* allows us to achieve better scalability, if only RLOCs are exported to the Internet's core, therefore reducing the number of BGP entries.

However, to route packets from the source EID to the destination EID, RLOCs are supposed to cache a mapping between both RLOCs and EIDs. In addition, cached mappings are stored temporarily. To improve this system, *Farinacci et al.* [FFO07b] attempted to separate the mapping service from the tunnelling service, by creating a distributed mapping service similar to DNS. Upon the occurrence of a missing cache entry, the RLOC will query a DNS-like server.

Iannone et al. [IB07] concluded that this service is both scalable and incremental. However such system implies storing and distributing the content to routers, thus we should expect non-negligible traffic from queries to locate an identifier. In addition, there is always a trade-off between a router's cache and the needed bandwidth to perform queries to a mapping server. Finding a balance between both is not a trivial matter. As a final remark, RLOCs and mapping servers need to be coherently updated, otherwise routing coherence between end-hosts is not guaranteed.

Global Internet Service Provider (G-ISP)

Global Internet Service Provider (G-ISP)[CS06] is a new concept that resembles an overlay network [ABKM01] built on top of the actual Internet. The main idea of the G-ISP, is to offer *reachability* to remote ASes that nearby networks cannot offer. With this rationale in mind, the G-ISP acts as an additional provider, with the intention of improving slow convergence and end-to-end Quality of Service at the inter-domain level. This concept also offers multicast support, but its use is not explored on this section.

An advantage of the G-ISP, is the *backwards* compatibility with the Border Gateway Protocol to receive/export routes. However, this paradigm only offers indirect connectivity through virtual links, contrasting with today's notion of direct connectivity. As a result, new protocol extensions must be added to BGP, since these virtual links might traverse more than one AS.

Upon the reception of a customer's/G-ISP's route, the G-ISP/customer prepends the route with the intermediate ASes that form the virtual link. Otherwise, routing loops can happen. To alleviate a network's convergence time, the G-ISP model only exports short AS-paths; according to the authors of the model, the longer the AS-path, the longer the convergence time.

The G-ISP model assumes that the number of ASes between the G-ISP and the G-ISP's clients is small. To fulfil this objective, the authors expect to build a network that only covers ASes whose distance between any pair source-destination is $2r + 1$, with $r \epsilon \{1, 2\}$ $r = 1$ or $r = 2$.

The G-ISP's Quality of Service (QoS) would benefit from the previous assumption if bilateral agreements are established between the G-ISP and its clients. However, neighbouring ASes from the G-ISP might not be interested to establish bilateral agreements. Intra-domain techniques could still be employed at the G-ISP for better QoS [NBBB98]; but the G-ISP model relies on the BGP protocol that lacks support for QoS.

Regarding virtual link failures, the model does not refer how to solve these failures. It should offer redundancy measures to solve these situations. In addition, the use of BGP's independent policies might provoke anomalous routing and thereby fail to offer any improvements.

2.4.2 Intra-domain protocol extensions

Putting aside the last topic, the following solutions focus on improvements for Intra-domain routing, though their rationale can be applied to Inter-domain routing. Both XL[LVPS08] and FCP[LCR$^+$07] can be thought of as extensions to reduce or suppress updates that diminish the convergence time of a network.

Approximate Link-State (XL)

XL[LVPS08] is a link-state protocol whose rationale can be applied to any standard link-state protocol like OSPF. The main concern of this protocol is to diffuse updates to its neighbours selectively, as long it complies to any of these rules by order:

1. The update is used to announce a cost increase (link failure);

2. The neighbour is used on the protocol's shortest path tree;

3. The cost to reach a given destination has improved by a factor of $1 + \epsilon$

If an update does not comply with any of these rules then the protocol suppresses it. The authors show that it diminished the number of updates, thus reducing the network's convergence time. Suppressing updates has the disadvantage of using suboptimal routes. In addition, it does not support multipath routing, therefore rendering the protocol useless if multiple paths are fit to be used.

Failure Carrying Protocol (FCP)

With a different rationale, FCP[LCR$^+$07] can be used as a *watchdog* to either a link-state protocol or BGP, suppressing all updates. The protocol assumes that each node has a consistent map of the network reliably flooded and distributed by a coordinator or several replicated ones [CCF$^+$05].

The protocol relies on tagging data packets with failed links at each traversed router. This way it is possible to avoid loops with a null convergence time, as long as all nodes share the same perspective of the network. Without a consistent view of the network, the authors recommend the use of source routing. If the source path has invalid links, a new path from the origin is recomputed.

We can also extend BGP with FCP, so any policy violations from source-routing are *corrected* by marking them as invalid links. Nonetheless, this *watchdog* is defeated for not storing any of the link failures temporarily, decreasing its robustness for consequent routing decisions.

The option of suppressing the convergence time has its appeal, but *tagging* routes with link failure information increases the packets overhead. As a solution, the authors of the protocol suggest the use of known labels replacing the links. To use these labels, some standardization should be applied at the inter-domain level to avoid routing loops. Furthermore, the protocol subverts itself for using suboptimal routes, in the same manner as XL.

2.4.3 New Inter-domain protocols

At the inter-domain level, there are two major contributions: Hybrid Link-State Protocol[SCE+05] and New Inter-domain Routing Architecture[YCB07]. Both aim to solve scalability and convergence problems based on the hypothesis that the Internet follows a hierarchical structure.

Hybrid Link-State Protocol (HLP)

HLP assumes a hierarchical structure of Provider to Customer relationships rooted at each *tier-1* provider. Each provider forms its own hierarchy, thus a multi-homed AS belongs to multiple hierarchies.

The protocol routes packets based on AS identifiers, improving routing table scalability. It also differs from BGP since policy relationships are explicitly published and forms a hierarchy. Topological changes inside a hierarchy are reported using link-state messages and between peers of different hierarchies it uses Fragmented Path-Vector (FPV) messages to report them.

FPV messages contain the following doublet (P_i, C_i), with P_i being the path and C_i the cost to reach destination i. P_i shows an ordered list of *hierarchical border* nodes that reach the destination AS. Upon the reception of a FPV message, its content is updated and a new message is flooded at each node of the hierarchy.

The system also uses another mechanism for better convergence: *cost-hiding*. *Cost-hiding* suppresses route announcements whose cost does not surpass a default value of Δ.

Despite *complying* scalability and convergence requirements to hierarchies, it fails to cope with the current Internet's business model. Section 2.2.2 revolved on this issue with recent studies showing that the Internet presents a scale-free nature opposed to the idea of a hierarchy.

Besides the previous facts, multi-homed ASes have to process the link-state algorithm for each hierarchy, putting pressure on the protocol's scalability.

New Inter-domain Routing Architecture (NIRA)

The NIRA protocol, just as HLP, assumes a hierarchical routing structure. Each hierarchy has a tier-1 provider that belongs to the *Core region*. Each of these providers assigns recursively IPv6 prefixes to their customers. This hierarchy is labeled as a customer's *access network*, or simply an *up-graph*. Opposite to the *Core*, P2P relationships can have *non-core* visible addresses and assign them recursively to their customers.

To disseminate routing information, NIRA uses a routing protocol named Topology Information Propagation Protocol (TIPP). TIPP has two components : A path-vector component that diffuses provider-level routes and a link-state component used to control topological changes inside a hierarchy.

For scalability and convergence sake a domain may configure TIPP to prohibit the dissemination of routing messages between domains, and the protocol only forwards link-state messages between transit domains.

So far NIRA's concept is similar to HLP. However it allows a user to choose the traversed routes for its packets, constricting the user to his set of providers. This way if a user sends packets to a destination, these will be forwarded based on the user's and destination's address in a hierarchical sense: first upwards on the user's *up-graph* and then downwards on the destination's *up-graph*.

NIRA supports multipath. When a user that wishes to use alternative routes, he can query a Name-to-Route Lookup Service (NRLS) server that works similarly to a DNS server, thus retrieving the remaining destination's addresses. Despite letting a user choose its own providers, NIRA fails to address the current Internet's Business Model, in the same manner as HLP. Taking into account the multi-homing reality of the Internet, the NIRA model is more capable of handling hierarchies than HLP, since it allows a user to choose its own set of hierarchies.

2.5 Summary

Table 2.5 summarizes the pros and cons of the previously discussed protocols. Based on the findings from recent studies of the Internet and the latest trends of the studied protocols, it is possible to draw some insights on what should be the future of Inter-domain routing.

First it is essential to build a system that scales well, HLP and NIRA took a first step on this direction creating separate regions, though the assumption of the Internet structured as a hierarchy is not absolutely true.

G-ISP on the other hand tries something similar to a concept close to an overlay network to reduce the network's convergence time. However, it still relies on BGP that features other issues besides this one.

Routing correctness is also an important matter. FCP brought the idea of distributing a network map to all nodes of a network ensuring routing correctness at the inter-domain level. Despite this advantage not all domains might want to publish their relationships.

Another incisive matter is the current address system that does not discern the identity of a node from its location. A service like ID-Mapping (from the LISP architecture) should be valuable for the future since its idea allows a protocol to locate a node based on its identification (Routing *vs.* Reachability).

To end this summary some of the previous proposals suppress routing announcements based on a route's cost, thereby improving the network's convergence time. Although they don't ensure routing correctness.

This thesis intends to merge some of these views and improve the current state of the art, which move us further to the next chapter.

Protocol	Pros	Cons	Class
DV	• Scalable	• Count-to-Infinity problem	Intra-domain
Link-State	• Small convergence time	• Not Scalable	Intra-domain
BGP	• Policies Expression through the use of attributes • Flexibility	• Uncoordinated policies • Great convergence time • Not Scalable[1] • Lack of QOS	Inter-domain
LISP	• Improves Internet Scalability	• Non negligible traffic • Storage and Distribution	Independent
G-ISP	• Improves convergence time • End-to-end QoS	• Routing correctness of BGP	Inter-domain and Independent
XL	• Improves convergence time	• Usage of suboptimal routes	Intra-domain
FCP	• Suppresses the convergence time	• Usage of suboptimal routes	Intra-domain
HLP	• Improves convergence time and scalability • Explicit use of policies	• Based on the assumption of a hierarchical structure • More than one shortest path tree for multihomed ASes	Inter-domain
NIRA	• Improves convergence time and scalability	• Based on the assumption of a hierarchical structure	Inter-domain

Table 2.2: Protocols Summary

[1] Under the following circumstances: multihomed ASes, multipath routing and without Classless Inter Domain Routing (CIDR)

Chapter 3

An approach to multi-region routing

3.1 Introduction

The current chapter presents a solution for today's inter-domain routing. In the last chapter, we realized how complex is the Border Gateway Protocol, despite its flexibility; for example, attribute manipulation might turn into a paradigm between flexibility and complexity. Autonomous System administrators manipulate attributes with the purpose of altering routing mechanisms, but also for other matters, such as the definition of prefix-based Virtual Private Networks (VPNs). This disparity of functionalities might turn the system poorly adapted to topology changes. In addition, BGP does not take full advantage of multipath routing, or even of a multihomed scenario.

Presenting a new solution that maintains BGP's flexibility and features, is not straightforward. To fulfil these objectives, we need to separate functionalities without clinging to an overused feature such as attribute manipulation.

Amaral et al.'s research [ABP08] proposed a new architecture for Inter-domain routing, Dynamic Topological Information Architecture (DTIA), that provides us a simple reachability protocol. On this architecture, routers build paths based on a static map of the network and co-operate to learn link failures. As an advantage over BGP, functionalities such as routing and traffic engineering can be implemented on top of it. This thesis extends *Amaral et al.*'s work with the intention of improving routing scalability and convergence, through multi-region routing.

First it is presented an overview of DTIA's rationale and afterwards an extension to multi-region routing.

3.2 Dynamic Topological Information Architecture (DTIA) - Basic Rationale

The DTIA protocol introduces a new approach to inter-domain routing that emphasizes the modularity of different concerns: routing, reachability, naming and addressing. However other modules can be built on top of it, such as traffic engineering. As aforementioned on section 3.1, this separation of features is an advantage over the Border Gateway Protocol. However, to replace BGP's functionalities with attribute manipulation, DTIA uses a set of rules to validate and differentiate paths. Since the system handles paths differently from BGP, *i.e.* in a modular manner, it shows great adaptation to multipath and multihoming.

3.2.1 Model Assumptions

The protocol's model makes three distinct assumptions:

1. The maintenance of the current business model based on Autonomous Systems and Internet Service Providers;

2. The acknowledgement of a hierarchical structure based on customer-provider relationships, nevertheless augmented with Peer to Peer relationships at the same(and different) level(s) of the hierarchy;

3. Inter-domain links that form the Internet are stable over time.

The first assumption is made under the judgement that current Autonomous System administrators will not practice a different business model in the near future; however, the protocol is receptive to incremental changes.

In agreement with the last chapter, the second assumption [AGA^{+}09] is justified from *Gao*'s work[Gao01] that identified an hierarchical structure based on customer-provider relationships. However, recent data from CAIDA[cai09b] shows an enrichment of connections over the years that blurs the idea of a pure hierarchical structure. Three tendencies have been currently identified from CAIDA's figures: direct links bypassing tiers; peer relationships between ASes that exchange large amounts of traffic; and regional cliques at *middle* levels of the hierarchy. As concluded from last chapter, HLP and NIRA are not fitted, given the actual tendencies. In the same manner, BGP cannot take advantage of such network in terms of multipath and backup purposes, since the protocol just exports the *best* path and backup links are tuned according to the topology.

The final assumption is justified by the fact that inter-domain links are based on business relationships. Topological changes occur in a controlled manner. For real-time purposes it does not matter if a link exists; in a time sensitive case it is more important to know whether a link failed or not. Moreover, intra-AS failures are more probable than inter-AS failures, reinforcing the idea that inter-domain links are stable[1].

3.2.2 Design Choices

The model's assumptions have led to the following design choices:

1. Reachability is based on AS connections and not on prefixes;
2. A set of rules replaces prefix manipulation;
3. Routers get a static map of the network and co-operate to learn about failures;
4. Maps and co-operations are limited to regions.

The first design choice makes sense considering the size of BGP's routing tables. We have realized that BGP cannot remain scalable, if it still builds prefix-paths for each physical link. As a solution, prefix-aggregation could be used but we have seen that it does not work well with the use of multihoming, since a multihomed AS might receive more traffic from non-aggregating prefixes. The use of Autonomous System connections would be a viable option to reduce the size of routing tables, given the number of ASes. As a consequence, this alternative is better in terms of scalability, since it builds paths based on AS numbers instead of prefixes. This decision has diverging opinions, some in favour [SCE+05], others against it [Bon07]. However, it brings two problems: First, packets can follow multiple paths with different transit times, which has an impact to the congestion control mechanism of TCP (The calculus of the Round Trip Time becomes more complex

[1] However, Intra-AS failures may lead to Inter-AS failures

and the reaction of TCP when it receives unordered packets must be changed). Second, a mapping between ASes and prefixes must exist.

We should assume the existence of a mapping service between ASes and prefixes, that supports host multihoming. Moreover, it should also support mobility in terms of prefix assignment to ASes, to deal with mobility demands in military networks.

With respect to the second choice, we have mentioned how unmanageable BGP has become with prefix manipulation, since it covers many purposes besides routing. Their implementation requires a high degree of coordination between AS representatives, implying the knowledge of a precise topology. Upon the occurrence of a link failure the system could become unpredictable when tuned to an exact topology. As an alternative, we should define a closed system that captures the essential features of inter-domain routing.

For the third choice, we have already assumed that inter-domain links are stable due the existence of business relationships. Moreover, ASes connect securely inside a protected room. An inter-domain protocol should only concern with the dynamic part of the network and not its discovery. Contrasting with BGP, the algorithm for the dynamic part should be light and the general algorithm should enable traffic engineering characteristics. BGP on the other hand, relies heavily on mechanisms for network discovery and network management.

DTIA assumes that a central entity(or various replicated) delivers a static map of the network to routers. It is not guaranteed that the map is the actual state of the network, because of failures, but all routers know the same information and act dynamically upon it. This approach was somehow followed by FCP. As a difference from traditional routing protocols, there is no need to discover the network's graph and the protocol is simplified in terms of exchanged messages. The main concerns are to warn routers about failures, re-route packets that encounter a failure, and warn routers when a failure is solved. In addition, only relevant routers are warned about failures according to precise rules.

The fourth choice deals directly with the subject of this thesis. We know that global events in BGP are related to withdrawal and announcement of prefixes. These events are confined to regions depending on their position at the *hierarchy*. However regions in BGP are difficult to define, since the protocol builds several graphs over a set of ASes.

HLP attempted to define regions, based on a hierarchy of Customer to Provider relationships separated by one-hop Peer to Peer relationships. However, the use of heavy multihoming at middle-levels of the hierarchy makes the protocol rather complex; with this rationale, ASes routers might belong to several link-state trees.

DTIA defines regions as a set of ASes with a few restrictions. At each region, a graph of the network is built and delivered to routers. As an example, RIPE has a database that describes policy relationships between ASes [NCC09]. Its content could be used for an *European* region.

The Internet is divided in multiple regions. Packets going from one region to another, either use a direct link from the remote destination(if valid), or they climb up the hierarchy and go down in the destination region. Regardless of the size, regions always include tier-1 ASes. The definition of region will be thoroughly explained on section 3.4.

3.3 Architecture

This section explains DTIA's architecture. The protocol stands out from BGP, since it separates different concerns in a layered approach. Traditionally, a routing algorithm performs under two distinct operations:

1. Policy - It defines the link characteristics, such as the link's metrics or attributes;
2. Mechanism - It determines how the network graph is discovered and defines the route selection algorithm.

BGP it is entirely based on prefix policy, which means that the policy component has direct consequences on the network discovery for each prefix. Opposite to BGP, DTIA separates functions in three layers: Reachability, Routing and Traffic Engineering. The first two handle most of BGP's characteristics; the third deals with some remaining BGP characteristics and can handle more features. It is not handled on this thesis and is currently under research at the Telecommunications Group at FCT-UNL.

At the first layer, the protocol handles the path computation from AS X to all destinations and regions; the obtained set of paths is defined as $P_r(X)$. Each path follows a *valley-free philosophy* [SARK02], where packets received from a provider are not forwarded to another provider. The protocol computes several paths for the same destination, providing a basis for multipath routing.

Building *valley-free* paths implies the use of the *common policies* referred on section 2.1.2. These policies are enough to deal with 99% of AS relationships, but DTIA extends them further with two extra relationships. The new relationships are used for sibling and backup purposes. DTIA assumes that all ASes use the same set of rules and link labels to build paths; on the opposite side, not all BGP routers apply consistent rules(meaning the use of attributes) on advertised routes.

Different Routing algorithms can be applied on $P_r(X)$ to perform routing operations. A multipath algorithm is defined on DTIA that calculates $R_r(X)$, a subset of $P_r(X)$. The information from $R_r(X)$ could be further used to apply load balancing and(or) traffic engineering algorithms.

Sections 3.3.1 and 3.3.2 explain thoroughly the reachability and routing modules of DTIA.

3.3.1 Reachability

On section 3.2.2, it was assumed that a central entity would deliver a graph of the region to ASes. Each time a new graph is generated, a sequence number of the graph is incremented. This graph $G(V, A)$ is structured as a directed graph; where $V(G)$ are the vertices that model ASes and A are the arcs representing the link between ASes. DTIA assumes the following relationships for a link:

1. Provider-Customer - The provider accepts all traffic from the client. Two arcs are considered: one in the Provider to Customer ($p2c$) direction and another on the Customer to Provider ($c2p$) direction;

2. Peer to Peer ($p2p$) - ASes provide connectivity for their direct or indirect customers. However, no transit traffic is allowed from the peer. The protocol assumes an arc in each direction;

3. Peer to Peer allowing backup (*p2pbkp*) - The same as before, except that transit traffic is only accepted if no other path exists. Once more there is one arc in each direction;

4. Peer to Peer allowing transit traffic (*p2patt*) - Transit traffic is allowed in any situation. One arc in each direction.

According to these relationships, the protocol defines two tables of rules for path validation. DTIA explores all ascending (*c2p*), descending (*p2c*) and horizontal (*p2patt*, *p2pbkp* and *p2p*) paths *hop-by-hop* in the forward direction. To avoid valley paths, an attribute *Direction* (D) is added to the path. The path's direction is set according to the first link's relationship:

1. If *c2p* then D is 1;

2. If *p2c* then D is 0;

3. If *p2pbkp* or *p2patt* then we have two paths, one with $D = 0$ and $D = 1$;

4. If *p2p* then D is 0.

When exploring paths, we should be also preoccupied with valid paths and with loops. Loops can happen if the links are Peer to Peer. As a solution, when a path reaches an AS, the AS number should be verified for loops at the given path. If the AS does not exist on the path, then its number is appended to the path.
The attribute D of the path can change while exploring. However, a descending path cannot change to an ascending path, according to the *valley-free philosophy*. Once an ascending path finds the first *p2c* or *p2p* link, it changes to a descending path.

Tables 3.1 and 3.2 show the validity rules for ascending and descending paths respectively; V marks the path as valid and X as invalid. As observed on table 3.1, an ascending path changes its direction D once it reaches a *p2c* link. Besides a descending path with departing *p2p* or *c2p* arcs is always invalid, as observed on table 3.2. Figure 3.1 shows a peculiar case with horizontal paths. AS A is connected to B and C using a *p2pbkp* relationship. Two paths are valid from A with directions $D = 0$ and $D = 1$ respectively. For the ascending case, it is possible to reach AS D; although for the descending case it is not possible. Considering a flow transmission between ASes E and D, the path E-A-D is used. If a failure occurs between ASes D and E, the path E-A-B-C-D should be used, since a backup path is only used if no other path exists.

		Next Link				
		p2c	*c2p*	*p2pbkp*	*p2p*	*p2patt*
Previous Link	*p2c*	-	-	-	-	-
	c2p	V; D=0	V	V	V	V
	p2pbkp	V; D=0	V	if(AS in set) X; else V	X	if(AS in set) X; else V
	p2p	V; D=0	X	X	X	X
	p2patt	V; D=0	V	if(AS in set) X; else V	X	if(AS in set) X; else V

Table 3.1: Path Validation for $D = 1$

		Next Link				
		p2c	*c2p*	*p2pbkp*	*p2p*	*p2patt*
Previous Link	*p2c*	V	X	V	X	V
	c2p	-	-	-	-	-
	p2pbkp	V	X	if(AS in set) X; else V	X	if(AS in set) X; else V
	p2p	V	X	X	X	X
	p2patt	V	X	if(AS in set) X; else V	X	if(AS in set) X; else V

Table 3.2: Path Validation for $D = 0$

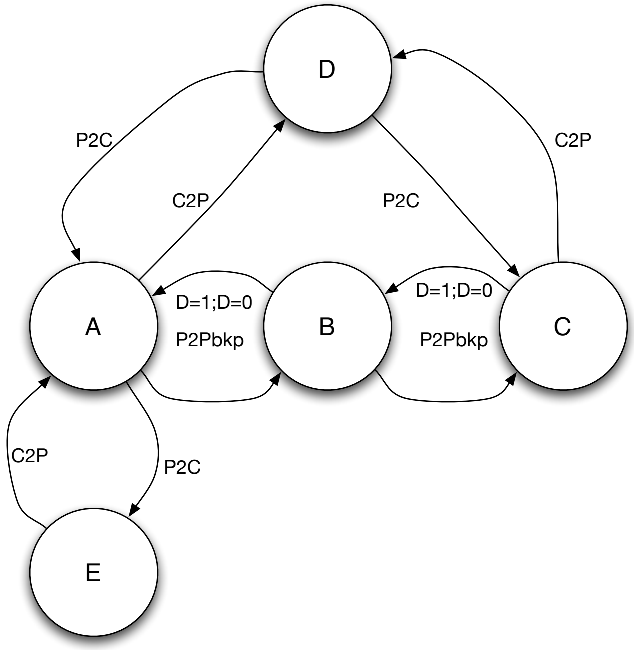

Figure 3.1: Path exploration with backup links

After obtaining the set of paths of $P_r(X)$, the routing algorithm will decide which paths to use to each destination. The authors of the protocol prove that the explored paths of $P_r(X)$ have no loops and state the following theorem[ABP08].

Theorem 1. *Assuming that there are no cycles in the provider customer relationships,* i.e. *no domain is a provider of one of its direct or indirect providers (peers are also indirect providers). A valid path between two ASes has no loops.*

Theorem 1 assures us that all paths are valid as long as they comply with the policy rules from tables 3.1 and 3.2. Otherwise the system would be unstable if all ASes had *conflicting* paths that do not respect the aforementioned rules.

3.3.2 Routing

This section explains thoroughly the routing layer from *Amaral et al.*'s work [ABP09]. The fact that $P_r(X)$ contains loop-free paths does not mean that the entire system is loop free. Considering that we are in control of a multipath system, some paths may enter in conflict with others and form a loop; even if this aspect is well controlled, the occurrence of a link failure might provoke conflicts of the same order.

To solve these aspects of routing correctness, a ranking system was defined to classify valid paths and a management algorithm was proposed to solve failures. The ranking mechanism works on a discrete-space that attributes a cost to paths. Paths that have the same rank are used equally, providing a basis for multipath routing.

The ranking mechanism is based on four preference rules:

1. No traffic is forwarded from one provider or peer to another provider or peer;

2. Customer routes are preferred over peer or provider routes;

3. Primary paths are always preferred to backup paths;

4. From primary paths, Peer to Peer allowing transit traffic and Provider to Customer paths (both have the same preference) are preferred over Peer to Peer paths. Customer to Provider paths have the worst preference.

The first two rules are usually applied in the Internet and the last two were added because of the new policy relationships. With these rules we should expect two effects: First, some paths might not be selected. Second, all selected paths are ranked. It is proved that if each AS uses paths with the highest ranking for routing, then the protocol will converge and packets will reach the destination AS without forming routing loops.

DTIA acts as a Path Vector protocol in the same manner as BGP, since it chooses routes according to their attributes and established preference. According to *Griffin et al.*, DTIA works likewise a Local Simulated Path Vector (LSPV)[GS05].

Routing Correctness

A routing protocol is correct if in a stable network (without changes) it obtains a set of loop free paths between every pair of nodes that have connectivity according to the rules for the link. DTIA's routing correctness is proved using the concept of routing algebra from *Sobrinho* [Sob05]. The algebraic property to ensure routing correctness is valid for Path Vector protocols [GS05], and DTIA can be seen as a Path Vector protocol.

A routing algebra is defined by a tuple $A = (\Sigma, \prec, \oplus L, \phi)$. Σ is a set of signatures that qualify paths; $\prec$ is a preference relationship between signatures (*e.g.* $\alpha \prec \beta$; α is preferred); L is the set of labels associated to links; $\oplus$ is a binary operation used to obtain path signatures; ϕ is a special signature that models invalid paths.

For this protocol, we have the following set of labels $L = \{p2patt, p2c, p2p, c2p, p2pbkp\}$ and signatures $\Sigma = \{\varepsilon, P2Patt, P2C, P2P, C2P, P2Pbkp\} \cup \{BKP \times N^{+}\}$. The ε signature is the initial path signature when there is only the node at the end of the path; the remaining signatures are similar to link/label types. Each AS uses $P_r(X)$ and table 3.3 to perform the calculations of the path signatures using the operation $\oplus$. The procedure is the following a link of label type l will be appended in the direction of the source node to a path of signature α, resulting in a new path signature $\beta = l \oplus \alpha$. Looking at table 3.3, if we append a *c2p* link to a $P2P$ path, then the resulting signature of the new path will be $C2P$; this means that the resulting signature is an ascending path to a provider followed by a $P2P$ path signature. Table 3.4 shows the ranking order by which paths are chosen; the higher the more preferred.

		Signature						
		ε	$P2Patt$	$P2C$	$P2P$	$P2Pbkp$	$C2P$	(BKP, y)
Label	*p2patt*	$P2Patt$	$P2Patt$	$P2C$	ϕ	$(BKP, 1)$	$C2P$	$(BKP, y+1)$
	p2c	$P2C$	$P2C$	$P2C$	ϕ	$(BKP, 1)$	ϕ	$(BKP, y+1)$
	p2p	$P2P$	$P2P$	$P2P$	ϕ	$(BKP, 1)$	ϕ	ϕ
	c2p	$C2P$	$C2P$	$C2P$	$C2P$	$C2P$	$C2P$	$(BKP, y+1)$
	p2pbkp	$P2Pbkp$	$(BKP, 1)$	$P2Pbkp$	ϕ	$(BKP, 1)$	$(BKP, 1)$	$(BKP, y+1)$

Table 3.3: $\oplus$ binary operation, at the leftmost column we have the appending link and at the upmost row we have the path's signature

ε
P2C = P2Patt
P2P = P2Pbkp
C2P
(BKP,1)
...
(BKP,n)

Table 3.4: Ranking Order

Observing carefully table 3.3, we notice that a *p2pbkp* link could be used either as a regular Peer to Peer link or as a backup link. In terms of preference, a *P2Pbkp* signature(reflecting its usage as a regular Peer to Peer (P2P)) has the same effect as a *P2P* signature. When the link is used as a backup, the resulting signature is (BKP, y). y is incremented each time a new link is appended. From this type of link, we can extract two examples:

1. Backup links used as normal peering: the resulting signature of *p2pbkp* $\oplus$ *P2C* is *P2Pbkp*. Prepending a *p2pbkp* link to a customer's path is equivalent to a *P2P* path, since we have a normal peering relationship.

2. Backup links used as backup: for backup paths the resulting signature is always (BKP, y). The value of y increases every time a *p2pbkp* link is used. For every new link in a backup path, the y integer is increased. Using a path with signature *p2c* $\oplus$ (BKP, y) is possible. However it decreases the path preference, since we are attaching a provider's link to a backup path.

Acquainted with the protocol's algebra, we need to understand the definitions of cycle and monotonicity before outlining the protocol's routing correctness. A cycle is a sequence of distinct nodes, except for the first and the last. A cycle is free, if at least one of its nodes forwards packets to the destination node out of the cycle; *i.e.* an outer path whose preference is higher than the next node belonging to the cycle.

Let us consider that each node i of the cycle has a signature α_i. But node i has other j paths to the destination that do not follow the cycle with signatures β_{ij}. If $S(x_i, x_{i-1}, x_{i-2})$ is the signature of the path $x_i\ x_{i-1}\ x_{i-2}$; the condition for a cycle to be free is:
Freeness of cycles: a cycle $x_1,\ x_2, ...,\ x_{n-1},\ x_n$ with $x_n = x_1$ is free, if there is an index i, $2 \leq i < n$ such that $\beta_{ij} \prec S(x_{i+1},\ x_{i+2}, ...,\ x_n)$.

From the previous definition, a cycle L is always free as long as there are paths at each node with a higher preference than using L. This definition leads us further to the following property:

Network freeness: A network is free, as long as all cycles are free.

Monotonicity is also an important property. An algebra is monotone, if the preference of the path does not increase when a link is prepended to the path.

Monotonicity of an algebra: An algebra is monotone if for all $\alpha \epsilon \Sigma$, and for all $l \epsilon L$, $\alpha \preceq l \oplus \alpha$. $\preceq$ is a preference relationship that denotes a possible monotonicity between two signatures.

We also have a stronger property than monotonicity: Strict monotonicity. This property ensures that adding a label to a path, decreases the preference of the resulting signature. From these properties, *Sobrinho*'s work proves the following theorems[Sob05]:

Theorem 2. *In a free network, a path-vector protocol converges to local optimal in-trees.*

Theorem 3. *If a path-vector protocol has a monotone algebra, then the protocol can converge to local in-trees, regardless of the network.*

Theorems 2 and 3 are important to prove that path-vector protocols converge. From these theorems, the authors of DTIA prove the following theorem [ABP09]:

Theorem 4. *Assuming that the network has no cycles in the provider-customer relationships, then DTIA's routing protocol converges using sets of cycle free paths.*

Theorem 4 is an important property since it proves that DTIA is a *stable* protocol; all ASes converge with the same set of cycle free paths. However we should be careful since with only *p*2*patt* links paths can form non-free cycles: each appended *p*2*patt* link does not decrease the preference of the path. From this fact we can conclude that the protocol is simply monotone. To tie-break $P2Patt$ paths, the authors suggest a simple solution that chooses the $P2Patt$ path with the least number of links, or with the least number of *p*2*patt* link labels.

If various paths have the same number of links, they all can be chosen as DTIA allows multipath routing. For example two $P2C$ paths with the shortest $P2Patt$ path simultaneously.

Routing Implementation - Forward Direction

The path signature calculation is a complex procedure. Calculations start at the end of the destinations and end at the source node. Applying this rationale in the forward direction is impossible, since policy violations are undetectable. Fortunately the distinction between reachability and routing makes $P_r(X)$ to contain valid paths and the use of DTIA's algebra allows the classification of paths on a kind of forward direction (with simple changes).

If we exclude the *p2patt* and *p2pbkp* labels, we have the same correspondence with the common policies of the current Internet business model. We observe that any path with a link extended to the origin's direction, results in a signature that equals the link or an invalid signature ϕ. This way, the signature of the path is defined by the last link. As links are appended to the path the order of preference maintains or decreases since the algebra is monotone. An appended link that raises the order of preference of the path, is a policy violation.

A simple algorithm that *walks* in the forward direction can be defined: let us consider S_i as the signature of a single link l_i, where $S_i = l_i \oplus \varepsilon$. Following the forward direction, we calculate the signature of each pair of ASes in the path. The resulting signature to a destination AS n is the least preferred signature from all S_i. The preference of the signature decreases monotonically according to the result of $\oplus$ with the label of the appended link. Consequently, the total path signature is defined by the link whose signature has the lowest rank.

Contrary to BGP, this process is possible since DTIA is aware of path violations; it calculates signatures of one-link only paths that are valid. Since $P_r(X)$ only has valid paths, it poses no problem to search paths on the forward direction; therefore the complexity is reduced.

Considering all labels from table 3.3, the rationale maintains. However, with *p2pbkp* links the calculation of each signature S_i is not so simple. A *p2pbkp* link could be used as a normal peering link or as a backup. To obtain the current signature S_i at link i, we should take into account the previous link $i-1$. The signature S_i at link i should be $S_i = l_{i-1} \oplus (l_i \oplus \varepsilon)$ with l_i as the current link label and l_{i-1} as link $i-1$ label. The path's signature for the first link ($i = 1$) is $S_1 = l_1 \oplus \varepsilon$.

The final signature of the path at link i is the least preferred signature from comparing signature S_i with the path's signature at link $i-1$. If the path's signature at $i-1$ is (BKP, y) with $y \geq 1$, then the resulting signature at link i is $(BKP, y+1)$. A (BKP, y)

path cannot raise or maintain its preference for each appended link. Afterwards, the path signature should be recorded at link i.

Figure 3.2 exemplifies the calculation of the path signature from source node X to node D. After the first *p2pbkp* link, the signature maintains the $C2P$ signature, since we are using AS B as a normal peer. However after the second *p2pbkp* link the resulting path signature is $(BKP, 1)$, since the signature considers the backup links A-B and B-C.

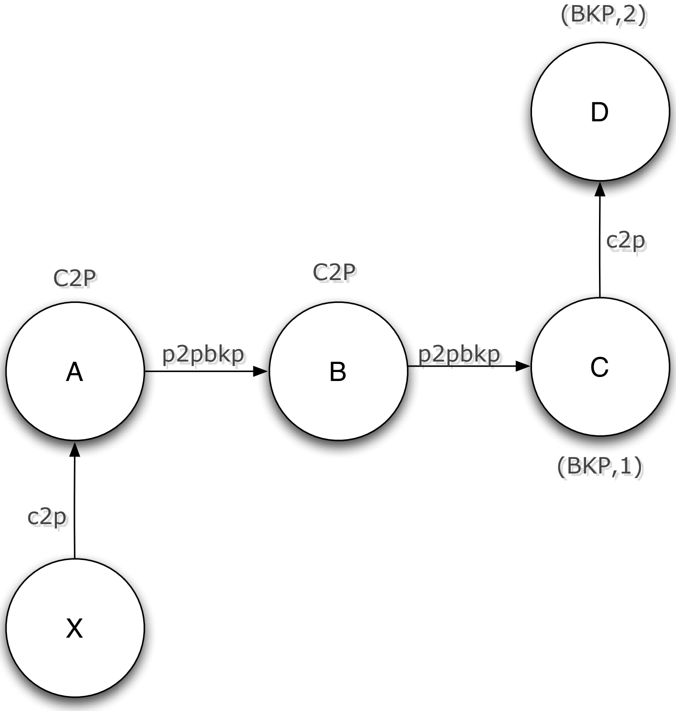

Figure 3.2: Signature calculation by walking in the forward direction.

3.3.3 Failure Management

The static graph does not guarantee that it represents the latest state of the network. The dynamic part of the protocol checks the reachability and routing layers for link failures, ensuring that routing loops are not formed during link failures; it should also restrain packet loss if at least a failure free path exists.

Once a link fails, routers disseminate control packets at reachability or routing layers. Control packets contain the link identification, its direction and the sequence number of the graph. When a control packet arrives, the AS checks if all previously reachable ASes are still reachable without using the failed link. If at least one AS becomes unreachable, the dissemination of the control packet continues; the dissemination follows the rules of tables 3.2 and 3.1. Otherwise, if all ASes remain reachable, the dissemination stops. Further evaluation is performed at the routing level.

Link failures at the routing layer might alter the paths signatures to some ASes. Routing loops might occur if a new path belongs to a lower rank than the path previously used. Let us observe figure 3.3. If link A-B fails, according to the reachability layer both A and B can reach each other; no control packets are sent at this layer. At the routing level the signature from AS B to A before the failure is $P2C$. After the failure the signature alters to $(BKP, 1)$ (B-C-D-A path). Since B is aware of the failure, it will route packets through C; however, C is not aware of this failure and prefers to use B as the next hop to route packets; since C-B-A is a $P2Pbkp$ path it has a higher preference than C-D-A whose signature is $C2P$. In this situation we have a loop between B and C.

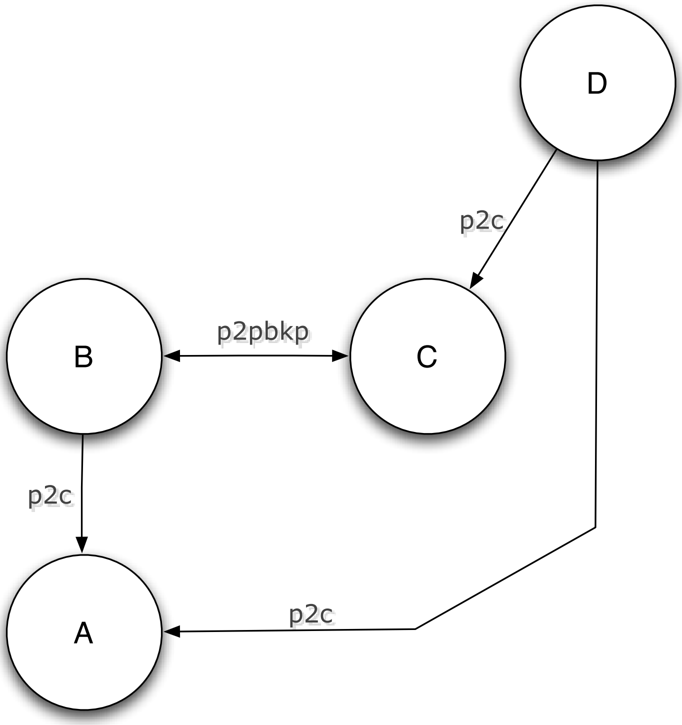

Figure 3.3: Network example of a routing loop if link A-B fails.

To assure routing correctness, AS C needs to be warned of the link failure A-B. Since AS B changed its selected path to a lower ranked one, this AS should notify AS C in order to assure routing correctness. Once AS C is notified, it will use this new information to route subsequent packets.

In general terms, the conditions for dissemination are the following: if a path has its signature changed then control packets are disseminated according to the rules. The control packets contain the identification of all failed links that the sender AS knows about. Upon the reception of a control packet, an AS only identifies links that it is not aware of their failure, otherwise it discards the packet. If all paths keep the preference signature then the dissemination stops.

Over time the graphs that ASes have might not be the same, since notifications are only disseminated when a link failure affects an AS. This is a powerful feature as a contention mechanism. Control packets are not sent to ASes whose valid paths are not affected by the link failure. If a link comes up again, the dissemination criteria is the same. For example an unreachable AS becomes reachable or a path with a higher preference is now used.

The dissemination's scope is directly related with the degree of multi-homing of the region. A high degree of multi-homing, makes the disseminating region smaller. It is less probable that an AS looses reachability since there might be more paths with the same order of preference, stopping the dissemination of control packets.

If an AS fails, *i.e.* all of its links fail (it is a rare event), then the entire region is warned. However, if a stub AS connected to only one provider fails, the provider will not warn the entire region. Data packets will fail at the provider. This is consistent with the current Internet. Even today, data packets might reach an AS just to verify that the prefix is not valid at that moment.

Routing correctness under the presence of failures

The authors prove that every concerned AS is warned in terms of reachability and routing. Transient loops are also contained and do not survive while control packets are disseminated. Finally the authors prove that if there is a path to the destination, no packet is lost and the protocol converges[ABP08].

Theorem 5. *The control packet dissemination guarantees to inform all ASes that a previously reachable AS becomes unreachable after a link failure.*

DTIA's authors prove theorem 5 by contradiction. Assume that an AS is supposed to receive a control packet, but it does not. This situation only occurs if either the AS looses reachability or all ASes have alternative paths around the failure that reach all previously reachable ASes. Both situation cannot occur simultaneously. Therefore, all ASes that have valid paths with the failed link are warned.

If we are in the presence of multiple link failures, then it is possible that some control packets will not reach some destinations. In these situations, an AS should store control packets for its neighbours until they become reachable again. Once the neighbour's link comes up again, any pending control packets are sent to this neighbour.

Theorem 5 also applies for restored links. Every AS that has a valid path using the restored link is notified. When an AS receives a link up control packet, it should cancel any pending packets for the same link as down.

Theorem 6. *The control packet dissemination guarantees to inform all AS that have to change routing decisions to maintain routing convergence.*

Guaranteeing reachability is not enough when a topology change occurs; it is imperative that paths do not loop when a link's state is altered. Assume that a link's state alters at instant t. DTIA's authors prove that at $t = t^+$ all ASes that have not converged to the same set of cycle free paths are warned with a control packet [ABP09]. We need to ensure that routing decisions are uniform to all nodes; upon the reception of a control packet if the routing decision of an AS is not altered, *i.e.* the routing decision is the same at both instants $t = t^+$ and $t = t^-$ (before the link's state alteration), then the packet dissemination stops. However, for $P2Patt$ and (BKP, y) paths' signatures there are subtle aspects. From section 3.3.2 we have learned that these paths can only be used one at the time. In case of a link failure, even if these signatures maintain their preference for a destination D, a control packet has to be forwarded. Theorems 5 and 6 have powerful properties, since both combined restrain the number of advertisements on the network.

Theorem 7. *Transient loops caused by control packet inconsistency are contained to one hop and packets loop at most one time between these two routers.*

DTIA's authors prove this theorem based on previous theorems. If a link's state changes, theorem 5 guarantees that all ASes that are affected in terms of reachability are warned. Theorem 6 also ensures that all ASes that changed routing decisions are warned. However, transient loops can occur if an AS X has processed a control packet but its neighbour X_i has not, forcing data packets to loop between X and X_i. After AS X finishes processing control packet p, it forwards p to X_i. AS X_i might have sent data packets back by this link but now it will invalidate the link $X - X_i$. If alternative paths exist, data packets will use them; otherwise they are discarded. This property guarantees that the system will remain stable, since transient loops are contained to one hop.

Theorem 8. *Assuming that there is an alternative path to destination D during failures, no data packets are lost.*

Theorem 8 assures us that a packet p is always delivered to its destination, since previous theorems guarantee that a data packet only follows cycle free paths during failures. Moreover, a packet loops at most one hop during transient failures. According to DTIA's authors [ABP08], a packet p is dropped if there are no valid paths to destination D which contradicts theorem 8 assumption.

3.4 Multi-region Routing

This sections covers the main contribution of this thesis - The multi-region routing. The DTIA's architecture assumed the existence of regions. This aspect was not elaborated in the technical reports, except for the basics. DTIA aims at improving most of BGP's limitations: routing table growth, multi-homing, churn rate, range of routing events and scalability. The key feature of implementing multi-region routing on DTIA is to improve scalability. This work is reported in a publication in the Institute of Electrical and Electronics Engineers (IEEE) *Globecom'09* Conference[AGA+09] that describes the main principles and rationale for multi-region routing.

To implement the notion of a region, we need to consider some additional features. For example, we assumed the existence of a service that performs the mapping between ASes and prefixes; with the implementation of regions we also need a mapping service that maps ASes numbers with their respective region. The aforementioned feature could be well integrated without any changes to the previous statements in section 3.2.2. The distribution of the region's graph should also be extended; a central entity should distribute a map of the region alongside with the inter-region connections. With this extension, ASes are able to calculate valid paths to other regions.

We will describe what is a region and the necessary changes to perform multi-region routing in the sequel.

3.4.1 Region definition

The work from *Amaral et al.* [ABP09] states that the number of ASes should be such that the time needed to perform the path calculation is realistic. According to their work, a number of $11,000$ ASes is doable. In addition, regions should have the following characteristics:

1. An AS must have paths to all ASes of the region (this is not drastic, since having a provider at a high level solves the issue);

2. Each region must have ASes connected to all other regions and route packets according to the rules presented in the last section (This characteristic assumes that a region has a provider at the tier-1, or an AS as a provider at the tier-1).

The second characteristic is important to avoid the definition of an inter-region reachability and routing protocol. Consequently the number of regions should be small, which implies a large set of ASes per region. As aforementioned, the region graph should contain the indication of links to other regions. Other considerations are also applied for packet forwarding. For example, a packet that it is intended to stay on a region should never leave it, in order to avoid coming back again.

Despite this definition of region, *Amaral et al.*'s work lacks information that guarantees convergence for inter-region paths. This thesis' defined a new set of rules for the multi-region case that allows the system to scale and converge.

ASes do not know how to calculate the complete inter-region path. Inter-region paths are segmented by two paths, each one belonging to its respective region. Inter-region links can be of any type as the relationships used for intra-region. We will see that the type of these inter-region links has direct consequences on which ASes are reached in order to comply with the policy rules. To guarantee reachability to other regions border-ASes exchange the ASes they can reach in their region.

With these characteristics in mind we need to define a different routing scheme in terms of path validation and ranking preference for the inter-region case because it is not possible to know the entire path.

3.4.2 Inter-Region Routing

From section 3.3 we have become acquainted with DTIA's architecture for intra-region routing. For the inter-region case we need to extend DTIA's functionalities to obtain a scalable system that converges. DTIA assumes routing based on AS identifiers instead of prefixes, which allows the system to scale. Contradictorily DTIA might suffer from scalability issues if the number of ASes grows to a large extent in the future. Various effects are expected from this fact: larger number of entries in $P(X)$, the path exploration algorithm takes more time to compute all paths, and link failures might provoke slower convergence of the network.

As a solution for scalability, NIRA and HLP tried to implement the notion of region as mentioned on chapter 2. Their rationale performs poorly when confronted with a multi-homed scenario; a multi-homed AS computes a *shortest*-path tree for each region.

To avoid an unstable system that does not scale, we should follow the following requirements for a multi-region solution:

1. The number of ASes from other regions should not influence the number of routing entries;
2. The complexity of the route computation algorithm should not be influenced by the number of ASes from other regions, nor the fact the source AS is multi-homed;
3. Route updates should be contained on their respective source region. However, if an inter-region link's state is altered, this change should be notified to both regions.

This thesis proposes a hierarchical solution to implement multi-region routing. Routing is based on a region's identifier for remote ASes; ASes identifiers are only used between border-ASes and inside the region.

The complete path for inter-region routing has two segments. The first segment follows a path from the source AS to the border-AS of the first region. After the border-link, follows the second segment that reaches the destination; this segment follows the intra-region rules from section 3.3. Path convergence only depends on the first segment, since information from the second region is omitted (for containment and scalability purposes). Besides, path convergence for the intra-region case was already proved.

There are two factors that ASes rely on for multi-region routing: the type of the path to reach an internal border-AS and the type of link used to cross to the other region. We have three distinct cases:

1. Reach a border-AS through an ascending path (either *c2p* or *p2patt* links) and the border link's type is *c2p* or *p2patt*. Based on the reachability rules, any departing border-link opens valid paths at the second region. All remote ASes are reachable at the destination region;

2. Reach a border-AS through an ascending path (either *c2p* or *p2patt* links) and the border link's type is any type but *c2p* or *p2patt*. Only a restrict set of ASes is reachable on the other region; departing border links might form invalid paths. For example, a *p2p* border link only gives access to clients at the remote region due to the reachability rules;

3. Reach a border-AS through a descending path (*p2c* links only). This case is rather complex: the border-AS has to know if a packet is following a descending direction or an ascending one to forward correctly the path. This problem does not exist in the intra-region case, where the complete path is calculated.

The first case enables reachability to any AS in the remote region and could be used extensively. However, it would turn the Internet a strict hierarchical structure, a feature that is disappearing with *p2p* links. The second case provides reachability to a strict group of ASes at the other region; nonetheless it lacks the knowledge of which ASes are reachable. The last case has a high probability of having invalid paths, providing reachability to a reduced set of ASes.

To maintain the simplicity of DTIA and avoid multi-region signalling, we must define a new routing algebra that validates inter-region paths. Path validation must follow these rules:

- The path validation process begins at the inter-region link that greatly influences the final signature;

- The validation process only uses paths in ascending direction. If both directions were considered, two problems might arise: because we are validating partial paths (from the border to the source), the direction of the path at the other region would have to be known; in addition, ASes would have to trade two lists of reachable ASes depending on the direction that packets reach the border-AS.

Border-ASes assume that data packets arrive to their border neighbours in the ascending direction and traverse the inter-region links. Based on this rationale, border-ASes calculate the list of reachable ASes and give it to their border neighbours. In section 3.4.1, we have defined that at least one border-AS exists with an inter-region link with a *p2patt* or *c2p* label. This means that we have a provider (or a peer that allows transit traffic) on the second region. Furthermore, the existence of a tier-1 provider inside a region means that all ASes of the region can reach it via an ascending path. This fact is consistent with the actual business model of the Internet, since it assures connectivity to all destinations.

If an AS receives a packet to a certain region, it has to choose which path to use in order to reach a border-AS of the destination region. The highest preference goes to *c2p* or *p2patt* paths; however a side effect of only choosing these paths is the concentration of traffic at higher hierarchical levels of the Internet, as stated above. It would be ideal that data packets would traverse border-ASes using paths from case 2, but also headed to border-ASes from case 1 if the destination is not in the AS set for the case 2 situation. A possible procedure is the following: at each intermediate border-AS, it would be verified if the destination AS is reachable; if so, the packet is forwarded to the other region. Using paths from case 2 is perfectly consistent with the current business model of the Internet, as long as the border-ASes with *p2p* or *p2c* inter-region links can reach the destination. To take advantage of these paths, the source ASes could choose the most preferred paths that traverse more border ASes; it would maximize the odds of a packet leaving earlier the region.

Each time a packet reaches a border-AS, it is verified if the destination is reachable. Otherwise the packet goes up in the hierarchy to another border-AS. This process continues, until the packet is delivered to the other region or it finds a *p2patt* or *c2p* border-link.

If no paths are found with *c2p* or *p2patt* border-links, then the source AS chooses paths with a lower rank. However, using paths with lower preference does not guarantee reachability, even if a valid path exists. This is the trade-off for not using signalling; nonetheless this situation only happens if an inter-AS failure occurs, which is rare. Case 3 fits poorly on this scenario, since a packet arriving to a border-AS might not find its destination and cannot go upwards.

3.4.3 Inter-Region Routing Correctness

It was said above that routing correctness also applies for the inter-region case. A routing protocol is correct if in a stable network (without changes) it obtains a set of loop free paths between every pair of nodes that have connectivity according to the link's policy. The inter-region algebra supports the rationale from section 3.4.2 and the paths' signatures are also calculated on the *backwards* direction, as explained on section 3.3.2. The sets of signatures Σ and labels L are the same from the intra-region case but a new $\oplus$ operation was created alongside with a new ranking table. Table 3.5 shows the $\oplus$ operation and Table 3.6 the signature's ranking. $P2Patt$ and $C2P$ paths assure reachability to any destination and they have the highest preference. Next in terms of preference, we have the $P2C$ signature followed by $P2Pbkp$ and $P2P$. $P2C$ and peer signatures only

		Signature						
		ε	$P2Patt$	$P2P$	$P2C$	$P2Pbkp$	$C2P$	(BKP, y)
Label	*p2patt*	$P2Patt$	$P2Patt$	ϕ	$P2C$	$(BKP, 1)$	$C2P$	$(BKP, y+1)$
	p2c	$P2C$	ϕ	ϕ	ϕ	ϕ	ϕ	$(BKP, y+1)$
	p2p	$P2P$	ϕ	ϕ	$P2P$	$(BKP, 1)$	ϕ	ϕ
	c2p	$C2P$	$C2P$	$P2P$	$P2C$	$P2Pbkp$	$C2P$	$(BKP, y+1)$
	p2pbkp	$P2Pbkp$	$(BKP, 1)$	ϕ	$(BKP, 1)$	$(BKP, 1)$	$(BKP, 1)$	$(BKP, y+1)$

Table 3.5: $\oplus$ binary operation for Multi-Region, at the leftmost column we have the appending link and at the upmost row we have the path's signature.

ε
$C2P = P2Patt$
$P2C$
$P2P = P2Pbkp$
$(BKP, 1)$
...
(BKP, n)

Table 3.6: Order of preference for inter-region traffic.

assure reachability to a strict set of ASes. The reason of having $P2C$ with preference over peer paths is to reflect current economic policies. However, in terms of reachability, both guarantee similar conditions to reach remote destinations.

If a node cannot reach a remote region through primary paths, backup paths (BKP, y) are used instead. It behaves exactly the same way as for the intra-region case. Still some features could be implemented just for inter-region traffic: in failure situations, a border-AS might not reach the destination in case 2, but a backup inter-region link could. However, this situation is only handled with signalling.

Based on this algebra and the results from *Sobrinho*'s work [Sob05], we prove the following two theorems that are similar to the intra-region case.

Theorem 9. *Assuming that the network has no cycles in the provider-customer relationships; the DTIA protocol converges using sets of loop-free paths to the remote region.*

Proof. Similarly to the intra-region case, inter-region routing is also monotone and not strictly monotone. So, cycles can be formed but they have to be free. If all cycles are free and the protocol is monotone, then the protocol converges.
For each path in $P_r(X)$ a signature is calculated at the inter-region link and ending at source AS X. Operation $\oplus$ from table 3.5 is applied; the rationale previously mentioned at section 3.3.2 to calculate a path's signature is also used.

From table 3.6 we can observe that the algebra is monotone; a path with signature β when extended with a new link l results a signature α whose preference is not higher than β $(\beta \preceq \alpha)$. Still the algebra is not strictly monotone since in some situations adding a link does not decrease the preference of the path; this case might force a packet to stay inside a cycle. Cycles can be formed in three situations:

1. The cycle's links have labels $c2p$;
2. The cycle's links have labels $c2p$ and $p2patt$;
3. The cycle's links have labels $p2patt$.

The first and second case do not exist according to the theorem's assumption. But in the third case a loop could exist. Similarly to the intra-region case we add an extra mechanism to make it strictly monotone for the third case. If we decrease the preference of a $P2Patt$ path each time a *p2patt* is added, then the preference of the path is no longer maintained with consecutive *p2patt* links. In terms of preference, only the path with less *p2patt* links can be used. But multipath could still be used: for example two $C2P$ paths with a $P2Patt$ path with the least number of *p2patt* links, simultaneously. This concludes the theorem's proof. □

Theorem 10. *Assuming that the preference rules from table 3.6 are followed, if either a* $C2P$ *or* $P2Patt$ *inter-region path exists, then the protocol converges to a set of loop-free paths that reach the destination AS.*

Proof. Assume the policy rules used to validate paths in $P_r(X)$ building process. If an ascending path is available, the remote border-AS is available through a *c2p* or *p2patt* link; therefore the inter-region path can be extended to any inter-region link at the remote region. If there are no invalid paths, then all remote ASes are reachable from the origin region through an ascending path. □

Note that this algebra resembles the same properties as the intra-region algebra. As such, the algebra's monotonicity allows the calculation of a path's signature by walking forward instead of walking backwards from the destination until the source AS. An extended link from a path either maintains a path signature or decreases it. The algorithm to calculate a path's signature is the same as the algorithm from section 3.3.2.

3.4.4 Failure Management

The failure management algorithm for the inter-region case is similar to the intra-region algorithm. A control packet is always sent every time a reachability or routing change has occurred. There are however two differences. The first is related to backup inter-region links. These are only used in case all other inter-region links are down. The second concerns *c2p* or *p2patt* inter-region links that fail. It might happen that certain ASes become unreachable. However this is a serious failure and at the Internet there are not many records of this type of failures. Redundant mechanisms can be conceived in the future.

For links that are restored, a similar algorithm is also applied to disseminate a control packet. The control packet is disseminated if a previously unreachable AS becomes reachable again or a more preferred path is now used.

As mentioned before, the scope of dissemination is directly related to the degree of multi-homing of the region. A high degree of multi-homing makes the disseminating region smaller, since it is highly improbable the loss of reachability to any destination. Compared to XL[LVPS08], DTIA uses more control packets due to multipath coherence requirements. XL only has to notify changes to nodes that affect directly the shortest path.

3.4.5 Deployment

For future deployment of this thesis' work, we cannot disrupt the current *de-facto* protocol BGP. We would need several years for a full-fledged deployment. However, we can start deploying DTIA with multi-region routing at smaller ASes.

Islands of regions would be deployed inter-working with BGP-based ASes. Nonetheless we need a mapping service that maps all BGP-based ASes' prefixes into an external region. DNS could be extended to perform the mapping between prefixes, ASes and regions. Furthermore, data from CAIDA or RIPE could be used to create a reliable service that distributes the region's graph.

Inter-region routing would follow the rules from section 3.4.2. DTIA's paths could be advertised to BGP regions from DTIA's border-ASes with translated ASes' prefix; however the degree of prefix advertisements should be decided.

As years progress, ASes would progressively change to DTIA from the bottom tiers to the top. This bottom-up deployment brings some advantages. If a DTIA region is connected to an external BGP region through a *c2p* link, there is no need for a mapping service, since all unknown prefixes outside of the region belong to BGP; furthermore, there is no need to reverse map BGP advertisements at border-ASes. Once regions become directly connected, the mapping service is needed to reach remote destinations; a default map to BGP can still exist until it is no longer useful.

Chapter 4

Protocol's Implementation

4.1 Introduction

This chapter's objective is to explain DTIA's implementation and its feasibility for multi-region. Chapter 3 formally presented the protocol with some insights on used techniques, some to reduce its algorithmic complexity. These techniques bring some trade-offs that are further discussed in section 4.2.

The protocol's rationale will be thoroughly explained through the visual aid of flowcharts. These flowcharts are used for easier interpretation of the protocol on section 4.3.

For a valid test bench of the protocol we have used the network simulator 2 (ns-2) [nsR09]. To test inter-domain scenarios, some modifications were added to the simulator's core to comply with our needs. These modifications are further explained in section 4.4.

4.2 Protocol's Considerations

Chapter 3 presented DTIA as a modular protocol. It reveals simplicity for separating reachability from routing. However, separating distinct functionalities might introduce some algorithmic *overload* when exploring the network's graph. We could reduce the protocol's overhead by integrating the path exploration algorithm with the path's signature calculation.

Section 3.3.2 proved that it is possible to calculate a path's signature by walking forward from the source AS to all destinations; this functionality could be easily merged with the path exploration procedure that also walks on the forward direction. As a result, while we are exploring the network's graph hop-by-hop, we are also calculating the paths' signatures. This *parallelism* of functions could scale with wider topologies. As a trade-off, it offers less flexibility for future updates of the protocol, since both layers are not separate.

Regarding multi-region routing, a few changes were also considered. Section 3.4 states that border-ASes exchange a list of reachable ASes of their region, but it does not mention how is the list structured. We have considered that each element of the list is structured as (D, S); where D is the destination AS and S is the most preferred path's signature that reaches AS D. The inclusion of this information is useful to differentiate remote paths that reach the same destination.

DTIA's authors mention that the path exploration algorithm performs reasonably well with $11,000$ ASes. The number of ASes alone might not be enough. If the ASes are massively connected it could *stress* routers in the calculation of the path exploration algorithm for each region. *Amaral et al.*[AGA+09] envisaged that five to ten regions should be enough to handle the resulting higher number of regions. However if the number of ASes grows to a great extent in the future, a solution must be adopted to handle the resulting higher number of regions. This thesis proposes a scalable solution to implement the path exploration that explores the region's graph only once for all purposes.

As a side note, this thesis did not focused on implementing a service that distributes a graph G of the network.

4.3 Algorithms

4.3.1 Path Exploration: Intra-region

This section explains the path exploration algorithm for the intra-region level based on the considerations of section 4.2 and the rationale of chapter 3. First we introduce the data structure of the current algorithm.

A path identified as P rooted at source node X, is defined as a sequence of ASes that does not loop, *i.e.* does not repeat any previously explored link. In our implementation a path P is regarded as a list of links. Each link of the path is identified by its index j where $1 \leq j \leq n$ with n as the path length. Several attributes are recorded at each index: the link's label l_j, the AS number X_j, the path's signature S_j, the path's direction D_j and a set of tuples T_j. The AS number is used to avoid loops. T_j is a list of first hop tuples t_{jm}, with $1 \leq m \leq M$, where M is the maximum number of first hop tuples. Tuple t_{jm} has two elements: FH is the AS identifier of the first hop, and d is the distance from the first hop FH until index j of path P. Tuple t_{jm} is structured as (FH, d).

A set of first hop tuples is saved at each index j of path P, in order to re-use path segments when exploring the network's graph, *e.g.* a path Q could reach AS X_j under the same conditions as path P. Further details of this operation are explained later. Table 4.1 illustrates the data structure of a generic path P.

Path P					
j	link's label l_j	X_j	Set of tuples T_j	S_j	D_j
1	l_1	X_1	$T_1 = \{t_{11}, t_{12}, ..., t_{1m}\}$	S_1	D_1
2	l_2	X_2	$T_2 = \{t_{21}, t_{22}, ..., t_{2m}\}$	S_2	D_2
...	...	...	...	...	...
n	l_n	X_n	$T_n = \{t_{n1}, t_{n2}, ..., t_{nm}\}$	S_n	D_n

Table 4.1: Path P's general structure.

As an example, let us observe figure 4.1. We have two paths P_1 and P_2 that connect X to X_2. Applying the routing rules from section 3.3.2, the final result of each path's structure is shown on tables 4.2 and 4.3.

Path P_1					
j	link's label l_j	X_j	Set of tuples T_j	S_j	D_j
1	*p2c*	X_1	$T_1 = \{(X_1, 1)\}$	*P2C*	0
2	*p2c*	X_2	$T_2 = \{(X_1, 2)\}$	*P2C*	0

Table 4.2: Path P_1's tuples ordered by index.

Path $P2$					
j	link's label l_j	X_j	Set of tuples T_j	S_j	D_j
1	*p2c*	X_2	$T_1 = \{(X_2; 1)\}$	*P2C*	0

Table 4.3: Path P_2's tuple.

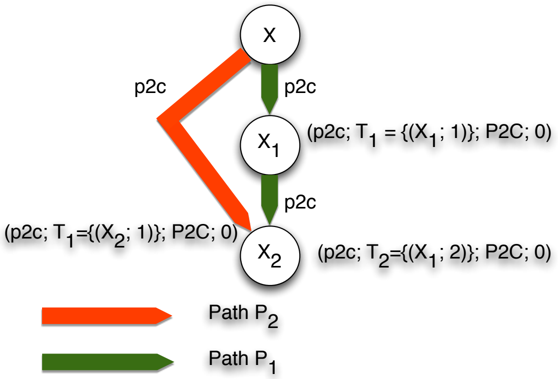
X
p2c
p2c
X1
(p2c; T1 = {(X1; 1)}; P2C; 0)
p2c
(p2c; T1={(X2; 1)}; P2C; 0)
X2
(p2c; T2={(X1; 2)}; P2C; 0)
Path P2
Path P1

Figure 4.1: Figure exemplifying the data structures at each paths' index.

To spare a router's resources, we should prevent a path P that reaches AS X_j under the same conditions as a path Q, from re-exploring the same explored links from path Q. P's segment after X_j will be equal to Q's segment if P reaches X_j under the same reachability and routing conditions, which means: with the same direction D, the same link label l and the same path signature S. Therefore, the elements of multiple paths can be merged. The resulting element stores in T_j the set of paths to which it belongs.

We illustrate the path merge operation using the generic example of paths that reach AS X_j at table 4.4. P_w is the path identifier with $1 \leq w \leq maxPath$ where $maxPath$ is the total number of paths that reach X_j. D_w is the path's direction, S_w the path's signature and l_w is the link label that reached X_j. In a path merge operation at AS X_j, from path P to path Q, all tuples that reach X_j through P should be copied to all links starting from X_j at path Q. Resources are also saved when new paths fork from a previously explored path. Let us assume that AS X_j is reached through path P. Any valid link l_{X_j} from AS X_j with $l_{X_j} \leq l_{max}$ could extend path P. The remaining $l_{max} - 1$ links could *fork* from path P to create new paths that re-use path P explored links. Similarly to the merge

Paths that traversed AS X_j			
Path Identifier P_w	Direction D_w	Signature S_w	link label l_w
P_1	D_1	S_1	l_1
P_2	D_2	S_2	l_2
...	...	...	...
$P_{maxPath}$	$D_{maxPath}$	$S_{maxPath}$	$l_{maxPath}$

Table 4.4: Paths that traversed X_j with a certain direction D_w, signature S_w and link label l_w.

case, we should also *copy* the first hop tuples from P to the newly forked path. Path P should save pointers to all forked paths. In case a merge operation occurs, the new first hop tuples are saved at the forked paths.

Each time path P traverses an AS X_j, it copies a list of first hop tuples that never reached X_j before, or repeated first hop tuples that reach X_j are replaced if P's signature is better than the previous signature that reached X_j with the same first hops. Therefore, each path element will have at T_j the set of first hop ASes that reach it, and the highest priority path signature associated to the paths that go through each first hop. Table 4.5 exemplifies with the best first hops that reach X_j.

Best First Hops FH_i that reach X_j		
Signature S_i	FH_i	d_i
S_1	FH_1	d_1
S_2	FH_2	d_2
...	...	...
S_{Fmax}	FH_{Fmax}	d_{Fmax}

Table 4.5: Best First Hop tuples that reach AS X_j

FH_i is the first hop identifier with $1 < i < Fmax$, where $Fmax$ is the number of first hops in the table, d_i is the first hop's traversed distance to reach X_j and S_i is the first hop's signature.

Acquainted with the general data structures, we can thoroughly explain the general algorithm through the visual aid of flowcharts. The path exploration procedure starts by creating an initial set of paths from the active links of the source AS X; if X has an active link to each neighbour X_i with $1 \leq i \leq N$, then we can define M single paths P_w with $1 \leq w \leq M$, where $M \leq 2N$ due to the directions D_w of the paths. It is possible to have $M = 2N$ paths at the beginning, if all link types are *p2patt* or *p2pbkp*. From figure 4.2 we can observe that when the path exploration starts at the source, each neighbour AS can be reached by at most two paths with the same signature $S = l \oplus \varepsilon$, where l is the link's label (once more due to the direction of the path). At each initial link with index $j = 1$, we fill the respective values of the path's table according to the reachability and routing rules from sections 3.3.1 and 3.3.2 respectively. The path's values are filled as presented on table 4.1 and exemplified on tables 4.2 and 4.3. Afterwards we can record the paths' properties as the best ones that traverse and reach each initial AS, as exemplified on

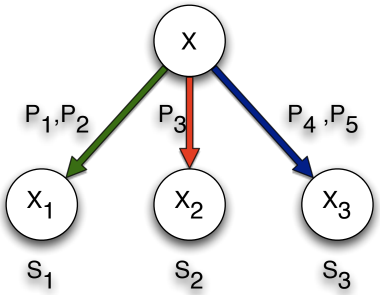

Figure 4.2: Path Initialization procedure

tables 4.4 and 4.5.

After the initialization of the first element of each path P_w, the general algorithm for path exploration begins. Figure 4.3 presents a flowchart with the general algorithm. As observed at the flowchart, the general algorithm explores the entire set of paths by processing each path P_w in a hop-by-hop process. A path is processed until it is no longer possible to walk on the network's graph, or path P_w has merged with another path. Under these conditions, the current procedure explores the next path P_{w+1}. When all paths are processed, the forwarding table is built based on the tuples recorded at each AS. A path is processed until it is no longer possible to *walk* on the network's graph, or path P_w has merged with another path.

The flowchart from figure 4.4 illustrates the algorithm that explores and validates the path P_w at each hop.

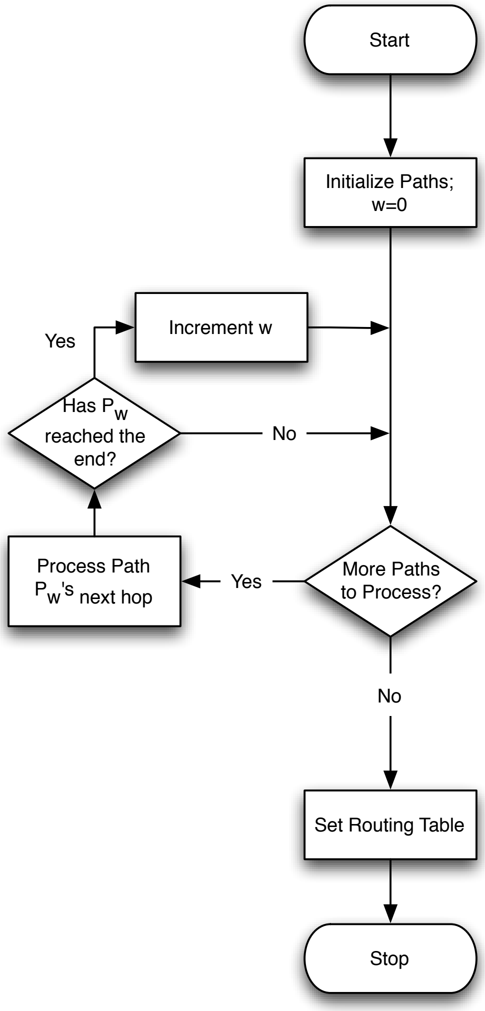

Figure 4.3: Path Exploration General Algorithm.

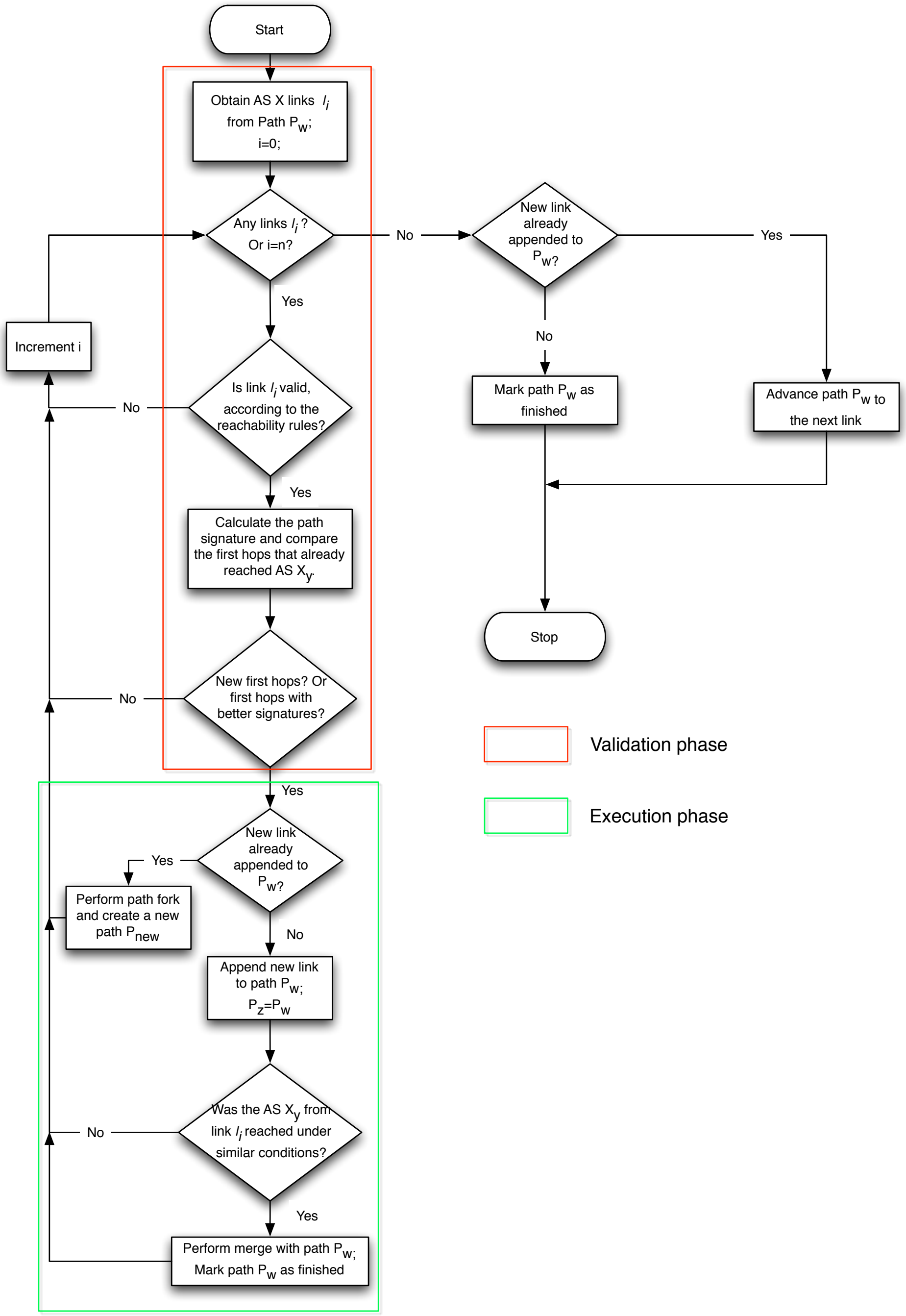
Start
Obtain AS X links l_i from Path P_w; i=0;
Any links l_i ? Or i=n?
No
New link already appended to P_w?
Yes
Yes
No
Increment i
Is link l_i valid, according to the reachability rules?
No
Mark path P_w as finished
Advance path P_w to the next link
Yes
Calculate the path signature and compare the first hops that already reached AS X_y.
Stop
No
New first hops? Or first hops with better signatures?
Validation phase
Execution phase
Yes
New link already appended to P_w?
Yes
Perform path fork and create a new path P_{new}
No
Append new link to path P_w; P_z=P_w
No
Was the AS X_y from link l_i reached under similar conditions?
Yes
Perform merge with path P_w; Mark path P_w as finished

Figure 4.4: Flowchart that illustrates the processing of path P_w.

Assume that the last explored AS from path P_w is AS X_j and that l_i is one of the X_j links, with $0 \leq i \leq l_{max}$ where l_{max} is the number of links of X_j. From the flowchart of figure 4.4 we have two distinct phases for the procedure that processes path P_w:

1. *Validation* - it defines whether a link is valid or not for further processing;

2. *Execution* - it *explores* the path P_w with the new link l_i and determines if path P_w should fork a new path P_{new} or merge into an existing path P_z.

If AS X_j does not have any links l_i, then the algorithm should end by marking path P_w as finished. Otherwise each link l_i from AS X_j is verified in the *validation* phase and explored in the *execution* phase.

At the *validation* phase, it is verified if AS X_y connected to link l_i is eligible to extend path P_w according to the reachability rules, *i.e.* it does not loop and follows the rules from tables 3.1 and 3.2. Afterwards the new path signature is computed based on the previous link signature and the rules from section 3.3.2. The first hops that use this path are compared with the first hops that already reached AS X_y. There are two situations that allow us to proceed to the *execution* phase:

1. Path P_w has new first hops that have not reached AS X_y before;

2. Path P_w has the same first hops that AS X_y has, however path P_w has first hops with better signatures than AS X_y. This situation allows us to increase the preference of the first hops that reached AS X_y. If the signature S_{P_w} of path P_w has more priority than the signature S_{X_y} of AS X_y, *i.e.* $S_{P_w} \prec S_{X_y}$, then the X_y's signature should be replaced by P_w's signature.

At the *execution* phase, the algorithm verifies if any link l_i from AS X_j has been appended to P_w. If not, the current link l_i is appended to path P_w. Otherwise a new path P_{new} is created and forked from P_w. P_{new} *inherits* all of P_w's updated tuples. All valid tuples, *i.e.* new first hops or old ones with better signatures, are recorded at AS X_y.

If the current link l_i was appended to P_w, we should verify if it is possible to merge path P_w with another path P_z. To merge both paths, P_w must reach AS X_y under the same conditions as P_z, which means having the same direction D, signature S and link label l. These conditions reproduce the same segment of links that start at AS X_y.

The flowchart from figure 4.5 illustrates the merge algorithm when a path P_w merges with a path P_z. This algorithm is divided in three phases:

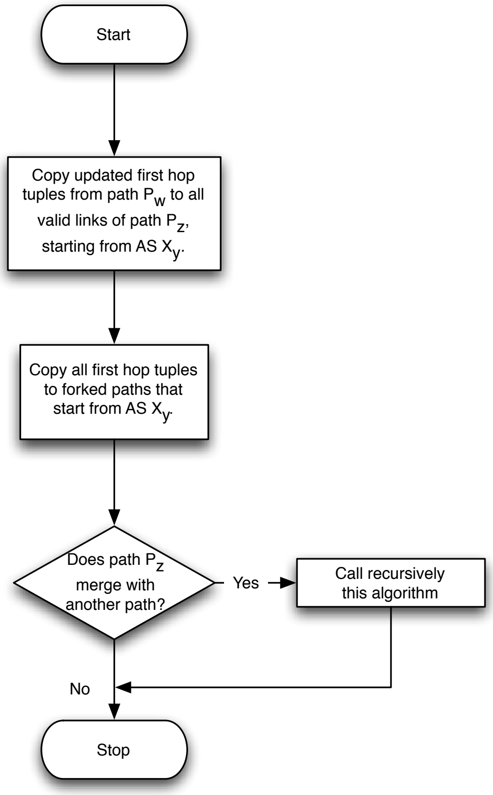
Start
Copy updated first hop tuples from path P_W to all valid links of path P_Z, starting from AS X_y.
Copy all first hop tuples to forked paths that start from AS X_y.
Does path P_Z merge with another path?
Yes
Call recursively this algorithm
No
Stop

Figure 4.5: Merge of a path P_w Algorithm.

1. Copy the set of updated tuples from the previous path P_w to all links of path P_z, starting at AS X_y;

2. Copy the set of updated tuples to all paths that forked from P_z, starting at AS X_y;

3. If path P_z is merged with a path P_{zz}, the merge algorithm is called recursively to update P_{zz} with new tuples.

Once the merge algorithm ends, the procedure that validates and explores X_j's links continues. When all paths are processed, the general algorithm subsequently defines the routing table for all destinations.

Each destination X_j has a list of best first hops with their respective signature and distance. These first hops are ordered according to the signature and the distance from the source AS X to destination X_j. Tuples are ranked according to three rules:

1. Assume that $S_1 \neq S_2$, where S_1 is the signature of the tuple $(FH_1, \_)$ and S_2 is the signature of the tuple $(FH_2, \_)$. The first tuple is more preferred than the second if S_1 has a higher preference than S_2;

2. Assume that $S_1 = S_2$, where S_1 is the signature of the tuple (FH_1, d_1) and S_2 is the signature of the tuple (FH_2, d_2). The first tuple is more preferred if d_1 is lower than d_2, *i.e.* FH_1 needs less links than FH_2 to reach the destination.

3. If $S_1 = S_2$ and $d_1 = d_2$, by default we assume that tuple (FH_1, d_1) is equally ranked as (FH_2, d_2).

Note that these criteria are just used to order the tuples t_{jm} recorded at each destination; the second rule is only useful when using a single route or to differentiate $P2Patt$ and backup paths. To build the multipath routing table, at each destination we select the best ranked first-hops that have equivalent signatures, according to the rules explained at chapter 3.

Intra-region routing in a multi-region scenario apply the same algorithms. Border-ASes use the algorithm to obtain the destinations reachable from their border-neighbours. Each destination from the border-neighbour's list is structured as (D, S), with D as the destination and S as the most preferred signature that reaches D. Since we know the path's signature at the remote region, we can rank the remote first border-hops according to the defined rules of this section.

Best First Hops that reach X_j		
Signature	First Hop	distance
$P2Patt$	a	2
$P2Patt$	b	9
$P2C$	c	1
$P2C$	d	3
$P2P$	e	3
$P2P$	f	3

Table 4.6: Best first hop tuples that arrive destination X_j.

Table 4.6 exemplifies the first hop tuples that arrive at destination X_j. The first hops are enumerated from a to f. Each tuple does not show the path's direction D, since its value does not influence the ranking decision. Assuming that the path exploration algorithm ended, we rank all tuples according to the aforementioned rules. The tuple's ranking is

Final Ranking		
Signature	First Hop	distance
P2C	c	1
P2Patt	a	2
P2C	d	3
P2Patt	b	9
P2P	e	3
P2P	f	3

Table 4.7: Ranking of the first hop tuples that arrived X_j.

shown on table 4.7. The first hops that have *P2Patt* or *P2C* signatures are selected to forward packets to destination X_j. However, according to the rules defined on chapter 3, the selected next hops would be *c*, *a* and *d*. We cannot use more than one *P2Patt* path simultaneously; neighbour *b* needs more hops to reach X_j than neighbour *a* does, therefore neighbour *a* is selected.

4.3.2 Path Exploration: Inter-Region

This section explains how to perform the path exploration algorithm for the multi-region case. Exploring paths for the multi-region case differs slightly from the intra-region presented at section 4.3.1. The main rationale behind both cases is similar. However the algorithm for the multi-region case must be *independent* of the intra-region case; both cases have *conflicting* preferences according to the preference rankings of tables 3.4 and 3.6.

The main difference between the algorithm that handles the inter-region case and the intra-region algorithm are that the path list also includes the links that reach other regions. The presentation below is based on the theoretical model of section 3.4 and on the assumptions of section 4.2, the differences between the intra-region and multi-region cases are further discussed.

The inter-region case also uses the set of paths defined above for the intra-region case. However, it adds a new set of tuples B_A to support inter-region based routing. Each element of this set is a tuple (B, R), where B is the remote border-AS and R is B's region. This allows a route selection based on a region identifier.

The general algorithm applied for the intra-region case, on figure 4.3, is also applied for the inter-region case. In the same fashion as the intra-region case, the path initialization procedure creates a set of paths P_w with $1 \leq w \leq M$, that reach an initial set of ASes X_i with $1 \leq i \leq N$ and $M \leq 2N$.

The source node X at each initial path P_w, obtains from neighbour X_i all valid neighbour links, including the border-links that reach other regions compatible to the reachability rules; each border-link that reaches a border-AS B from a remote region R is appended to the list B_A as a tuple. B_A lists all reachable remote border-ASes.

Regarding the path exploration algorithm, it works similarly to the intra-region case. A tuple $t_{jm} = (FH, d, B_A)$ is appended to path P_w, alongside with the path's signature S and direction D. Table 4.1 can still be used for this case. The only difference is the recorded tuple t_{jm} that also contains the list of reachable border-ASes.

We register to a table exactly similar to 4.4, path P_w's properties at the AS X_j, for merging purposes. The first hop characteristics should be also recorded at X_j's best hops table as exemplified by table 4.8. S_w is the path's signature, FH_w is the first hop from path P_w, d_w is the traversed distance of the first hop tuple and B_{A_w} is the list of reachable remote border-ASes, with $1 \leq w \leq Fmax$ where $Fmax$ is the maximum number of first hops. Table 4.8 could be also used to register the best hops for a given region R.

Best Region First Hops that reach X_j			
Signature S_w	FH_w	d_w	B_{A_w}
S_1	FH_1	d_1	B_{A_1}
S_2	FH_2	d_2	B_{A_2}
...	...	...	...
S_{Fmax}	FH_{Fmax}	d_{Fmax}	$B_{A_{Fmax}}$

Table 4.8: Best Region First Hops at AS X_j. This table can also be used for a region R.

Section 4.3.1 referred that a path P_w ends exploring the network's graph under two conditions: either the last explored AS X_j (with j as the link's index) does not have more links, or path P_w merged with another path. Here it is also considered that a path ends once it reaches a border-AS from a remote region; the network graph of a remote region is unknown, therefore it is *impossible* to explore more links.

Concerning the algorithm that processes a path at each hop for the intra-region case (the flowchart from figure 4.4), we have two phases: *validation* and *execution*. Applying the rationale for the multi-region case, we need to modify the *validation* phase. If AS X_j is an inter-region AS we mark path P_w as finished. Otherwise X_j's links are followed. Regarding the first hop comparison at the *validation* phase, some changes were introduced since we need to achieve the maximum number of border-ASes for all regions. If path P_w achieves AS X_y (X_y is connected to one of X_j's links) with new first hops, then the algorithm continues to the *execution* phase. Suppose that path P_w has the same first hops as AS X_y, then we need to verify if path P_w has better characteristics to replace X_y's best hops tuple. The following rules must be verified:

1. Suppose the signature S_{P_w} and the tuple $(FH, d_{P_w}, B_{A_{P_w}})$ from path P_w, and signature S_{X_y} and the tuple $(FH, d_{X_y}, B_{A_{X_y}})$ from AS X_y. FH is the first hop, d is the distance and B_A is the list of visited border-ASes. P_w's tuple and signature should replace AS X_y's tuple and signature if $S_{P_w} \prec S_{X_y}$.

2. The following rule tie-breaks the previous one. If $S_{P_w} = S_{X_y}$, then P_w's tuple should replace X_y's tuple characteristics if path Pw can reach each region with more visited border-ASes than X_y.

Analysing the second rule we can conclude that the solution is non-optimal, since we are not exploring the region's graph for each remote region. This solution could be useful when faced with several regions; but from five to ten regions it should be feasible according to DTIA's authors [ABP09].

Regarding the *execution* phase on figure 4.4, it does not suffer major changes in terms of rationale for the multi-region case, unless collecting at each appended link all reachable border-ASes and update the tuple-set T_j at link's index j of path P_w. For the merge algorithm in figure 4.5, no changes are required for the inter-region case.

Once all paths are explored, the general algorithm defines the routing table for each region R. Similarly to the intra-region case, the first hops are ranked according to the following rules to tie-break them:

1. Assume that $S_1 \neq S_2$, S_1 is the signature of first hop FH_1 and S_2 is the signature of the first hop FH_2. FH_1 is more preferred than FH_2 if S_1 is more preferred than S_2;

2. Assume that $S_1 = S_2$, S_1 is the signature of first hop FH_1 and S_2 is the signature of first hop FH_2. If we have a tuple $(FH_1, \_, B_{A_1})$ and a tuple $(FH_2, \_, B_{A_2})$, with B_{A_1} as the number of border-ASes that FH_1 can reach and B_{A_2} as the number of border-ASes that FH_2 can reach. Then FH_1 is more preferred than FH_2, if it can reach more border-ASes than FH_2.

3. Assume that $S_1 = S_2$ and that B_{A_1} has the same number of visited border-ASes as B_{A_2}. Tuple (FH_1, d_1, B_{A_1}) belongs to FH_1 and tuple (FH_2, d_2, B_{A_2}) belongs to FH_2. d_1 and d_2 are the traversed distances to reach region R. FH_1's tuple is better than FH_2's tuple if d_1 is lower than d_2, *i.e.* FH_1 needs less links than FH_2 to reach the destination region R;

4. If the previous rules do not apply then we consider that both tuples are equally ranked.

At the end, the ranking procedure selects the first hops with the most preferred signature for multipath routing, according to section 3.4.

4.3.3 Failure Management Algorithm

The failure management algorithm follows the rationale explained on sections 3.3.3 and 3.4.4. No considerations were added for the theoretical models of intra-region and multi-region cases.

Figure 4.6 illustrates the flowchart for an AS X detecting a change of a link's state. If the link l changed its state, *i.e.* from up to down or vice-versa, AS X must verify the following rules to warn its *in-region* neighbours:

1. Verify if any AS Z in the network changed its state of reachability, *i.e.* either Z is now unreachable or reachable;

2. Verify if the previous signature S_Z of an AS Z has changed (either the preference decreased or increased);

3. Verify if AS X has lost reachability to any region R;

4. Verify if the previous signature S_R of a region R has changed (either the preference decreased or increased).

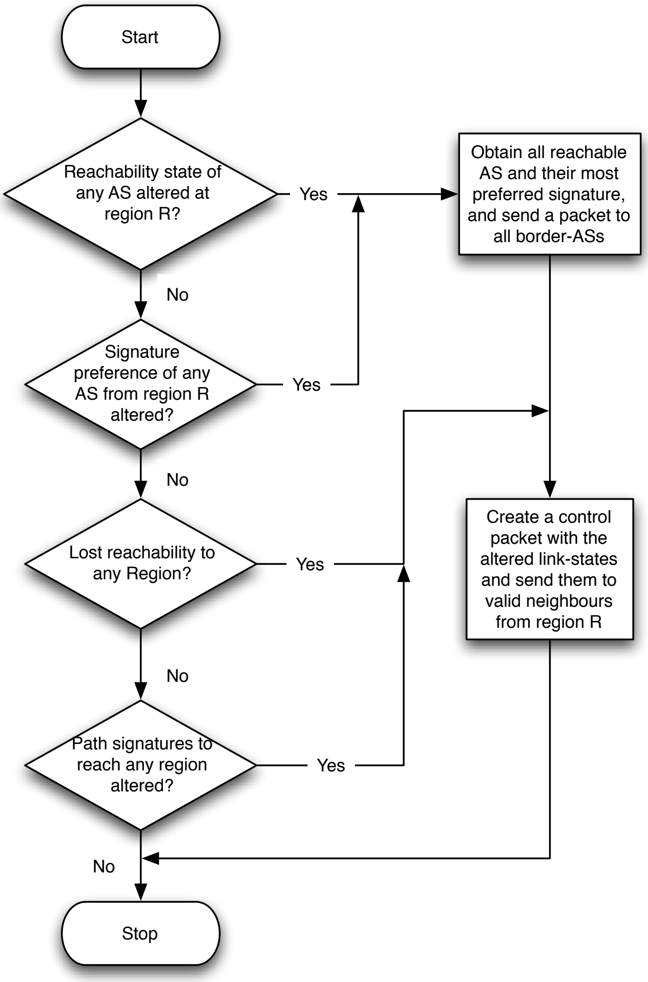
Start
Reachability state of any AS altered at region R?
Yes
Obtain all reachable AS and their most preferred signature, and send a packet to all border-ASs
No
Signature preference of any AS from region R altered?
Yes
No
Lost reachability to any Region?
Yes
Create a control packet with the altered link-states and send them to valid neighbours from region R
No
Path signatures to reach any region altered?
Yes
No
Stop

Figure 4.6: Decision process to disseminate a control packet.

If any of these rules are confirmed, then AS X's *in-region* neighbours are notified of the altered links; this notification follows the reachability rules of section 3.3.1. Neighbours receiving a control packet p should verify the same rules from figure 4.6 to continue the dissemination process. If none is confirmed, the packet is dropped and the dissemination stops. If an AS Y is down and is eligible to receive packet p, a copy of packet p is saved until AS Y comes up again.

Multi-region neighbours are notified if any of the first two rules are confirmed. A list of reachable ASes from X's region is sent to them; this list complies with the routing rules of section 3.3.2. At the end, multi-region neighbours update their databases and re-check the ranking of signatures for the affected region.

4.4 The network simulator 2 (ns-2)

This section presents the ns-2 [nsR09] simulator and the introduced changes to implement this thesis' proposed protocol. ns-2 is a valuable tool for researchers to test network protocols, either wired or wireless. This simulator gives a proper basis to modify or create mechanisms at each layer of the OSI model [osi09]. The software's source code is open, which gives users enough *flexibility* to modify it and correct some of its flaws. However, using this software implies a great knowledge of its mechanisms, we assume that the reader has a basic knowledge of ns-2 mechanisms[1].
ns-2 mechanisms are supported through the *Tcl*[tcl09] scripting language and the *C++* programming language. *Tcl* is used to setup simulation scripts, but also as an interface to perform commands to *C++* objects that are *mirrored* in *Tcl.* Regarding the *C++* language, it is used to define protocols' mechanisms. As an example a *Tcl* script may perform a command that triggers an event to disable a physical link. This event may call a *C++* routine to warn a routing protocol that a link is down; as a consequence, the routing protocol will calculate the new routes for all destinations.

To test a network scenario, ns-2 reads a simulation script in *Tcl* and performs a discrete-state simulation. For each protocol, independent of its layer, a *Tcl* object is created at each node. This object is usually named as *agent* and it serves as an endpoint for packets. The interaction between layers is supported by a special module called *classifier.* A *classifier* is also used for other purposes, such as forwarding packets to other nodes.

The ns-2 simulator provides several routing protocols to test bench, but none of them are fit for an inter-domain scenario. This thesis re-used ns-2's link-state protocol, since most of its basic mechanisms are essential for important events, *e.g.* link's state alteration or packet reception.

Tcl commands are used at each routing agent, to define the links' label and the neighbours' AS number and region. The following lines present a sample of a *Tcl* script defining the node's AS number and region.

```
set rtobj [$node rtObject?]
set rtproto [$rtobj rtProto? THESIS]
```

[1] For a thorough reading of the software's manual see [nsM09]. A quick tutorial is also available from *Marc Greis* web page [nsT09]

```
3 $rtproto cmd setAS $ASnumber
4 $rtproto cmd setRegion $ASregion
```

The first code line is used to obtain a node's routing object; a routing object contains all routing protocol agents that the node is using. The second code line is used to obtain this thesis' routing protocol agent. The third line executes the command *setAS* at the routing agent, informing the *mirrored C++* agent about its AS number. Regarding the fourth line, it defines the node's region with the *setRegion* command.

To define the link's relationship between two nodes, we use each node's routing agent to define the link's properties; these changes are only reflected at the routing agent. The following *Tcl* sample defines a link's relationship between two nodes:

```
1 $rtproto1 cmd setPolicy [$node2 id] $AS2 $REGION2 $POLICY2
2 $rtproto2 cmd setPolicy [$node1 id] $AS1 $REGION1 $POLICY1
```

The command *setPolicy* defines the link's policy label from the perspective of the routing agent, it accepts four arguments in the following order:

1. The neighbour's identification;
2. The neighbour's AS number;
3. The neighbour's region;
4. The neighbour's relationship with the routing agent. There are five string policy types : *PEER*, *CUSTOMER*, *PROVIDER*, *PEERATT* and *BACKUP*.

The previous example listed two commands; each command orders the routing agent of *node1* and *node2*, to assign the policy's characteristics from *node2* and *node1* respectively.

To distribute the network's topology at each region, we have re-used link-state's flooding mechanism to distribute the nodes links' state; this process is stopped once all nodes know the network's graph. The following *Tcl* commands are related with the network's graph:

```
1 $rtproto cmd sendUpdates
2 $rtproto cmd startRouting
3 $rtproto cmd tradeRouting
4 $rtproto cmd stopRegionSpread
```

The first command orders the routing agent in *C++* to start flooding link-state packets inside the node's region. As for the second command, it instructs the routing agent to perform the path exploration algorithm. Subsequently the third command orders the routing agent to send a list of reachable destinations to all border-ASes; the *tradeRouting* command does not work unless the routing agent performed the second command. After issuing the fourth command, all nodes will commence using the developed protocol of this thesis. Each function of the link-state protocol was subsequently altered to meet our needs, according to the algorithms on section 4.3.

Further changes were also included on the ns-2 package: A mapping service and a new classifier module. Concerning the first, a node's routing agent may not know the identification of some destinations since they might belong to other regions. To solve this issue, a new object was created to perform a mapping service. This object maps a node's identification with its respective region; each node has an instance of this service to obtain the region identifier of unknown destinations. The following *Tcl* commands are related with the mapping service:

```
set asns [new ASNS]
$asns cmd add-entry $nodeId $ASnumber $Region
$rtproto add-asns $asns
```

The first command creates the mapping service named *asns*. Regarding the second command, it registers the node's information at the *asns* object. The *add-entry* command has three inputs: the node's identifier, AS number and region. Regarding the *add-asns* command, it assigns the mapping service object to the node's routing agent.

A new classifier module was also created; it supports multipath routing for multi-region. A packet p destined to a remote destination X_z, must traverse a series of intermediate border-ASes. Assume that border-AS X receives p; X must check if p can be forwarded to a border-neighbour. Otherwise packet p must go upwards on the hierarchy. This rationale must follow the reachability rules of chapter 3.

Normally a classifier does not support *conditional* forwarding based on the aforementioned rationale; it forwards packets based on the output links that a routing protocol has selected. For this thesis, we have designed a classifier that follows intra-region and inter-region rules; in addition, this classifier must respect the reachability rules.

Figure 4.7 exemplifies the usefulness of the new classifier. Assume that AS X has two border-links to region R; the first link is a normal peering link ($p2p$) and the second is a provider's link ($c2p$). Let us consider that the first link is directly connected to AS X_z; the second allows *full* reachability through AS X_p. If AS X receives a packet p from a $p2pbkp$ link destined to X_z, then X must forward the packet through the $c2p$ link. Although X is directly connected to X_z, X should forward data packets to X_p; otherwise, it would not respect the reachability rules.

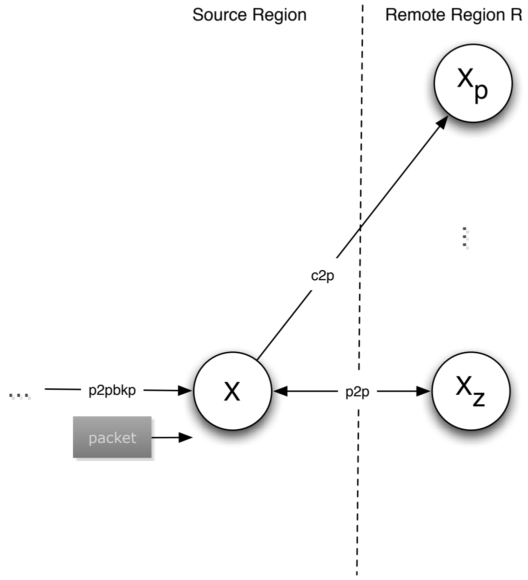

Figure 4.7: Packet forwarding issues using a normal classifier.

The new classifier, re-uses the *MultipathForwarder* class that forwards packets in round-robin. A restrictions table T was introduced to verify if it is possible to forward packets based on the reachability rules. This table is indexed by the input link l_i and the output link l_o. Table 4.9 exemplifies an AS X restrictions table with three neighbours: W, V and Y. ASes W and V are providers of X, while Y is a customer of X. We can observe that any packet coming from W cannot be forwarded to V or *vice-versa*. Since Y is a customer of X, AS X can forward packets from W or V to AS Y.

		Output Link lo		
		W	V	Y
Input Link li	W	-	Invalid	Valid
	V	Invalid	-	Valid
	Y	Valid	Valid	-

Table 4.9: Restrictions table from AS X.

Furthermore, a sequence of decisions is executed on the classifier's code to forward packets:

1. Check in round-robin if any of the first-hops are valid to reach destination D. If D is not reachable, or any combination of the input and output links do not *respect* the restrictions from table T, then we should execute the next decision. Otherwise the packet is forwarded through a valid combination of the input and output links.

2. If D is a remote destination, then we should check in round-robin if any of the first-hops that have full reachability to D's region are valid. If D's region is not reachable, or any combination of the input and output links do not *respect* the restrictions from table T, then we should drop the packet. Otherwise the packet is forwarded through a valid combination of the input and output links.

The first rule is applied to intra-region and multi-region destinations; these destinations are reachable within the source node's region. A reachable multi-region destination could be either a directly connected border-AS, or a reachable destination from a border-AS neighbour. Concerning the second rule, it is only applied if a remote destination D does not comply the first rule. A packet is dropped if both decision rules are not confirmed. If any of these rules comply, the packet is forwarded.

Chapter 5

Performance Analysis

5.1 Introduction

This chapter presents a performance analysis of the developed protocol. For a fair analysis of the protocol, we have compared it against BGP. We used BGP++[bg309], a BGP implementation for the ns-2 simulator. This simulator is based on an open-source implementation of the BGP protocol, the GNU/Zebra[gnu09].

Section 5.2 presents the tested topology for our experiments; this topology is based on CAIDA's data. Afterwards, section 5.3 presents our experiments concerning different topics:

- Routing table analysis for the intra-region and inter-region cases;
- Packet signalling analysis, concerning the intra-region and inter-region cases;
- Delay analysis of data packets at the inter-region level.

5.2 Topology Characteristics

Here we present the tested topology for our experiments. The topology was obtained from CAIDA's AS Relationships Data Research project [cai09a]; we have selected 54 ASes for our research. A bigger topology could have been used, however ns-2 consumes excessively computer resources. Figure 5.1 illustrates the topology seen in ns/nam; ns/nam is the ns-2 built-in network animator package. This topology was partitioned in two regions, each one with 27 ASes. The rules from section 3.4 were followed to ensure that all ASes have full reachability to any destination (remote or not); as a result five tier-1 providers were selected for each region. The regions are named R_0 and R_1. On figure 5.1, the nodes in yellow are the ASes from region R_0 and the nodes in white are the ASes from region R_1.

The ASes' Institution for each node of figure 5.1 is listed on table 5.1.

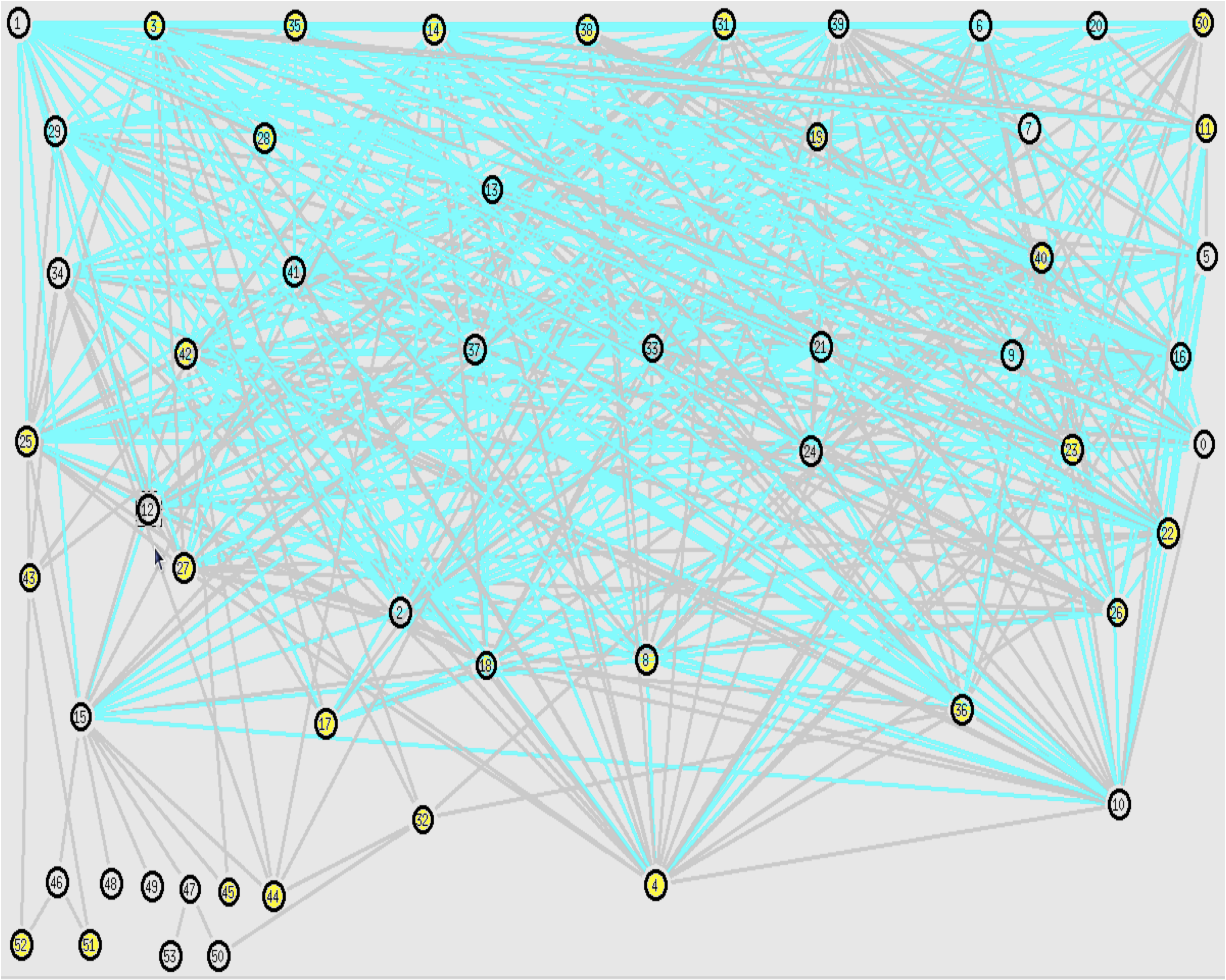

Figure 5.1: ns/nam topology. Links in cyan are peering links; grey links represent provider to customer relationships from top to bottom. Nodes in yellow belong to region R_0 and nodes in white belong to region R_1.

Node	AS	Institution Name	Region
0	8708	RDSNET	R_1
1	6939	Hurricane Electric	R_1
2	2497	Internet Initiative Japan Inc.	R_1
3	3549	Global Crossing	R_0
4	12956	Telefonica	R_0
5	6830	UPC Broadband	R_1
6	4323	TW Telecom Holdings	R_1
7	9002	RETN Limited	R_1
8	5400	BT Global Services	R_0
9	4766	Korea Telecom	R_1
10	6762	Telecom Italia Sparkle	R_1
11	22773	Cox Communications	R_0
12	5413	GX Networks	R_1
13	1299	TeliaSonera AB Networks	R_1
14	174	Cogent Communications	R_0
15	8657	Portugal Telecom	R_1
16	3303	SWISSCOM	R_1
17	3216	Golden Telecom	R_0
18	1273	Cable and Wireless IP GSOC Europe	R_0
19	19151	WV FIBER LLC	R_0
20	2828	XO Communications	R_1
21	13237	LambdaNet Communications	R_1
22	2516	KDDI Corp.	R_0
23	3786	LG DACOM Corporation	R_0
24	8928	Interoute Communications	R_0
25	286	KPN Internet Solutions	R_0
26	6539	Bell Canada	R_0
27	3491	Beyond The Network America	R_0
28	20932	IP-MAN.Net Engineering	R_1
29	6461	MFN - Metromedia Fiber Network	R_1
30	7018	AT&T WorldNet Services	R_0
31	701	MCI Communications Services	R_0
32	2860	Novis	R_0
33	3561	Savvis	R_1
34	702	MCI Communications Services	R_1
35	209	Qwest Communications Company	R_0
36	5511	France Telecom - Orange	R_0
37	3257	Tinet SpA	R_1
38	1239	Sprint	R_0
39	3356	Level 3 Communications	R_1
40	3320	Deutsche Telekom AG	R_0
41	2914	NTT America, Inc.	R_1
42	6453	TELEGLOBE IP ENGINEERING	R_0
43	9186	ONI TELECOM	R_0
44	13156	CABOVISAO	R_0
45	12542	TVCABO	R_0
46	3243	TELEPAC	R_1
47	15525	PT PRIME	R_1
48	15457	Cabo Tv Madeirense	R_1
49	42863	TMN	R_1
50	35038	INESC	R_1
51	34873	IGIF-Ministério da Saúde	R_0
52	25253	Caixa Geral de Depósitos	R_0
53	43643	Tap Air Portugal	R_1

Table 5.1: Nodes identification from figure 5.1

Figure 5.1 shows a massively connected network, yet it is only a small glimpse of the Internet. From top to bottom, ASes are connected through *p2c* links in grey; links in cyan are normal peering relationships. At the top we have ten tier-1 ASes that are massively connected to ASes from lower layers with peering links. To comply with the full reachability conditions, it is assumed that tier-1 providers are connected with each other with *p2patt* links. HLP and NIRA cannot *inhabit* in such topology; both protocols were modelled considering that the Internet is a pure hierarchy, without peering links traversing tiers.

Before characterizing further the topology from figure 5.1, we should explain the definition of a node's degree. A node's degree is the number of links that a node has to other neighbours[FFF99]. The chart from figure 5.2 illustrates the ASes' degree cumulative percentage on figure 5.1; the chart was obtained using *OpenOffice Calc*[ooR09].

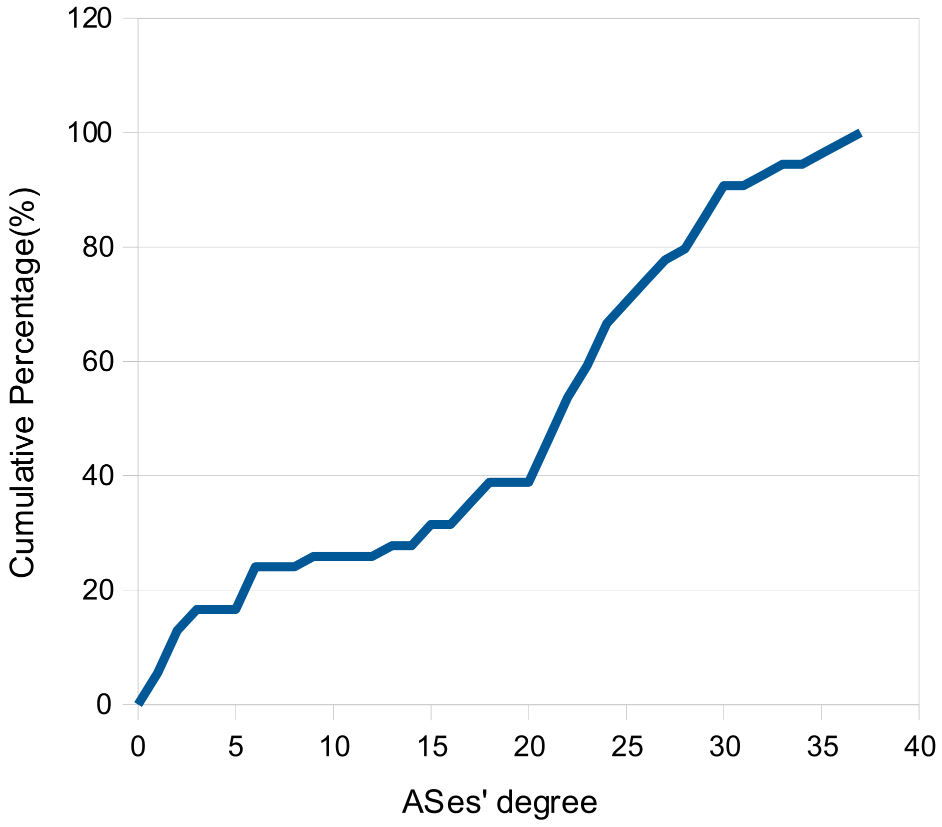

Figure 5.2: ASes' degree distribution in percentage.

Chapter 2 stated that the Internet does not follow a pure power-law distribution, according to CAIDA's members. The chart from figure 5.2 somehow supports the aforementioned statement. Partitioning the graphic, we can obtain three distinct functions as power-laws. Define n as the node's degree and $CP(n)$ as the cumulative in percentage. From *OpenOffice Calc* we have obtained the trend curves that characterize the topology, with R as the correlation factor; equation 5.1 shows the curves' expression:

$$CP(n) = \begin{cases} 6.98 \times n^{0.62} & \text{if } n \leq 10 \text{ , } R^2 = 0.91 \\ 1.1 \times n^{1.25} & \text{if } 11 \leq n \leq 25 \text{ , } R^2 = 0.91 \\ 5.36 \times n^{0.82} & \text{if } 26 \leq n \leq 37 \text{ , } R^2 = 0.93 \end{cases} \tag{5.1}$$

From the correlation factors we can conclude that CAIDA's data resembles a power-law distribution. If we had more ASes to test, the cumulative percentage would be closer to a scale-free network according to recent studies.

The topology from figure 5.1 is further characterized for the inter-region case. The charts from figures 5.3 and 5.4, compare regions $R0$ and $R1$ in terms of intra-region links and inter-region links respectively.

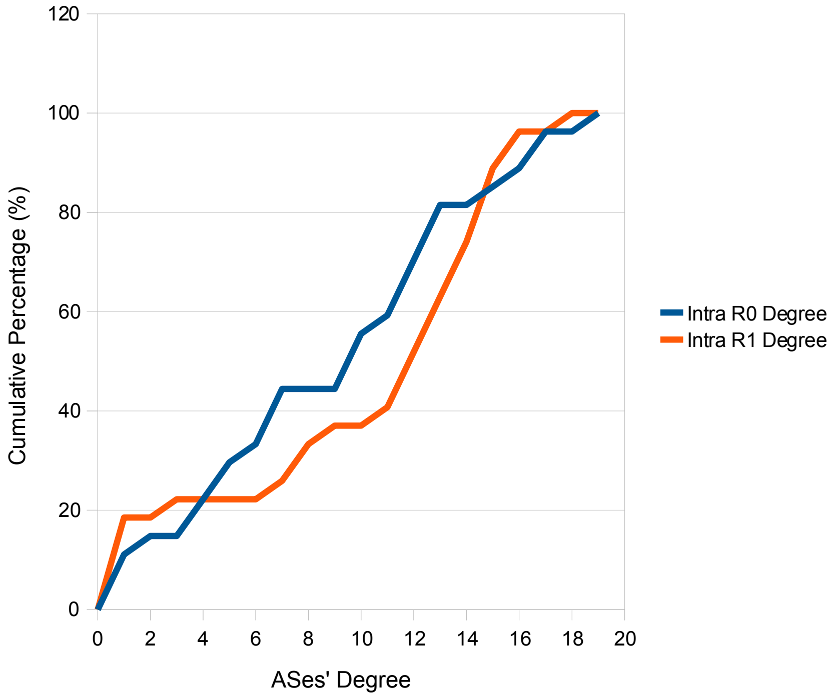

Figure 5.3: Comparison of ASes' degree for intra-region links between regions R_0 and R_1.

Comparing the nodes' degree for intra-region links in figure 5.3, we can observe that both regions are *unevenly* distributed. An example of this *inequality* is the small *umbrella* of ASes that belongs to region R_1, on the left bottom corner of figure 5.1; this *umbrella* covers a small set of intra-region links from Portuguese domains. On region R_0 we have the opposite example: AS *Telefonica*(node 4) has 22 $c2p$ and 8 $p2p$ links. This dissimilarity allows us to perceive the differences between both regions. From these examples, we can also conclude that the degree of multi-homing of an AS is not correlated to its tier [AGA+09].

Observing the chart from figure 5.4, it is noticeable that some ASes from region R_1 do not have inter-region links. We can regard from figure 5.1 that some domains are stub-ASes. These stubs must belong to the same region as their providers; otherwise these stubs would not reach any destination from their region.

Table 5.2 distinguishes the number of inter-region links for each type of relationship.

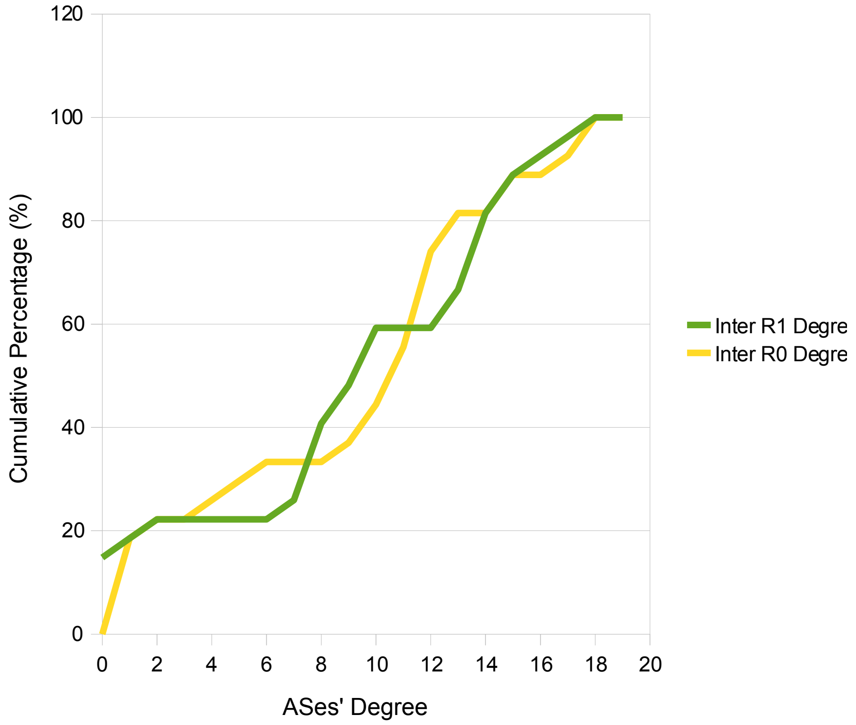

Figure 5.4: Comparison of ASes' degree for inter-region links between regions R_0 and R_1.

#*c2p* Links		#*p2p* Links	#*p2patt* Links	Total
Region 0	Region 1			
56	38	141	18	253

Table 5.2: Number of Inter-region Links of each type.

As observed from table 5.2, we have a total of 253 inter-region links; the total number of links for a single region is 517. The number of inter-region links is almost half of the total number of links. We need to differ the inter-region links that allow full reachability from those that only reach a strict set of remote destinations. We can observe that region R_0 has more C2P and P2Patt links than R_1; in result region R_0 has less chances of loosing reachability to R_1 than the opposite. Nonetheless, chapter 3 stated that the probability of an inter-domain link failing is rare.

If we analyse inter-region links P2P and P2C, region R_1 is in numerical advantage compared to R_0. A packet with origin R_1 has an higher probability of leaving R_1 earlier through a P2P or P2C link than a packet sent from R_0.

Familiarized with the network's topology, we can proceed to the next section that describes our experiments in ns-2.

5.3 Experiments

This current section describes the experiments made on the ns-2 simulator. Several analysis were carried out:

1. Routing table analysis on section 5.3.1;

2. Routing messages analysis on section 5.3.2;

3. Delay analysis of data packets at the inter-region level on section 5.3.3.

On the first experiment, it is verified how well the protocol scales at the inter-region level, concerning the routing tables' size. The BGP++ implementation was not tested on this trial; from chapter 2 we have learned that BGP cannot take full advantage of multipath routing. In terms of routing entries, it would be unfair to compare BGP with our implementation, since BGP only advertises the *best* route.

Regarding the second experiment, an analysis is made in terms of packet signalling; for this trial we have used the BGP++ implementation for comparison.

The last experiment assesses our protocol in terms of packet delay at the inter-region level.

5.3.1 Routing table analysis

Section 3.4 described that routing tables should not be influenced by the number of remote ASes; however border-routers that have strict reachability receive a list of reachable destinations from their border-neighbours. It is important to verify the scalability of this thesis' protocol in terms of the number of routing entries.

The topology from figure 5.1 was used in ns-2 to obtain the number of routing entries at each node. Since our system supports multipath routing, we account the total number of reachable entries regardless of the first hops' signature. Concerning the inter-region case, we have considered the total number of entries as the sum of the intra-region entries with the inter-region entries.

The chart from figure 5.5 compares the number of routing entries for the single region and inter-region cases; the chart compares both cases with cumulative percentages. From figure 5.5, we conclude that the number of entries diminishes significantly with the introduction of regions; this can be justified by the number of links that allow full reachability to the remote region, as perceived from section 5.2.

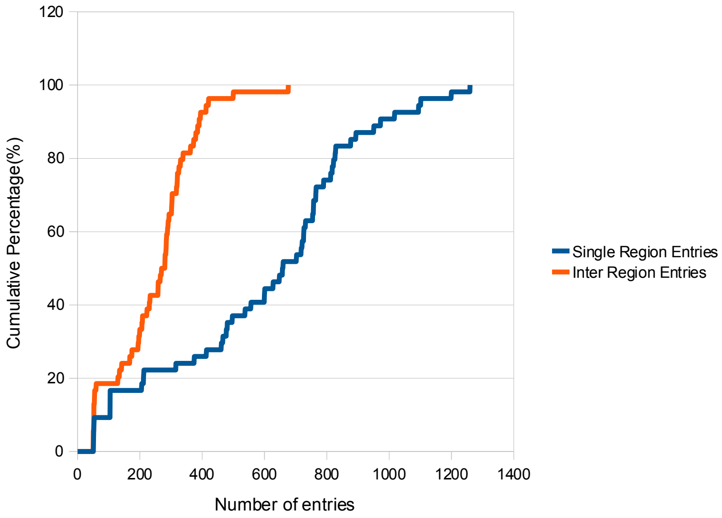

Figure 5.5: Comparison of the number of routing entries for both intra-region and multi-region.

The numbers of remote entries are limited because the details from a remote region are hidden. However border-ASes receive lists of reachable destinations from their remote-neighbour that has strict reachability. We have collected the number of remote destinations that are reachable at each remote neighbour; these values were further processed to obtain the mean value $\overline{x}$ and the respective deviation value σ; table 5.3 shows the results. An average of 87.7% of the topology is reachable through the remote border-ASes that have strict reachability.

$\overline{x}$	σ
23.72	1.43

Table 5.3: Mean value of destinations that are reached remotely from border routers that have strict reachability.

The results from table 5.3 are not surprising if we observe the number of inter-region links from table 5.2; the total number of *p*2*p* and *p*2*c* inter-region links are a significant fraction of the total number of links, close to 45%. If we had a non massive multi-homed topology, these figures would be lower in terms of remote destinations that are reachable through border-neighbours.

The chart from figure 5.6 presents the cumulative percentages for each kind of inter-region entries; it compares the number of first hops that allow full reachability with those that only allow strict reachability to a remote region.

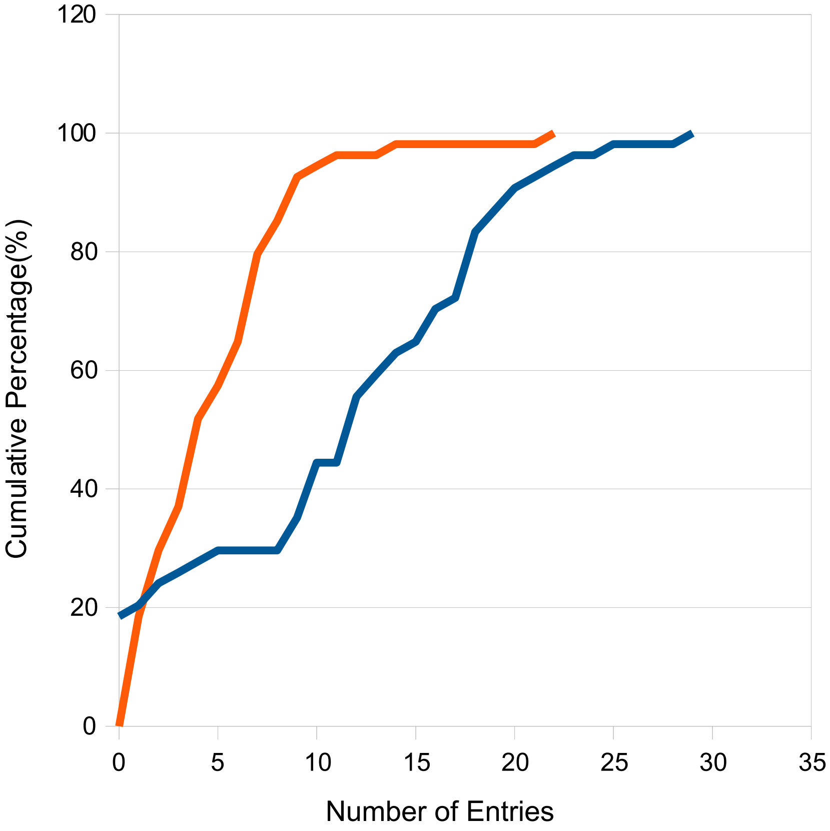

Figure 5.6: Comparison of the number of first hops that allow full and strict reachability for multi-region routing.

It is visible in table 5.2 that the number of entries that allow strict reachability is larger than the entries that allow full reachability. These connections are associated to the massive number of *p*2*p* inter-region links that traverse tiers, observed on figure 5.1. HLP and NIRA would likely fail in such scenario since they compute shortest-path trees for each hierarchy; whereas this thesis' protocol only computes paths for the router's region.

With the introduction of regions, we should expect some trade-offs in terms of multipath. With a massive multi-homed topology, it is foreseeable a vast number of available paths. However if we partition the topology from figure 5.1 as two regions, the number of intra-region paths is reduced. The chart from figure 5.7 shows the cumulative percentage of $C2P$ and $P2Patt$ intra-region paths for a single region and for two regions. These links allow full reachability to other regions. They define the maximum number of paths that can be used to connect to an external region. It is noticeable that the number of paths was reduced for two regions, nonetheless this difference it is not so significant. The results from figure 5.5 show that we have achieved better scalability in terms of routing entries.

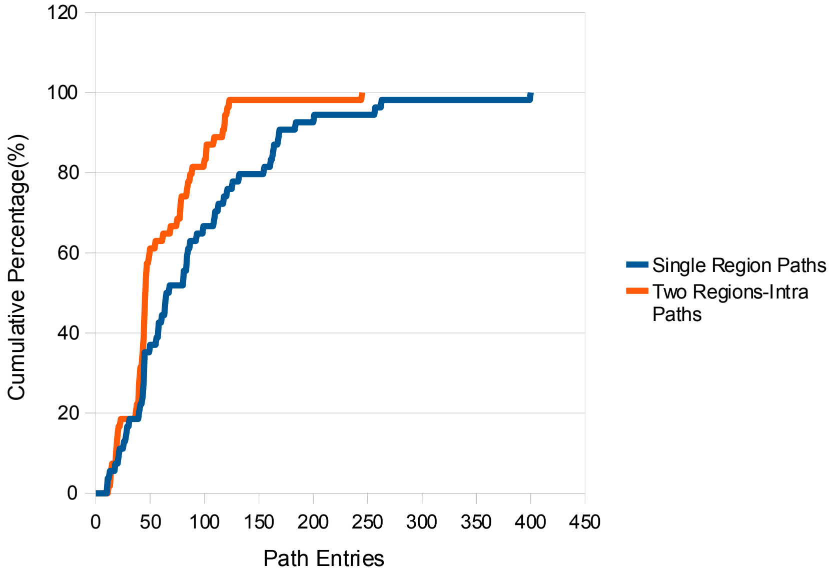

Figure 5.7: Comparison of available intra-region paths for a single region in blue and for two regions in red.

5.3.2 Packet signalling analysis

This section analyses the performance of the proposed protocol in terms of packet signalling overhead. Section 3.4 stated that routing messages should be contained on the region that originated them; except for inter-region links' state changes that are notified to both regions.

For this experiment, 70 link failures were randomly picked. At the inter-region level, we have divided these failures in two groups: 37 of them are intra-region failures and 33 are border-link failures. For each isolated failure we have registered the number of affected ASes, *i.e.* ASes that were warned with a routing message.

We have used the BGP++ implementation for a fair comparison between our protocol and BGP. Business relationships on BGP++ were configured with the aid of the *community* attribute, as exemplified on section 2.3.2. These relationships follow two basic rules:

1. No traffic is forwarded from one provider or peer to another provider or peer;

2. Customer routes are preferred over peer or provider routes.

The chart from figure 5.8 presents the cumulative percentages of the affected ASes for various scenarios:

1. Link failures for a single region with DTIA (DTIA 1);

2. Link failures for a single region with BGP (BGP);

3. Intra-region failures for two regions with the proposed protocol (DTIA 2*i*);

4. Inter-region failures for two regions with the proposed protocol (DTIA 2*b*);

Regarding the experimented topology as a single region, the DTIA protocol scales well compared to BGP. DTIA achieves faster routing convergence, since it has an high probability of finding an alternative path with the same preference, according to the data of sections 5.2 and 5.3.1 and the rules from section 3.3.3. BGP does not converge as fast as DTIA, since it announces a new route every time the *best* route to a given destination d is altered.
It is also noticeable in figure 5.8 that some link failures on the current topology might warn up to 49 ASes with DTIA; this represents a large fraction of the current topology.

The figure shows that partitioning the topology in two regions, improves the containment of control packets using the proposed implementation. For intra-region failures, routing messages are contained at the region that originated them; at most half of the experimented topology is warned.

Regarding inter-region failures, we have the *risk* of warning both regions affected by the inter-region link. Although, if we had three or more connected regions, an inter-region

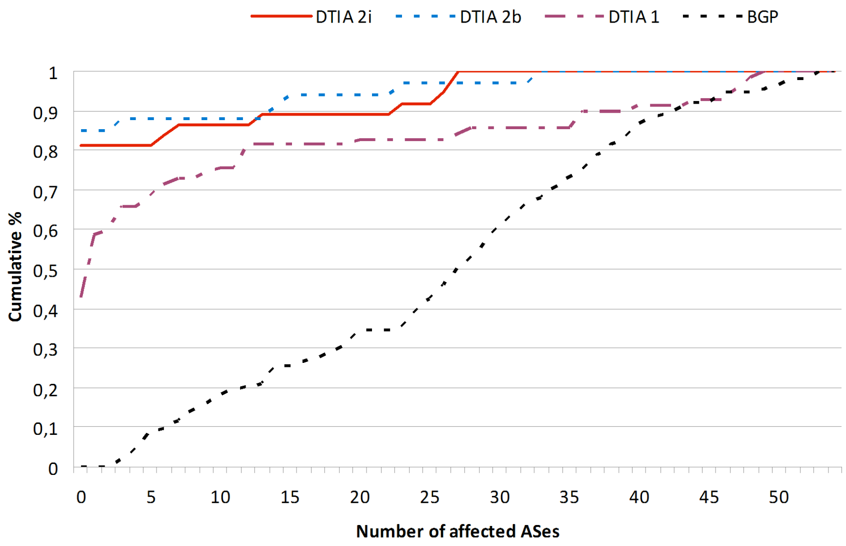

Figure 5.8: Cumulative percentage of affected ASes. DTIA 1 presents the results for a single-region; for two regions DTIA 2i presents the results for intra-region links and DTIA 2b for inter-region links.

failure would only trigger routing messages at both regions affected by the failure. The obtained results show that the probability of finding an alternative path with the same preference as before the inter-region failure also substantiates the obtained data, according to sections 5.2 and 5.3.1.

As a final signalling analysis, we have also experimented node failures, *i.e.* all node's links fail. Figure 5.9 shows the cumulative percentages of warned ASes in a single region and for two regions. From figure 5.9, we can notice that between 19 to 51 affected ASes, the topology has better containment of routing messages for two regions. Outside of this interval, DTIA and this thesis' proposed implementation are approximately the same performance.

A node failure inside a single region is supposed to affect at most all ASes. Subsequently on a multi-region topology, a node failure either warns the node's region or all regions that are connected to this node. Recalling the topology characteristics of section 5.2, inter-region links are a significant portion of the total number of links. This fact supports our results; if we have a *crucial* node X_z that allows full reachability inside and outside of its region, it is natural that X_z's failure affects more than one region. However, some of the failures of a border-AS were hidden by an alternative path to another border-AS with the same AS signature. We can conclude from figure 5.9 that partitioning a topology in

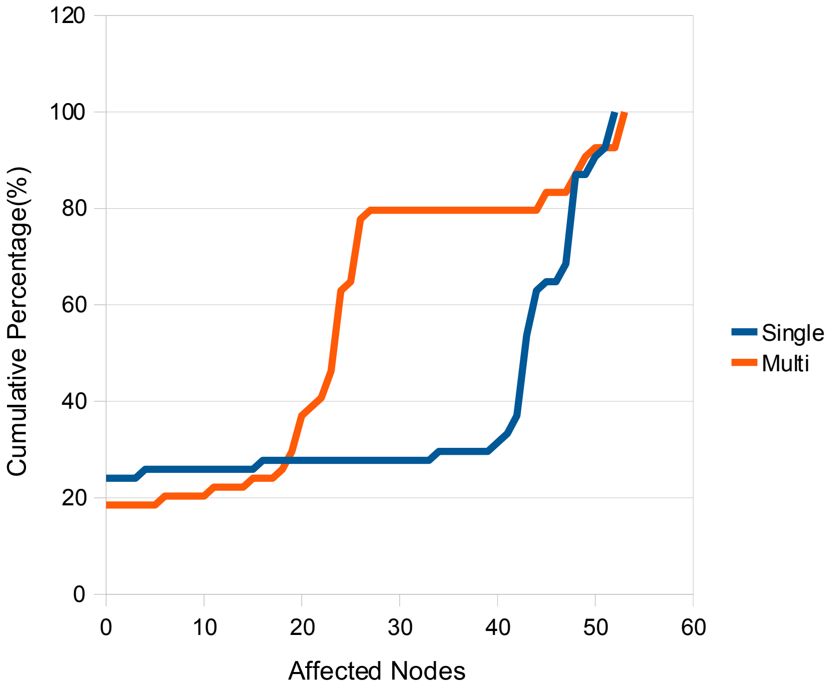

Figure 5.9: Cumulative percentage of affected ASes. The blue curve presents the results for a single-region, as for the red curve it presents the results for two-regions.

several regions is a powerful *tool* to limit packet signalling; however if a node failure occurs, it is possible that more than two regions are warned. If we are planning to partition a topology into several regions, then we need a careful groundwork; regions should have enough redundant links to assure better containment of control packets.

5.3.3 Packet delay analysis

The definition of an inter-region protocol brings advantages in a partitioned topology in terms of routing scalability and containment. Nonetheless we should foresee some trade-offs with the introduction of regions; data packets could traverse more hops to a remote destination than they would on a single region.

The current section analyses the number of hops that data packets traverse to reach their destination; a comparison is made between the single region DTIA and multi-region DTIA cases.

For a fair comparison we have randomly picked 50 node pairs, where each pair is composed of a source and destination from different regions. This experiment registered the number of hops that a packet p needs to reach its destination, for each pair source-destination.

The chart from figure 5.10 presents the cumulative percentages for the single region and multi-region cases.

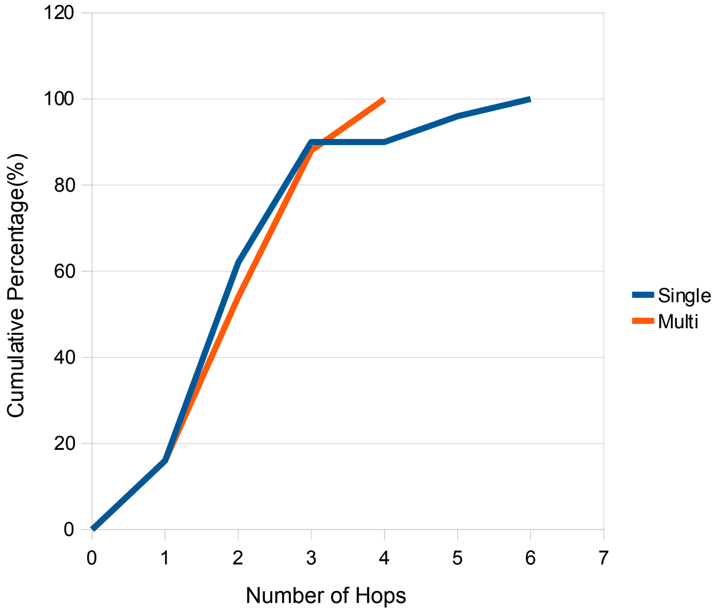

Figure 5.10: Cumulative percentage of the number of hops. The blue curve presents the results for a single-region, as for the red curve it presents the results for two-regions.

From figure 5.10 it is noticeable that both cumulative percentages are practically similar from 1 to 3 hops; however for the multi-region case we have at most 4 hops, whereas for the single region case we have 6 hops. These differences result from the routing rules that differ in both cases: packets do not follow descending paths at the inter-region level. Due to the massively multi-homed scenario, tier-1 providers provide full reachability to any destination within a distance of 2 hops between the source and the destination [ABP09]. Therefore, some routes get shorter when inter-region routing is used. Nonetheless we have observed a small degradation in terms of traversed hops, close to 6% of the examples; whereas 92% did not suffer any degradation.

If we had considered a pure hierarchical topology, the degradation of traversed hops would be noticeable. A pure hierarchical structure does not have peering links traversing tiers; subsequently the non existence of $p2p$ links reaching remote tier-1 providers would worsen the number of traversed hops at the inter-region case.

Chapter 6

Conclusions

The current chapter resumes this thesis' conclusions based on previous chapters. Section 6.1 contains a small synthesis and section 6.2 states this thesis' final considerations. Section 6.3, it enumerates a few topics for further work.

6.1 Synthesis

This section briefly describes each chapter's content. On section 1.2 the main hypothesis of this thesis was formulated. The following chapters validate the formulated hypothesis.

On chapter 2 we have discussed the structure of the Internet and the various opinions that characterize it. A brief discussion of the state of the art was made; current standard protocols and academical solutions were compared. We have analysed the *pros* and *cons* for each solution and taking into account Internet's structure. From this analysis we have identified important characteristics for this thesis' proposed solution on chapter 3. A brief description of the DTIA protocol is made on this chapter; its rationale was an important foundation of this thesis. An inter-region solution is proposed as the main contribution of this thesis. The proposed solution maintains the current Internet business model and restricts a set of policies to ASes connections. Furthermore, it is suggested a gradual deployment of our solution to replace BGP.

Chapter 4 overviews this thesis' implementation, through the visual aid of flowcharts. Further details of the DTIA protocol and of the proposed architecture were also included. The changes made to the ns-2 simulator classifier module were also presented For further validation of the proposed model, chapter 5 describes the experimented topology on the ns-2 simulator, and compares the obtained results with the BGP and DTIA protocols.

6.2 Conclusions

Chapter 2 introduced the standard protocol for inter-domain routing, BGP. BGP usage on the Internet limits the Internet's future evolution; the protocol fails to adapt in terms of routing convergence, since not all ASes apply coherent policies . BGP was thought as a reachability protocol, however ASes' administrators often configure BGP routers for other purposes outside of routing.

Two proposed solutions, NIRA and HLP, tried to replace BGP by defining the concept of inter-region routing; this concept is supposed to restrain the scale of routing algorithms. However both solutions failed to implement an architecture well adapted to the current Internet business model; these academical proposals assume the Internet as a pure hierarchical structure. Chapter 2 discussed several perspectives of the Internet's structure. CAIDA's studies revealed that the Internet does not follow a pure hierarchical structure; yet these studies show that the Internet's properties resemble a scale-free network.

The importance of the Internet's structure is a crucial theme to define a new routing architecture. DTIA takes into account the current Internet's structure. According to chapter 3, it defined a modular architecture that separates different functionalities as opposed to BGP; furthermore, DTIA's failure management algorithm improves the containment of routing messages. However it failed to scale with large topologies because it missed an inter-region routing protocol.

We have proposed a new solution that extends DTIA's work, adding further support for inter-region routing. This proposal improves DTIA's scalability, since the introduction of the multi-region concept helps to conceal information from each region; furthermore routing advertisements are contained to the region where they are originated. Regarding inter-region failures, routers just warn the affected regions. This proved to be a powerful tool for fast routing convergence; the results of chapter 5, confirmed the feasibility of the proposed implementation.

With the current Internet scenario, it is imperative to deploy a scalable solution that gradually replaces BGP without disrupting the Internet. This thesis proposed a deployable solution that places small DTIA islands interworking with BGP as an external region; as years progress BGP might be slowly replaced with an inter-region solution.

6.3 Future work

During the development of the proposed solution, some assumptions were made to support our architecture model. Our model proposes inter-region routing based on an AS's identifier; subsequently, we need a service that performs the mapping between an AS's prefixes with its respective AS number and region. The DNS service could be extended with this functionality. Furthermore, to perform the path exploration algorithm, we need a *standard* distribution service that delivers the network graph to routers; RIPE's database could be a step forward on that direction for an European region.

DTIA's modularity allows the proposed architecture to be further extended. DTIA separates orthogonal functionalities, such as reachability and routing; we could improve this architecture for inter-region routing by adding other modules; for example, we could perform load balancing based on the data of the reachability and routing layers. A deployable solution that interworks with BGP was also proposed; such functionality could be added as a new module.

If these changes are further developed in the future, we could turn a new page on interdomain routing.

Appendix A

DTIA: Routing at the Interdomain level

Next page presents the technical report that explains the routing module of DTIA's architecture.

DTIA – Routing at Inter-Domain Level

Pedro Amaral, Luis Bernardo, and Paulo Pinto, *Member, IEEE*

***Abstract*—This manuscript describes an inter-domain routing architecture called DTIA (*Dynamic Topological Information Architecture*) which aims at replacing BGP without creating a disruptive reality. DTIA separates various aspects having a layered approach to the problem: it begins with reachability, then routing, and finishes with traffic engineering. This paper is about the second aspect. Our approach was to select relevant BGP features that should be part of the architecture and construct the routing protocol. Other features will be handled at higher level. One major requirement has been not to change IP packets and the commercial relations in the Internet. Autonomous Systems (ASes) receive a network map and they only exchange signaling about failures. They perform routing based on link types (provider-costumer, peer, primary, backup, etc.) and routing rules, defining a closed system. We show that this system is monotone guaranteeing convergence of the routing protocol and creating a multipath system with very little overhead. DTIA routes packets using AS identifiers instead of network prefixes requiring a mapping service between them. The separation between reachability and routing provides some advantages being one of them the reduction of algorithm complexity. We use "regions" to cope with scalability and the reduction of algorithm complexity allows us to have quite large regions.**

***Index Terms*— BGP; convergence; inter-domain routing; policy routing; scalability**

I. Introduction

THE current protocol for inter-domain routing, BGP (Border Gateway Protocol), is a backbone of the current Internet. Therefore, any replacement or even any changes to it is a very sensitive matter. However, over the years several weaknesses and inefficiencies [1] have been identified and most of them will get worse with time. An accelerating element, which is the focus of this paper, is multihoming. More powerful ways to take advantage of it can be devised than the ones provided by BGP. Multihoming brings the possibility of multipath routes which is a feature not covered by the basic BGP.

BGP is fairly simple and very flexible. It uses prefix-based routing and the flexibility in using the attributes allows very precise manipulations prefix by prefix for common routing aspects making it highly tuned (and tunable). Examples are attribute manipulation (AS Path prepending, local preferences, the MED attribute to suggest preferred routes), prefix aggregation or de-aggregation, use of communities, etc.

The flexibility is such that manipulations started also to be used for aspects marginally related to routing or even quite outside. Examples are the definition of backup links with behaviors rather complex and dependent on the topology or the type of link (if the link is between a customer and a provider, or between peers at stub level or at provider level, etc.), or the construction of prefix-based VPNs.

The current reality is a complex system that is highly sensitive to the coordination and simultaneous implementation in all Autonomous Systems (ASes) in a region [2]. Most of the times manipulations take into account the precise topology, and the overall system becomes unstable when a failure happens instead of showing adaptation.

This paper proposes a new vision for inter-domain routing that still preserves the more important features of the current Internet. It has four main characteristics: a) a three layer approach of concerns: reachability, routing, and traffic-engineering; b) highly adapted to multihoming and multipath; c) attribute manipulation is replaced by a set of rules; d) the network is considered a static network and solely link failures produce dynamism.

The most novel aspect about this work is a new systematic view of the inter-domain routing problem based on the innovative use of several techniques reported in the literature.

We should keep in mind that any new solution for inter-domain routing cannot feature all the facilities available today in BGP and still remain simple. Some features have to be considered secondary and be performed in other ways. The difficulty is the identification and agreement amongst the community on which features should be considered primary and secondary.

This paper builds on [3] that presents the inter-domain reachability protocol. We based our architecture in three main assumptions and considered four design choices. If we accept them our architecture, described in section IV, becomes very simple. Section V onwards describe reachability (very briefly), routing in the absence and presence of failures, how we envisage the deployment, the related work and some experiments to prove the feasibility of our choices. A general assessment is made before the conclusion section.

II. Main Assumptions

The three main starting assumptions to construct the architecture are: the maintenance of the current business model based on Autonomous Systems (AS) and Internet

Manuscript received October 9, 2001. This work was supported in part by the Portuguese Science and Technology Foundation (FCT/MCTES) under the grant SFRH/BD/44476/2008.

All authors are with the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 CAPARICA, PORTUGAL; Pedro Amaral, phone: +351 21 294 85 45; fax: +351 21 294 85 32; pfa@fct.unl.pt.

Service Providers; the mix hierarchical/peer structure of the current Internet; and the long lasting and reliable characteristics of the inter-AS links.

Given the current economic and social importance of the Internet it is unlikely that an architecture based on a different business model will be quickly adopted by the active players. Our purpose is to define a system that changes the business model as little as possible but will be able to evolve in the future, instead of proposing disruptive business models.

Back in 2000 a hierarchical structure based on customer-provider relations and forming a three-tier structure could be clearly identified [4]. At that time there was already a large percentage of multihoming, and peering at provider nodes. Over the years the Internet became more richly connected [5] with: a higher number of direct links between ASes (blurring the three tier structure, creating an even higher degree of multihoming, and enlarging its breadth instead of its depth); and a strong peering at regional level (US, Europe, Asia). It is as if relatively lower ASes in the hierarchy prefer to connect directly to other ASes with whom they exchange large amounts of traffic. This creates a web of links that might, or might not, be used by others depending on the type of link (more precisely the advertisement that is made). BGP is unable to fully exploit such a network, both in terms of multi-path and even for the case of backup. Multi-path needs advertisement of more than just the best path, and every backup has to be highly tuned in topological terms not to become the first choice.

The links between ASes are pretty stable over the time because they are based on business relationships. Any changes happen in a controlled manner. The time sensitive issue is whether the link failed or not, and not so much if it exists or not. In terms of physical reliability the reality today is also quite different from the past. ASes are connected inside of a room in a much protected environment. It is not uncommon that an organization places a router in the room (sometimes in another continent) to connect to other ASes. The consequences are that intra-AS failures are more probable than inter-AS failures strengthening the argument that inter-AS connections are stable (intra-AS failures can be solved differently, and probably more easily than inter-AS failures. Obviously one failure can lead to the other).

III. Our Design Choices

Routing is based on AS connections and not on prefixes.

The fact that the BGP is prefix based has several consequences. Given the fact that each time only the best route is advertised the end result is the construction of several graphs (per prefix) over the physical links. Therefore, the knowledge of the physical topology of the network is not a first class concern for BGP (although it can be inferred [4]). As the attributes are also based on prefixes the routing behavior (the actual graphs) can be very different in a region making the system very complex and hard to manage. Therefore, it is not easy to use the topological information to accelerate convergence when transient failures happen.

Working at prefix level enlarged the size of the routing tables and it is consensual that this growth must be contained [6]. One way to reduce the growth rate is to rely on prefix aggregation. In architectural terms this will not work because BGP is really based on prefixes and they are the knots to change behaviors. For instance, traffic engineering and load balancing can be based on separating flows (prefixes) that belong to an AS, enlarging the routing tables. Note that performing these tasks using prefixes is quite inefficient because traffic for a prefix can change over the time. It is a rough solution to the problem highly suited to the characteristics of BGP. The use of multihoming makes aggregation even harder: consider an AS getting its prefixes from provider 1 and having other *n* providers. Every provider but provider 1 cannot aggregate the prefixes. Even provider 1 may not want to aggregate – if it does it might get no traffic because more specific longest match paths are preferred.

Routing with AS labels provides a significant reduction of the routing table, given the number of ASes. This decision is controversial with some opinions against it [7] and others in favor [8]. It brings further advantages: traffic engineering and load balancing can be performed amongst ASes providing a more efficient solution based on a single graph compared to the prefix solution; multihoming is reduced to a choice of paths and ASes without any consequences to the size of the routing table. Two problems exist: 1) packets can follow different paths with different transit times making it necessary to adapt the congestion control algorithm of TCP (the calculation of the Round Trip Time becomes more complex and the reaction of TCP to the reception of a number of packets out of order must be reconsidered); and 2) a mapping between prefixes and ASes must exist.

We assume that there is a service to map prefixes to ASes. This service can support host multihoming. It can also support mobility in terms of prefix assignments to ASes to cope with mobility requirements seen in military networks.

A set of rules replaces prefix manipulation.

BGP uses attributes in the UPDATE packet to describe the prefix characteristics. UPDATE packets received go through a filtering process and can have their attributes manipulated before their route is placed in the routing table. Routes in the table suffer a similar process (filtering and manipulation) before being sent to neighbors in UPDATE packets. The attribute manipulation provides most of the flexibility of BGP. Over the years attributes have been used to produce specific effects on routing enriching the ways ASes interact. But a high degree of coordination is needed in their implementation often with table meetings between AS representatives.

This collateral damage in convergence is due to the expressive freedom on attribute manipulation. Firstly, over the times the attributes started to be used for other purposes than the ones they were designed for, creating a cumbersome system (for instance, prepending AS number in the AS Path [9], or using the community attribute to define VPNs [10]);

and secondly because some techniques make use of highly expressive semantics providing freedom on establishing rules, producing a large scope of intervention and difficulties in living without them (examples are the usage of regular expression manipulation on the AS Path, or the meaning of the community attribute numbers that are not standardized and can be anything an AS wants [10]).

Moreover, BGP should be a protocol able to learn prefixes dynamically and act accordingly. If we look closer, the attribute manipulation destroyed this feature and some relevant manipulations assume a complete knowledge of the topology of the network in the region. There are many examples mainly involving AS prepending and multihoming. Some of them are: a) consider an AS with two providers and providers of these providers. In order to make load balancing the stub AS has to know the path until a NAP (Network Access Point) (or a common AS) in order to know how many times it should prepend the AS Path; b) the same arguments for the choice and meaning of numbers for the community attribute when used to achieve AS Path prepending; c) in multihomed scenarios prefix aggregation can completely drive away traffic if we do not take into consideration how prefixes are advertised through the other branches; d) consider the situation of two AS providers having each one a different stub AS client and a backup link between these clients. In order for each provider not to use the backup link to forward traffic to the other provider's client, local preferences must be carefully assigned and the knowledge of the topology is necessary. Configuring the system so tuned to precise topologies can make it unpredictable when links fail.

Clearly BGP got a life of its own in the sense that the mechanisms that were defined were extended to perform new features. This is tremendously flexible, powerful, and demands great expertise from the engineers performing network tasks. The alternative is to define a closed system in the form, for instance, of a set of rules that can describe the most important features of inter-domain routing subject to the previous main assumptions. Trying to compete and replace the existing system will be challenging. This paper tries to contribute to this challenge. But, as it was stated at the beginning, it is impossible to have all BGP features and still remain simple.

Routers get a static map of the network and co-operate to learn about failures

What has been learnt over the years is that most of the routing events on the current Internet due to dynamic changes come from the prefix advertisement part (something we do not have). The other source of events is link failures and most of the times these failures are not at higher levels of the hierarchy containing the disturbed area.

On the other hand, we have seen that the structure of the network is highly static due both to the legal nature of the relations between ASes and physically by the way ASes connect to each other (most of the times inside of a room). Therefore, an algorithm for inter-domain routing should focus more on handling the dynamic part of the network (caused by the seldom failures) and not so much on discovering the graph of the network. Moreover, the algorithms for the dynamic part should be light and the general algorithm should focus more on enabling traffic engineering features. These are exactly the opposite characteristics of BGP (heavy mechanism for graph definition and dynamic management).

We assume that a central entity (possibly replicated) delivers a static map of the network (or a region, see the following design choice) to routers. There is no guarantee that the static map is the real picture of the network due to failures. Nevertheless, all routers know the same information and can act upon it. The protocol assumes a static reality and builds a dynamic reality due to failures. This approach was followed with different purposes by [11]. As there is no need to discover the graph, the traditional routing paradigms do not apply (distance-vector, path-vector, and link state) and the dynamic part of the protocol is simplified in terms of messages exchanged. The major problems to solve are to warn routers about failures, re-route data packets that encounter a failure, and warn routers when the failure is solved. The dissemination of failure information should only disturb the relevant routers with precise rules about its scope.

Maps and co-operations are limited to regions

Most of the concerns in inter-domain routing are local to the ascending (and descending) paths. Real global events in BGP are again related to the withdraw procedure of prefixes. Depending on their placement in the hierarchy and what aggregations exist, events in BGP are confined to regions.

BGP does not provide much help for the definition of a region due to the multiple graphs it constructs over a set of ASes. HLP [8], for instance, proposes the concept of a tree based on the customer-provider links and one hop peer-to-peer links to confine their algorithms. Due to the heavy use of multihoming at middle levels this concept can become complex with routers belonging to too many trees.

We propose a more rigid approach: a region is a set of ASes with a few restrictions and for each region the static graph is constructed and delivered to routers. Nowadays RIPE has already an embryonic database that can be used for this purpose[1]. This database [12] stores all policies of the European ASes. Its format is not suited yet for our purposes but it can be a first step. For our experiments we used a topology from the CAIDA AS Relationships Data research project [5], and the method described in [13] to infer relationships. A concrete definition of a region is given below.

IV. Architecture

A routing protocol has usually two components:

- Mechanism – defines how routes are known (e.g. distance vector) and defines a route selection algorithm

[1] It is used already by providers to verify prefixes advertisements from their clients (via filters).

(down to one route or more using e.g. Dijkstra's shortest path).

- Policy – defines the link characteristics (attributes or metrics); it has direct consequences on the route selection algorithm.

BGP is a prefix policy based protocol meaning that the policy component has also direct consequences on the definition of routes (per prefix).

As stated above, DTIA has a three layer approach. The first two layers cover most of the BGP characteristics (and the two general components above). The third level covers some remaining characteristics (e.g., controlling incoming traffic). and other traffic engineering issues but it is not addressed in this paper.

The reachability protocol [3] objective is to calculate the set of all valid paths from one AS X to any other AS inside the region, *r*, and to other regions, denoted as $P_r(X)$. Each valid path is valley and loop free (valley free means that a packet arriving from a provider cannot be sent to another provider). There can be more than one path to a destination (especially due to multihoming) providing a base for multipath routing.

A valid path is one that complies with the policies. The policies are applied at AS level and not at prefix level. DTIA covers the so-called *common policies* [14] extended with two extra relations. The *common policies* comprise the provider-customer and peer-to-peer relationships that are enough to deal with 99% of the relations used today in BGP [4][8]. The extra two are used for sibling like relationships and backup specific relations (as suggested in RFC 1998). We replaced the BGP advertisement algorithm (where policy rules influence the advertised routes) by labels on the links between ASes and a small set of rules for validating paths. The result is a stable and robust base upon which more complex algorithms can be built. By robust we mean that the known consistency problem of BGP is solved by having the same set of rules networkwide.

In general terms, different routing protocols can then be defined on top of $P_r(X)$. DTIA proposes one multi-path routing protocol, which calculates $R_r(X)$, a subset of $P_r(X)$. By featuring multi-path, $R_r(X)$ can be further used to implement traffic engineering and load balancing in the layer above.

This separation is crucial because it reduces the computational complexity of the protocols. Our labels and set of rules allow us to build $P_r(X)$ in the forward direction in a very simple way (BGP advertisements work in the backward direction from destination to source). $R_r(X)$ is then built from $P_r(X)$. This reduction of complexity allows the dimension of regions to be very large.

V. Reachability

The reachability protocol is reported in [3]. This section presents a brief overview.

A region graph is built by an entity (e.g. RIPE for the European region) and distributed to all nodes (ASes) of the region. Each time a new graph is generated an increasing sequence number is assigned to it. The graph *G(V,A)* is modeled as a directed graph with *V(G)* vertices that model ASes and *A(G)* arcs that model links between ASes. The arcs are labeled according to the commercial relationships between the ASes. We consider four types of inter-AS relationships:

Provider-Customer. One AS (the provider) accepts all traffic from the other AS (the client). Two arcs are considered: one in the provider-customer direction (*p2c*) and another in the customer-provider direction (*c2p*).

Peer-to-peer. ASes provide connectivity for their direct or indirect customers. No transit traffic from the peer is allowed. There is one arc in each direction (*p2p*).

Peer-to-peer allowing backup. The same as before but allows transit traffic if no other path exists. There is one arc in each direction (*p2pbk).*

Peer-to-peer allowing transit traffic. Transit traffic is allowed in any situation (this is not very usual but exists in the RIPE database). There is one arc in each direction (*p2patt*).

Based on these link labels the set of rules showed in Table I and II were defined to validate paths and construct $P_r(X)$. Basically the algorithm performs path exploration following all ascending (*c2p*), descending (*p2c*), and horizontal (*p2p*, *p2patt*, *p2pbk*) paths in a hop-by-hop process in the forward direction. To control valley paths a qualifier, named Direction (D), is added to each path. Direction is set according to the type of the first arc: if it is *c2p* D is set to 1; if it is *p2c* D is set to 0. If it is *p2pbk* or *p2patt*, two paths are considered: one with D=0 and another with D=1. Further processing will invalidate one of them. If it is *p2p* only the D=0 is considered.

Peer to peer arcs pose extra problems in terms of guaranteeing no loops for the paths. To solve them whenever such an arc is followed the departing AS number is recorded in an AS set for that path. Whenever an AS is reached using such an arc it is verified that this AS is not in the set.

The value of D can change in the course of the path exploration. A descending path (D=0) never changes to an ascending path (no valley paths are allowed). An ascending path is changed to a descending path when the first arc of type *p2c* occurs in that path.

Fig. 1 shows the exception case when a path begins with an arc of type horizontal. The process is running on B. Two paths are set to C, and again to D. Both C and D are included in the AS sets of both paths. When going to G the path with D=1 is

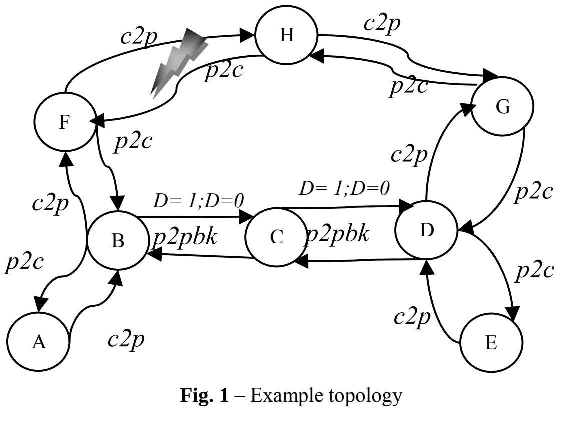

Fig. 1 – Example topology

valid and the other is invalidated (cannot follow a *c2p* arc). When going to E the D=0 path is valid and D=1 is invalidated (a *p2pbk* arc cannot be followed by a *c2p* arc).

Fig. 1 also shows how the link *p2pbk* works. AS A is connected to AS H. All traffic flows through A-B-F-H. If link F-H fails, then traffic can flow through A-B-C-D-G-H.

Tables I and II contain the validity rules (valid (V) or invalid (X)) for an arriving arc in the row and a departing arc in the column. Table I is used for paths with D=0 (descending paths). In a descending path *c2p* arcs and *p2p* arcs are always invalid. Table II is for the D=1 case (ascending paths). In an ascending path when the first *p2c* arc appears the Direction changes its value.

Table I – Rules to validate paths for D=0.

Result	*p2c*	*c2p*	*p2pbk*	*p2p*	*p2patt*
p2c	V	X	V	X	V
c2p	-	-	-	-	-
p2pbk	V	X	if(AS in set)X else V	X	if(AS in set)X else V
p2p	X	X	X	X	X
p2patt	V	X	if(AS in set)X else V	X	if(AS in set)X else V

Table II – Rules to validate paths for D=1.

Result	*p2c*	*c2p*	*p2pbk*	*p2p*	*p2patt*
p2c	-	-	-	-	-
c2p	V;D=0	V	V	V	V
p2pbk	V;D=0	V	if(AS in set)X else V	X	if(AS in set)X else V
p2p	V;D=0	X	X	X	X
p2patt	V;D=0	V	if(AS in set)X else V	X	if(AS in set)X else V

In [3] we prove the following theorem:

Theorem 1: *Assuming that*

There are no cycles in the provider-customer relationships[2].

A valid path between two AS in the region has no loops.

Note that distributing the graph and performing the path validation has the same end result of a link state protocol (i.e. all ASes have the entire topology of the region).

VI. Routing

The routing protocol cannot use all the paths in $P_r(X)$. Although each $P_r(X)$ has loop free valid paths, the entire system (all $P_r(X_i)$ considered together) does not form a loop free system for two reasons: as this is a multi-path system one path can conflict with another causing a loop; and even if this aspect is handled, if a link failure occurs similar conflicts can still happen.

To solve the problem a ranking mechanism was defined to classify the valid paths and a management algorithm was designed to handle failures. The ranking mechanism works as a cost for the path and uses a discrete space (as opposed to continuous). Paths having the same ranking value are treated similarly, thus providing the multi-path feature to the protocol.

The ranking mechanism is supported on four preference rules – two already well-known in the current Internet and two due to our extensions (the exact meaning of the path qualifiers in rule 4 will be obvious further down):

1. No traffic is forwarded from one provider or peer to another provider or peer.
2. Customer routes are preferred over peer or provider routes.
3. Primary paths are always preferred to backup paths.
4. Amongst primary paths, *P2Patt* and *P2C* paths are preferred (with equally value) to *P2P* paths. *C2P* paths have the worst preference.

Applying these rules to $P_r(X)$ has two effects: some valid paths are not considered for routing purposes, and all selected paths are ranked. We will prove that if each AS uses paths within the highest ranking value available at the forwarding moment the routing algorithm converges and the packets reach the destination AS without forming routing loops.

Note that by using this ranking mechanism our algorithm actually behaves like the algorithm of a policy based Path Vector protocol, choosing routes according to their attributes and established preference, although having the entire network map. DTIA's routing protocol is in fact a Local Simulated Path Vector (LSPV) protocol [15].

A. Protocol Correctness

Informally a protocol is correct if in a stable network (with no changes occurring) it determines a set of loop free paths between every pair of nodes that have connectivity according to the policy. In order to prove the correctness of DTIA's routing protocol we use the concept of a routing algebra based on the one in [16]. As DTIA's routing protocol is an LSPV one, the algebraic property to ensure correctness is the same as for Path Vector protocols [15].

A routing algebra A is a tuple $A = (\Sigma, \prec, \oplus, L, \phi)$. Σ is a set of *signatures* that qualify paths, $\prec$ defines a *preference relation* over signatures (e.g., with $\alpha \prec \beta$, α is preferred), L is the set of *labels* associated to links, $\oplus$ is a binary operation that maps a pair (label, signature) into a signature and will be used to obtain path signatures, and ϕ is the special signature to denote invalid paths.

We defined L = {*p2patt, p2c, p2p, c2p, p2pbk*} and Σ = {ε, *P2Patt, P2C, P2P, P2Pbk, C2P,* ϕ} ∪ {*BKP* × N^+}. The ε signature is the initial path signature when there is only the node where the path ends. The other signatures look quite similar to the link labels/types. Each AS works on $P_r(X)$ and uses Table IIII to calculate path signatures using the operation $\oplus$ (a link of type l is appended, in the direction to the source, to a path with a certain signature resulting in a new path signature). For instance, consider the grey cell in Table IIII with solid borders. The meaning is that a path with a signature *P2P* can be extended in the direction of the source by a link *c2p*. The signature of the path becomes *C2P*. This represents a packet travelling in ascending direction (to a provider) followed by a path with *P2P* signature. Looking at the column in the Table III we can see that this link is the only valid link to be appended to a *P2P* path.

[2] I.e. no domain is a provider of one of its direct or indirect providers assuming that peers are also indirect providers.

Table IV shows the preference order for the signatures (the values of the ranking mechanism). The higher the more preferred.

A peer-to-peer link that can be used as a backup (*p2pbk*) needs some further clarifications because it can either be used as a regular peer-to-peer link or as a backup link. This has consequences for the preference rules. In the former case the result signature of the path to which this link is appended is *P2Pbk* (and should have the same preference as a *P2P* path). When it is used as a backup link the result is a signature (*BKP,y*) with *y* getting strictly increasing natural numbers as new links are appended. Two concrete examples from Table III for this type of links are:

Backup links used as normal peering: consider the example, *p2pbk* ⊕ *P2C* = *P2Pbk*. It means that a path from a customer is extended to a peer, this is a normal peering relationship and therefore the resulting signature is *P2Pbk*.

Backup links used as backup: for backup paths the resulting signature is (*BKP, y)*. The value of *y* increases every time a *p2pbk* link is used for transit traffic. For instance, *p2pbk* ⊕ *C2P* = (*BKP, 1)* means that an AS can transit traffic between a peer and a provider in a backup situation. The path starts by having *y=1*. For every new link in a backup path the integer is increased. For instance, *p2c* ⊕ (*BKP, y)* = (*BKP, y+1)* means that extending a backup path to a provider is possible but decreases its preference.

A cycle is a sequence of distinct nodes except the first and last (i.e., x_1, x_2, ..., x_{n-1}, x_n with $x_n = x_1$). A cycle is free if at least one of its nodes forwards packets to the destination out of the cycle instead of around the cycle. I.e., at that node the preference for an outer path is greater than the preference for the following node in the cycle.

Consider that the paths around the cycle has signature α_i for node *i*. Consider also that node *i* has *j* other paths to the destination not following the cycle with signatures β_{ij}. Denote $S\ (x_i,\ x_{i-1},\ x_{i-2})$ the signature of the path x_i, x_{i-1}, x_{i-2}. The condition for a cycle to be free is the following:

Freeness of cycles: a cycle x_1, x_2, ..., x_{n-1}, x_n with $x_n = x_1$ is free if there is an index *i*, $2 \le i \le n$ such that $\beta_{ij} \prec S\ (x_{i+1}, x_{i+2}, x_{i+3}, \ldots, x_n)$

Another important property is monotonicity. An algebra is monotone if the preference of a path does not increase when the path is extended with a link.

Definition: An algebra is *monotone* if for all $\alpha \in \Sigma$, and for all $l \in L$, $\alpha \preceq \alpha \oplus l$

A stronger property is strictly monotonicity, in which case adding a label to a path must decrease the preference of the path. In [16], the following two theorems are proven:

Theorem 2: *In a free network, the path-vector protocol converges to local-optimal in-trees.*

Theorem 3: *If the algebra is monotone, then the path-vector protocol can be made to converge to local-optimal in-trees whatever the network.*

The idea behind theorem 3 is to break the non-free cycles if they exist, and then the monotonicity of the protocol makes it to converge. We can now state the following theorem:

Theorem 4: *Assuming that the network has no cycles in the provider-customer relationships, then DTIA's routing protocol converges using sets of cycle free paths.*

Proof: We start by showing that DTIA's routing protocol is monotone, but not strictly. Then we have to see that all cycles are free. Finally, if all cycles are free and the protocol is monotone, it converges

We can see in Table III that the algebra is monotone because a path with a certain signature extended by a link/label never results in a path with a more preferred signature. If we analyze each column the result is similar or less preferred than the signature on the first row. But it is not strictly monotonic because some paths keep the same preference when extended and the possibility of a non free cycle exists.

We will prove now that all cycles are free. There are 6 cases (columns of Table III) and we order each column in terms of preference of signatures, underlining the cases where the preference stays the same. For the *P2Patt* signature we have:

$$\underline{p2patt \oplus P2Patt = p2c \oplus P2Patt} \prec p2p \oplus P2Patt \prec c2p \oplus P2Patt \prec p2pbk \oplus P2Patt;$$

So, we can have non-free cycles if the cycles have only links of labels from the set {*p2patt*, *p2c*}. For the *P2C* signature we have:

$$\underline{p2patt \oplus P2C = p2c \oplus P2C} \prec p2p \oplus P2C = p2pbk \oplus P2C \prec \mathrm{c2p} \oplus P2C;$$

We can see that if the path is always extended with links from the label set {*p2patt*, *p2c*} a non-free cycle can be formed. For the *P2P* signatures we have:

$$c2p \oplus P2P \prec p2patt \oplus P2P = p2c \oplus P2P = p2p \oplus P2P = p2pbk \oplus P2P;$$

In this case it is strictly monotonic and all cycles are free.

Table IV - DTIA's ⊕ operation

⊕	ε	*P2Patt*	*P2C*	*P2P*	*P2Pbk*	*C2P*	*(BKP,y)*
p2patt	*P2Patt*	*P2Patt*	*P2C*	ϕ	*(BKP,1)*	*C2P*	*(BKP, y+1)*
p2c	*P2C*	*P2C*	*P2C*	ϕ	*(BKP,1)*	ϕ	*(BKP, y+1)*
p2p	*P2P*	*P2P*	*P2P*	ϕ	*(BKP,1)*	ϕ	ϕ
c2p	*C2P*	*C2P*	*C2P*	*C2P*	*C2P*	*C2P*	*(BKP, y+1)*
p2pbk	*P2Pbk*	*(BKP,1)*	*P2Pbk*	ϕ	*(BKP,1)*	*(BKP,1)*	*(BKP, y+1)*

Table III – order of preference

ε
P2Patt = *P2C*
P2P = *P2Pbk*
C2P
(*BKP*, 1)
…
(*BKP*, n)

For the *P2Pbk* signature we have:

$$c2p \oplus P2Pbk \prec p2patt \oplus P2Pbk = p2c \oplus P2Pbk = p2p \oplus P2Pbk = p2pbk \oplus P2Pbk.$$

Like for the *P2P* case there are no problems here. In the *C2P* signature we have:

$$\underline{p2patt \oplus C2P = c2p \oplus C2P} \prec p2pbk \oplus C2P \prec p2c \oplus C2P = p2p \oplus C2P;$$

Non free cycles can be formed by links with labels belonging to {*p2patt*, *c2p*}. Finally for the (*BKP,y)* signature we cannot write a relation as the ones above because each valid extension creates a path with a less preferred signature due to *y*.

If *y* was not defined, cycles containing links of the set {*p2patt*, *p2c, c2p, p2pbk*} would not be free and this poses a problem to the current internet business model.

In conclusion, a cycle is non-free in the following cases:

a. All its links have labels *p2c*.
b. All its links have labels *c2p*.
c. All its links have labels *p2c* and *p2patt.*
d. All its links have labels *c2p* and *p2patt.*
e. All its links have label *p2patt*.

Stating the other way around, if the network does not contain any of these sequences of links, all cycles are free and DTIA converges. Let´s analyze each one of the five cases.

The a. and b. cases are guaranteed not to exist by the assumption of Theorem 4. Regarding c. (or d.) the *p2patt* links are steps in a descending (or ascending) path of *p2c* (or *c2p*) links. But non-free cycles can still occur in awkward situations: if an AS is a provider of a *p2patt* peer of one of its providers (or if an AS is a client of a *p2patt* peer of one of its clients).

Fig. 2 clarifies the situation: AS B is a client of AS A and has a *p2patt* link to AS C. AS C is a provider of AS A. If this would happen in BGP and if the link B-C was a peer-to-peer the cycle would not be possible because AS B would not export routes learned from AS A to AS C. Also in DTIA if the B-C link is a regular *p2p* link the cycle would not exist since a rule marks a path with a *p2c* link followed by a *p2p* link as invalid. Allowing transit traffic in link B-C is, in terms of routes, similar to making AS B a provider of AS C (*p2patt* links have policy rules that are similar to those of *p2c* links in descending paths, or to those of *c2p* links in ascending paths). The difference is merely economical, with AS B not charging transit traffic. In conclusion, we consider the situations above as being in contradiction with the *no cycles in the provider-customer hierarchy* assumption.

Finally there is case e. In this case a cycle of *p2patt* links is non-free because the algebra is simply monotone. It also makes sense with the current Internet business model. A tie break mechanism must be defined. A simple one is to choose the path with less links (hops). Note that if more than one exists with the same number of hops, DTIA maintains the possibility to use all of them allowing multipath. Comparing to BGP it is just as if various paths were advertised instead of only the best one. To implement the tie break procedure the order of *P2Patt* paths is computed when $P_r(X)$ is built, by counting all consecutive links that have *p2patt* labels.

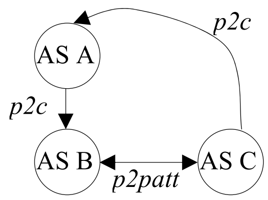

Fig. 2 – An awkward non-free-cycle

B. Implementation

The complexity of the calculations of the path signatures has a direct impact on the number of ASes that constitute a region. Calculations should start at destinations and end at sources (to model the route export process). It is known that the number of operations increases considerably with the size of the region. Going forward on the paths is not possible because violations of the policies are not detected. Fortunately there are two aspects in DTIA that makes the problem tractable: the separation between reachability and routing that produces $P_r(X)$ containing valid paths; and the characteristics of DTIA's algebra that allows the classification of the paths using a kind of forward direction.

Let's start by the algebra and by examining it without the *p2patt* and *p2pbk* labels. This matches the *common policies* of BGP, and corresponds to the grey area of Table III. We can see that a path with a certain signature, extended by a link in the direction of the origin, takes either the signature of this link or the signature ϕ (invalid). Therefore, the algebra is a *local preference* algebra where the signature of a path is defined by the last link. Another interesting characteristic is that as links are being appended to a path the signature of the path maintains the order of preference or decreases. This is due to the fact that our algebra is monotone. In real terms (i.e., taking into consideration the BGP policies) an appended link that raises the order of preference of the resulting path signature is a violation.

A simpler algorithm can then be defined that walks through the path in a kind of forward direction instead of the reverse one: first, consider S_i the signature of a path between two neighbor ASes with a single link of label l_i calculated from the destination to the source (I.e., $S_i = l_i \oplus \varepsilon$). Then, we follow the path starting at the source AS in the forward direction. We calculate the signature S_i for every pair of ASes in the path (in practice all single links). The signature of the total path to a given destination AS, AS *n,* is the least preferred of all the S_i. Let's see what happens using the usual method (extending backwards). We start with S_n and apply ⊕ to each appended link. The preference of the signature decreases monotonically according to the result of ⊕ with the label of the added links. Therefore the total path signature is defined by the link whose result has the lowest preference.

In Fig. 4 (a) we start at AS X and calculate the path signature *C2P* for the path X-A from destination A to source X. We do the same for A-B with result *P2P*. Since *C2P* is the lowest of all S_i, the signature for X-A-B is *C2P* (if we calculate the signature by applying directly ⊕ from B to X the result is the same). The same holds for X-A-B-C.

Why is this so simple? Or in other words, is this kind of forward walk equivalent to the backwards process of export in BGP? The problem here is that by going in the forward direction we can violate BGP rules and not be aware of some path violations. This happens because we are calculating signatures of one-link only paths (which are never invalid). But if we only analyze valid paths this poses no problem. Our distinction between reachability and routing is precious here because $P_r(X)$ has only valid paths and can be analyzed in this forward direction reducing the complexity.

Considering now DTIA's new labels (the entire Table III)

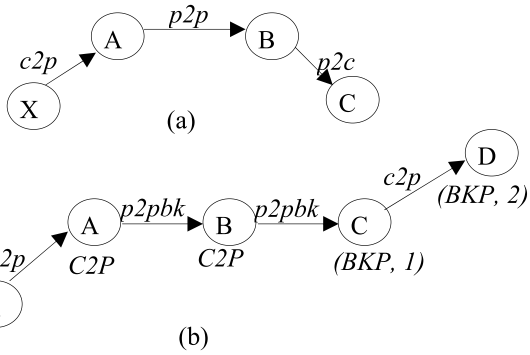

Fig. 4 – Calculation of path signatures
(a) DTIA´s correspondence of BGP's *common policies*
(b) DTIA's extensions

the final signature is still the lowest of the S_i due to the monotonocity of the routing algebra. However, the calculation of each S_i is a little bit more complex because in some situations if we only consider the link concerned we do not get a conclusive result. For instance, a *p2pbk* link can be used as a peer-to-peer link or as a backup link depending on the sequence of links used. To have a definitive result for a S_i it is enough to analyze backwards the sequence of this link with the previous link. The calculation of S_i is now given by $S_i = l_{i-1} \oplus (l_i \oplus \varepsilon)$. For the link departing from the source (the first one) we have $S_i = (li \oplus \varepsilon)$. The algorithm follows like this: we calculate the S_i of a link connecting AS N to AS K; then we compare it with the signature of the path from the source to AS N; take the least preferred and assign it to the path from source to AS K (and record at the source that AS K can be reached by that specific first link with the calculated path signature). Fig. 4 (b) shows an example. Near each AS it is written the path signature from AS X to that AS.

VII. Regions

Regions are used to maintain the scale of the algorithms. The number of ASes in a region should be such that the time to perform the calculations of the paths is realistic. According to our experiments a number just over 11,000 ASes is still manageable. Regions must also have the following two characteristics:

i) An AS in the region must have valid paths to all the other ASes in the region (this is not so drastic because having a provider at a high level solves the problem);

ii) Each region must have ASes that connect to *every other* regions and can route packets following the rules presented previously (this implies that a region must have a tier-1 AS, or an AS in the region has to have a tier-1 AS as its provider).

Characteristic ii) is important to avoid the definition of a protocol for inter-region reachability and routing. A consequence is that the total number of regions must be small, and therefore each one must have a great number of ASes. We envisaged that a number of regions between five and ten provides efficient working conditions for the current Internet. The experiments show that the computational complexity is within realistic values. Apart from the above characteristics no other restrictions apply. The region graph also contains the indication of the links to other regions. A slight modification of the algebra is necessary to have convergence for inter-region paths [17].

VIII. Failure Management

The static graph is no guarantee that the links are up. The dynamic part of the protocol is used to create awareness on link failures both at the reachability and routing levels. There are two goals: assure that no routing loops occur during failures and that no packets are lost if at least a failure free path exists to the destination.

Only links fail (a failing AS means all its links failed). Assume a link fails. Routers disseminate a control packet at reachability or at routing levels. Control packets contain the link identification, its Direction (up, down, or both) and the sequence number of the graph.

[3] covers the reachability part and only a paragraph summary is provided here. When a control packet arrives (or the failure of the link is detected in the case of the first router) the AS checks if it can still reach all reachable ASes without using the failed link. If, at least one reachable AS becomes unreachable, the control packet dissemination continues. The dissemination follows the rules of Tables I and II (thus the need to have the field Direction in the packet). If all ASes remain reachable the dissemination stops there; further evaluation is then performed at the routing level. So, a link can fail and no packet is ever sent at reachability level.

At routing level even if an AS maintains reachability after a link failure the new paths to some destinations might be different from the original ones. Routing loops might exist if these new paths belong to a class with lower preference than the one used until then. I.e., the new paths to a given AS have a less preferred signature than the ones being used.

Consider the example in Fig. 3. AS C has a *p2pbk* link with AS D, AS D is a customer of AS G and AS E is a customer of both AS C and AS G. If a failure occurs in link C-E all ASes

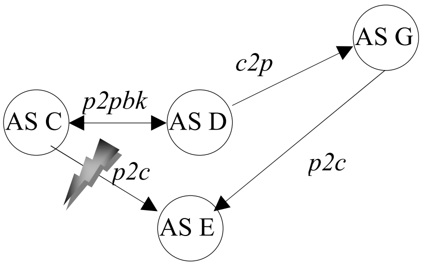

Fig. 3 – Failure example

remain reachable through valid paths from both C and E. If only reachability was considered no control packets would be sent for this case. However consider $P_r(C)$ and the destination AS E. Before the failure the path signature from C to E was *P2C* (the direct path). After the failure the signature is (*BKP*, 1) (D-G-E path). This change of path becomes possible just because of the failure. AS C knows about the failure and starts to route packets to E through it. But AS D is unaware and will continue to choose the *P2Pbk* path (the C-E path) over the least preferred *C2P* path (G-E). So, a loop occurs between AS D and AS C during the failure.

AS D must be notified about the failure because it is affected by it in terms of routing. This happens because AS C changed its selected path to a least preferred path than the original one, and in the network graph known by AS D (still with the failed link) this breaks the monotony of DTIA's routing algebra. As soon as AS D is notified about the failure it will start to use a map with this information and DTIA's convergence properties are assured. The conditions for dissemination are slightly more general: if any path from an AS has its signature changed (downgraded) control packets are disseminated to the neighbors (according to the rules). These control packets contain the identification of all the links this AS knows as failed (to consider the effect of multiple failures). When an AS receives a control packet only identifying failed links that it already knows about, the packet is discarded. Note that if all the paths keep the preference signature no notifications are issued.

Note also that over the time the graphs ASes have are not the same because ASes are only notified about the failures that have a direct effect on them (Sometimes control packets contain additional links but they have little impact in triggering the change of signatures). This is a very powerful contention mechanism.

The description in this section assumes that all ASes are synchronized in terms of map versions. The serial number included in the control packets is used to force the synchronization on graph versions. The update of map versions is left outside of this paper due to space reasons.

As the dissemination also follows the rules, control packets are not sent to ASes whose valid paths are not affected by the failure (e.g. an AS that detects a link failure to one of its providers does not send a control packet to another provider – in this case the failed link is already invalid since valleys are not allowed).

When the link comes up again a similar algorithm is used to disseminate a control packet with the link up information. I.e., dissemination continues if an AS that was unreachable because of this link becomes reachable, or if a more preferred path starts to be used with the correction of the failure. The dissemination uses reliable sessions.

The scope of the dissemination is directly related to the degree of multihoming in the region. A high degree of multihoming makes the disseminating region smaller (there is less losses of reachability to some AS); and in terms of routing more alternative paths with the same level of preference will exist (e.g. two *C2P* paths or two *P2C* paths) stopping the dissemination.

If a single-connected AS fails the dissemination always reaches the entire region (this AS becomes unreachable for everybody). However, the failure of an AS is a rare event unless they are stub ASes (that are even more likely to fail). For a stub AS connected to only one provider no control packets are sent from this provider to avoid warning the entire region. Therefore, data packets will fail at the provider. This lack of delivery guarantee is consistent in the current Internet because even today packets can reach a destination AS just to know that the prefix might not be valid at that moment (and for some reason it is still advertized, or not yet redrawn).

A. DTIA's routing correctness in presence of failures

To prove the correctness of the routing protocol when failures occur we start by proving that every concerned AS is informed (both in terms of reachability and routing). Then, we prove that transient loops are contained and do not survive the dissemination of the control packets. Finally, the last theorem proves that if there is a path to the destination no packet is lost and the protocol converges. The proofs of Theorems 5, 7 and 8 can be found in [3].

Theorem 5: *The control packet dissemination mechanism is guaranteed to inform every AS that experiences the following: a previously reachable AS becomes unreachable due to the failure.*

Theorem 6: *The control packet dissemination mechanism is guaranteed to inform every AS that has to change routing decisions to maintain convergence.*

Proof: Let *G* be the region static graph and *DG(t)* the region dynamic graph at time *t* (i.e., considering the failed links up to instant *t*). R_n is the set of all reachable ASes from a given AS *n*. $RD_n(t)$ is the set of current paths at time *t* being used to reach a destination AS D from AS *n*.

An AS (x1) detecting a failure checks if for all AS D $\in R_{x1}$, $RD_{x1}(t)$ has the same signature as $RD_{x1}(t^{-})$ (the path or paths used to destination D just before the failure). If not, it sends a control packet to its neighbors. This control packet is forwarded hop by hop until a hop *n-1*, (x_{n-1}), where $RD_{xn-1}(t)$ has the same signature as $RD_{xn-1}(t^{-})$. At this point the dissemination is stopped and so x_n does not receive a control packet.

The path signatures are calculated according to the operation ⊕ defined in Table III. At x_n for all ASes D reachable through x_{n-1} the signatures of the paths at time *t,* $RD_{xn}(t)$ result from using ⊕ to combine the label of the link x_n-x_{n-1} with the signature of the paths to D in x_{n-1}, $RD_{xn-1}(t)$. If at time $t=t^{-}$ (before the failure) $RD_{xn-1}(t^{-})$ has paths with the same signature than the ones in $RD_{xn-1}(t)$ (at failure time t) then if the label of the link x_n–x_{n-1} is the same at $t=t^{-}$ and $t=t$ the result of ⊕ will also be the same and therefore at x_n $RD_{xn}(t) = RD_{xn}(t^{-})$. This means that x_n does not need to change routing decisions and concludes our proof. Note that there is a subtle aspect concerning signature preferences. It occurs for paths with *P2Patt* and (*BKP, y*) signatures. For these signatures the preference decreases with the number of links of the path, and only one path (the shortest) can be used at a time. Therefore in this case, if the path in $RD_{xn-1}(t^{-})$ is different from the one at $RD_{xn-1}(t)$ (even if it maintains the signature), the control packet has to be forwarded. This is because the result of the ⊕ operation with the x_n-x_{n-1} label is different at x_n that is $RD_{xn}(t) \neq RD_{xn}(t^{-})$.

Theorem 7: *Transient loops caused by control packet inconsistency are contained to one hop and packets loop at most one time between these two routers.*

Theorem 8:
Condition 1: There is at least one available valid path to the destination D during failures.

If condition 1 holds no packet p is lost during the failures

IX. Deployment

The deployment process cannot be based on a synchronized change of the entire world at the same time. We assume that (a) the graph can evolve from the effort of RIPE, and (b) there is a mapping service to know AS identifiers from prefixes.

The deployment process has basically two aspects: how the graph is distributed to ASes; and how DTIA interworks with BGP-4. Let's start with the second one.

ASes running BGP-4 stay as they are. The new system has to be deployed from bottom to up always forming convex areas (i.e., an AS in a region cannot communicate directly to another AS in the region via BGP-4). This means that an AS can only change if all its customers have changed. Regions start to exist with graphs containing a few ASes, and assume that the rest of the world is in another region. ASes in the "new" world communicate with "old" world neighbors using BGP-4. Each time an AS receives prefix advertisements it translates them to ASes and learns destinations (this is the embryonic procedure for inter-region interaction that is covered in [17]). On the other hand, it advertises prefixes from the ASes in the region it can reach by valid paths. This implies that in these early deployment times a reverse mapping service between AS identifiers and prefixes must exist.

Note that the way advertisements are made (if prefixes are aggregated or not, etc.) have strong consequences in the entire system. The frontier between BGP and DTIA has not the flexibility of BGP and not yet the flexibility of DTIA. Engineer problems will exist that cannot be described in this paper (mainly related to attributes). But the interworking is possible. Probably the problems with the fine tuning can constitute and incentive for a quick adherence to the new system.

The other aspect is how the graph is distributed. The current Internet has the characteristic known as the "small world effect": each AS (except minor stub ASes that do not even run BGP) can reach another similar AS passing by a small number of relaying ASes (2 or 3). Reaching any higher level ASes is even shorter. If the AS that contains the server that distributes the graph knows about it and its routers relay the graph request packet to the server, a simple constrained flooding based algorithm can be used to locate the server. More than one server on more than one AS can coexist making the system redundant and faster.

X. Related Work

HLP [8] was an inspiration to our work. They also use AS identifiers instead of network prefixes and their mechanism for scalability is the definition of trees based on *tier-1* ASes. Inside the trees the link-state protocol is used and amongst trees (at *tier-1*) a path-vector is used. HLP takes advantage of the multihoming if the multihoming exist inside of a tree. Multihoming amongst ASes of different trees (something that is very real already) pose problems because it forces ASes to belong to several trees and run several link-state protocols. Their system also fails to address backup links and does not take into consideration the real web of peer-to-peer links or regional cliques that exist already.

Our option of providing the graph to routers can also be seen with minor variations in other works: in [18] routers create a network map "upon receiving structural information"; in [19] "fairly standard techniques" give all routers a consistent view of the potential set of links to enable them to construct the map; NIRA [11] uses a path-vector protocol.

How these systems handle dynamism (link failures) is also different: [18] relies on routers announcing their links from time to time. If announcements are not received the link is considered down; [19] uses data packets to transport link failure information; NIRA [11] relies on reactive mechanisms such as timeout or router feedback (ICMP) to inform routers that were not notified proactively by their routing protocol. Some systems assume as a design choice, as we do, that failures are only notified to the relevant actors [11] [18] [19] [20].

In terms of forwarding rules our system forms a closed monotone system and we rank this as a strong point. This is not the case for any other of the systems reported: HLP [8] assumes deeply the provider-costumer relation and only one-hop peer relations; [19] assumes BGP protocol runs on every router and policy violations are treated as link failures; in [18] each router uses three sets of rules to forward packets (these rules are pretty much regular expression manipulations).

A final aspect worth mentioning is that several systems rely on source-routing [11][18][19], and some need extra information in the IP packet (using probably IP options) or different information in standard fields [11][18][19].

XI. Experiments

This section contains various types of experiments. For the comparison between DTIA and BGP we used the *ns2* simulator (and BGP++ [21]). Due to *ns2* limitations in terms of computational resources the network sizes had to be limited. To study DTIA in terms of scalability we need bigger topologies and specially built emulators implemented in JAVA were used both for DTIA's aspects (path validation, etc.) as well as BGP's ones (route export and decision process). Finally, two other aspects (multihoming and multipath) were addressed in terms of procedural analysis due to the great dependency on concrete topologies that prevents any specific experiment to be clearly conclusive.

A. DTIA's Scalabilty.

In these experiments we intended to assess the necessary time to construct three tables: $P_r(X)$ and $R_r(X)$ that were described above and *FH (X)* which is a table with the various possible first hops to reach a destination AS. We used an AS level topology obtained from the CAIDA AS relationships Data research project [5] trimmed to obtain the topology of 76 countries from Europe and part of Asia (the RIPE region). The topology has 11,335 ASes and over 21,000 links. It is a large region chosen to provide insight about the upper limits of DTIA for the current Internet.

We calculate the *FH (X)* for each ASes in the topology and measured the processing time. We used a machine with an Intel Q6600 processor and 8 GB of RAM.

ASes have different characteristics according to their position in the customer provider hierarchy. However, there are already too many direct links between hierarchical levels to still reason on a simple tier-like structure. We decided to divide the ASes by the number of their neighbors. "*Higher level*" ASes usually have more links than smaller client ASes. We divided the 11,335 ASes into five groups. Fig. 5 plots the average *FH (X)* size and average processing time for each of the five groups (*x-axis*).

Regarding the *FH (X)* table size, and despite the large number of ASes, the largest routing table size is 46,127 for the >180 group. This group has only 0.16% of the ASes of the region. As we descend in the groups the number of entries diminishes. We have a maximum of 24,611 entries for the three lower groups (ASes with at most 80 neighbors) that account for 99.6% of the total region ASes.

The highest measured processing time was 1.03s for the >180 group which is quite reasonable.

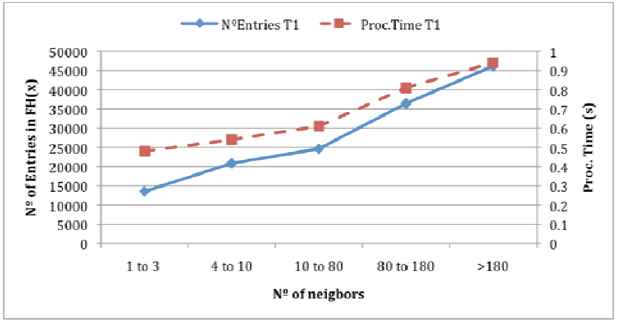

Fig. 5 – Number of *FH (X)* entries and processing time

For 99.6% of the total ASes we have a time smaller than 0.61s proving the feasibility of the DTIA's assumption of having a small number of very large regions.

The *FH (X)* table can have more than one entry for a specific destination thus creating a multi-path system. On the other hand, BGP routing table has only one entry per destination but can have more than one destination per AS due to prefix de-aggregation.

With these differences in mind we used the publically available data from the RIPE Routing Information Service [22] to see the size of a forwarding table in BGP for the same topology used in the DTIA experiments. The RIPE RIS service contains routing data from the real Internet. This routing information system is an AS (AS12654) with fifteen routers in different locations having more than 600 peers. Its routing table gives a large view of the global Routing Table in the Internet. The table has over 305,000 routes for the entire Internet. We then eliminated all routes for prefixes that do not belong to our region (11,335 ASes). The resulted table has 64,345 entries. Although, as we have seen, they are not directly comparable, it is almost 40% larger than the largest of the DTIA's tables.

B. Multihoming and Multipath Routing Support

1) Multihoming

An AS is said to be multihomed when it is connected to more than one provider. Multihoming is an increasing practice in today's Internet. One of its purposes is to provide fault tolerance and its use causes several difficulties in prefix aggregation when using BGP. This has a severe impact in Routing Table growth and therefore affects the Internet scalability [23]. Multihoming also introduces new possible paths and many times ASes want to perform load balancing between these paths [23]. We saw that DTIA's multipath routing capabilities can take advantage of the various paths for load balancing or other traffic manipulations. In this section we separate the aspects of multihoming (for fault tolerance) and multipath (for traffic engineering).

BGP has difficulties on prefix definitions due to multihoming because of the specific longest match preference rule, as explained above. Its consequence to the size of the routing table was also analyzed. In terms of exploitation of multihoming, certain ASes can take advantage of it, but it is only exploited locally. More specifically, one AS can receive advertisements for a certain prefix with different paths. In terms of fault tolerance one could think that a change from one path to the other would be simple. However, paths must be advertised and a failure still has consequences because a withdraw and a new advertisement must be sent. The only advantage is a fast local operation of that particular AS. Beyond this AS this particular multihoming is not known because the AS has to choose only one path.

DTIA's approach is simpler and more powerful. Link failures are only advertised if they have consequences (reachability or routing preferences) and every AS knows the various multihomings in the region.

2) *Multipath routing*

We refer to multipath routing as the existence of more than one route for the same destination at the same time. Multipath routing can be used for load balancing or other traffic engineering techniques.

The multihoming scenario adds further complexity in BGP if used for multipath. If an AS desires to use more than one path at the same time it has to subdivide its prefixes into two or more sub-prefixes, due to the single path nature of BGP. It is a common practice that additionally the AS announces the full aggregate through all providers to maintain connectivity in case of failure. This increases the aggregation problems and impairs BGP's scalability by increasing the routing table size. According to [23] 20%-25% of routing entries were due to load balancing, which was at the time the fastest growing cause to Routing Table growth.

Routers in the Internet often receive more than one path for each destination. They could install multiple routes in the forwarding table. In fact some router vendors now provide BGP-multipath capabilities (for instance, Juniper [24] and Cisco [25]). In both cases more than one equal cost BGP route can be installed in the routing table but only one is advertised. Announcing more than one route for a given destination would make load balancing possible without prefix manipulations. However in BGP several problems arise. First advertising and installing multiple paths introduces new difficulties in ensuring overall convergence and coordination between ASes would be necessary to ensure that routing decisions are coherent. Secondly the possible gain in Routing Table growth due to the usefulness of prefix subdivision would be impaired by the growth in the number of installed paths. Finally the number of messages exchanged will increase with the newly advertised routes.

The reality for DTIA is again quite simple: the fact that the paths are used or not at the same time has no impact at all. The procedures in the multipath case are the same as described above for the multihoming. It is up to a higher layer protocol to take advantage of them. No extra impact on routing table size occurs in this situation and convergence is guaranteed.

We performed an experiment to calculate the routing tables of a real topology presented in [17] using the implemented BGP emulator. The topology has 54 ASes and 517 links and it is built by a subset of stub ASes from Portugal, and the set of transit ASes that they use, up to *tier-1* (assumed as ASes without providers). It includes ten *tier-1* ASes at the top, and a set of lower tier transit ASes connected by *p2c* links. This subgraph of the RIPE region is densely connected, has a high degree of multihoming, and contains a great number of *p2p* links that connect ASes from multiple tiers.

We used only one prefix per AS. This simplification greatly reduces the routing table for BGP since typically ASes announce more than one prefix. The purpose was to make it more comparable with DTIA (which routes by AS instead of prefixes). The calculated routing tables represent a lower bound on the BGP routing table size.

We calculated the routes for regular BGP (best path only) and for BGP advertising up to 14 paths. Fig. 6 shows the number of route table entries for the various cases. The *common polices* were used to emulate the route export process (as it is not known, obviously). We divided the ASes into six groups according to the number of neighbors leaving the *tier-1* ASes in a separate group. The *tier-1* ASes form a clique with *p2p* connections between them. This implies that at least one valid path exists between every pair of destinations.

For BGP without multipath the average routing table size was, as expected, 54. As we increase the number of announced paths, *n*, the number of entries increases. We will have more paths for each of the 54 prefixes but there is a limit that is dependent of the specific topology. BGP inserts an entry in the table for each new path.

For DTIA we considered the *FH (X)* table as the forwarding table which has the different first hops of all the paths to a destination (if the information of the different paths behind that first hop is needed e.g. for traffic engineering purposes, it is available in $R_r(X)$). Note that this option reduces drastically the number of entries for lesser connected ASes. The numbers start with 54 and rise until 693.

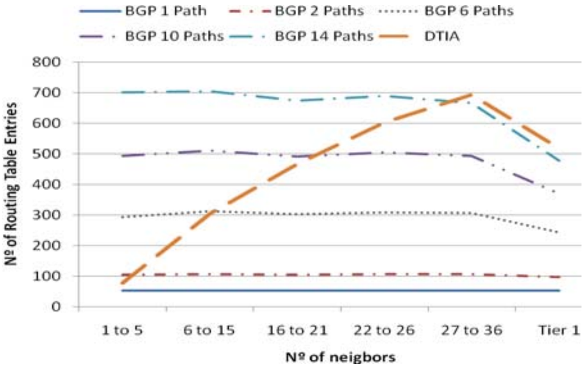

Fig. 6 – Routing table entries for BGP multipath and DTIA

Each time *n* increases there is a great increase in BGP and *n=14* is probably the maximum case for this topology, showing a very large number of entries (around 700 for 54 ASes) network wide. If we consider more than one prefix per AS and possible prefix manipulations, the complexity can be much higher. The number of exchanged messages increases linearly with *n*, even if we send multiple routes in the same update message. This is even more serious in case of failures with more paths to be withdrawn. Finally convergence would not be guaranteed worsening the convergence problems that BGP already exhibits.

DTIA provides multipath natively, ensuring convergence and using the extra paths to improve the behavior under failures. In terms of routing table scalability, even in the simple, unrealistic best case scenario of the experiment the results are better in DTIA.

In Fig. 6 we also observe that for the *n* >= 6 the *tier-1* ASes have smaller tables than the rest. Also some of the highly connected ASes that have only one provider (and a lot of clients and peers) have slightly smaller tables. The reason is the following: the link type from which more paths can be received using the *common polices* is the *p2c* type (all routes can be announced to a client). Therefore, since *tier-1* ASes have no providers they receive fewer routes when using a high number of paths.

The same happens in DTIA and the reason is similar, since the validity rules and routing algebra are also based in the *common policies*.

C. Failure Propagation / Churn

One of the biggest problems of BGP that impacts also its scalability is the high churn rate and slow convergence after routing events. The increase of multihoming either at stub level or in the mid-tier section of the Internet topology worsens the problem. In [26] the effect of the increased multihoming degree (mean number of providers per AS) in several zones of the topology was studied. An increase of 1.6 in churn was measured following an increase of 3 in the multihoming degree. This is a strong difference to DTIA. In DTIA, increasing the multihoming degree leads to a decrease of the churn rate. Failures are more likely to be contained in heavily multihomed topologies because it is more likely to have several alternative routes with the same signature to most destinations. Multihoming actually improves DTIAs convergence time and reduces churn after a routing event.

We conducted an experiment using the *ns2* simulator to compare the convergence and churn rate of DTIA and BGP to evaluate DTIA convergence after a routing event.

We used the same topology as in the multipath experiment. This topology is densely connected and has a high degree of multihoming. Sixty six single links failures were produced, and we measured the number of affected ASes, both for DTIA and BGP.

.

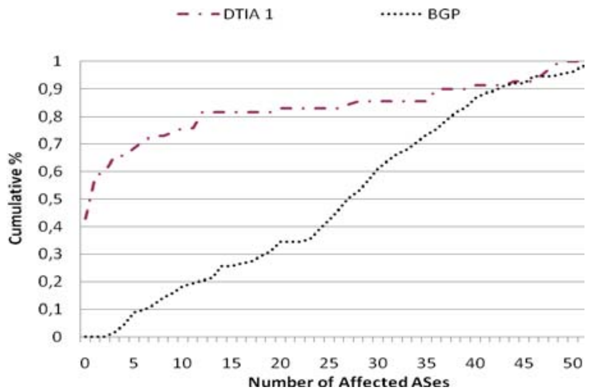

Fig. 7 – Cumulative percentage of affected ASes after a single link failure

Fig. 7 shows the cumulative percentage of ASes that receive a control packet in DTIA or that receive a route withdraw packet in BGP. The behavior of DTIA is very good: 70% of the failures affected less than 5 ASes. 80% of the failures affected less than 15 ASes and only 10 % affected more than 35 ASes. BGP cannot restrict churn so well. 5 or less ASes were affected only for 8% of the failures. 25 % of the failures affected 15 or less. This experiment indicates that DTIA greatly reduces the number of ASes that know about a failure and consequently reduces churn

XII. General Assessment

The purpose of this section is to provide a critical view about the system just presented, covering different aspects.

The work presented in this paper started from the features identified on BGP. The labels *p2c*, *p2p* and *c2p* are quite obvious and the others were introduced to provide a different and more natural approach to backups and to support sibling relations. How the protocol behaves in face of these labels is described by the rules of the first three tables. Again, the drive for the rules was a close relation with BGP. The sets of rules and labels form a monotonic algebra which we use to form a kind of “first-level” routing system. Based on this, and using the assignment of more traditional costs to the links (bandwidth, delay, monetary cost, etc.) a “second-level” routing can be defined to take advantage of the multipath, by performing traffic engineering, provider-customer backup use, etc.

Clearly our link types identify the “business part” of BGP. An interesting question is whether it is possible to include more BGP features enlarging the set of types and defining appropriate rules? If so, a requirement must be the maintenance of the monotonicity in order to keep the system “safe” in the sense of creating more easily a system that works and converges. Another question is trying to understand the border between “business policies” (first-level) and “traffic-policies” (second-level)? Seen from DTIA’s perspective, BGP mixes very much these two kinds of policies.

When we analyze our system probably the most important current feature that we do not support is the possibility to influence incoming traffic. This is currently performed using, for instance, the MED attribute[3]. Incoming traffic is pretty much a traffic engineering aspect and should be performed at the third level of our architecture.

Another relevant issue is the graph of the region. Why is an AS interested in stating that it has a certain link with a certain label? The *p2c* and *c2p* links are obvious – money. A *p2p* link might be in the graph or not. The adjacent ASes know the link exists, so there is no need to put it on the graph. But the AS should be interested in putting it in the graph for its clients to use it instead of routing their traffic over other valid paths through other ASes without generating revenue. *p2pbk* is also obvious because financial advantages can be agreed on its usage (both active and standby). Finally *p2patt* has an obvious advantage when both ASes belong to the same organization. When they do not, these types of links can be used to form cliques (at regional or city level). New business models can also be defined using this type of links (charging third party packets, for instance). What is interesting to see is that we can find monetary reasons for the ASes to participate in the

[3] Other ways are to work on AS Path prepending (requiring the knowledge of the topology) or disaggregate prefixes (putting pressure on the routing table).

building of the graph. Naturally the information gathering process is an administrative task.

One major decision in our system was to separate aspects: reachability, routing, higher level traffic engineering, naming and addressing (these last two are related to the mapping service and were less covered in this paper). By separating them, simpler and more appropriated solutions are defined for each one and the Internet can evolve in a simpler way than today. One characteristic of the Internet that is preventing the speed of the introduction of new solutions compared to other architectures such as for instance the cellular systems is the overload of features in a very small set of entities. Any change has tremendous consequences. Our system breaks with this tradition. We could think on another routing protocol based on the reachability protocol. The mapping service between prefixes and ASes could have mobility features – an entity could keep its address and change AS. Incidentally, prefixes have no meaning in our system. What is important is a translation mechanism between an identifier and an AS. The meaning of the identifier and how it could be mapped inside the AS is not part of the system. DTIA is an inter-domain routing system and only direct concerns on this matter are relevant.

XIII. Conclusion

This paper proposed a possible architecture for Internet inter-domain routing. It is a piecewise approach to the problem starting at reachability producing valid paths, then "first-level" routing taking business relations into consideration and then "higher-level" routing for more traffic related issues (only the routing part is covered in this paper).

The separation of the different concerns also helped to reach simple solutions at each level that are even though inter-related and can provide answers to the new challenges the Internet is facing.

Inter-domain routing is a very sensitive issue and a drastic change will never happen. Smooth changes might have some possibility and this paper is a contribution for a discussion on what should be the aspects that the community can consider as primary be willing to relinquish on the others.

This work opens new and exciting directions of research. The algorithms we used to calculate paths and work on the graphs can be improved. Traffic-policies to choose paths when many are available can be built as higher-level protocols addressing issues such as traffic engineering, backup provider-customer links, preference for university-related paths, etc., etc.

References

[1] Yannuzzi, M.; Masip-Bruin, X.; Bonaventure, O. *Open issues in Interdomain routing: a survey*, IEEE Network Nov/Dec 05, 19(6)

[2] Griffin, T., Shepherd, F., Wilfong, G., *The Stable Paths Problem and Inter-domain Routing*, IEEE Trans. Net. 10(2), Apr. 02, pp. 232–43

[3] Amaral, P., Bernardo, L., Pinto, P., X., *DTIA – An Architecture for Inter-domain Routing*, IEEE ICC'09, June 2009, Dresden, Germany

[4] Gao, L., *On inferring Autonomous System relationships in the Internet.* In IEEE/ACM Trans. Net., December 2001

[5] CAIDA.ASRelationshipsData.ResearchProject, http://www.caida.org/data/active/as-relationships/

[6] RFC4984: Report from the IAB Workshop on Routing and Addressing, September 2007

[7] Bonaventure, O., *Reconsidering the Internet Routing Architecture*, Internet draft, draft-bonaventure-irtf-rira-00.txt, 2007

[8] Subramaniam, L., Caesar, M., Ee, C., *et al.*, *HLP: A next Generation Inter-domain Routing Protocol*, SIGCOMM 2005, Philadelphia

[9] Chang, R., and Lo, M., *Inbound Traffic Engineering for Multihomed ASes Using AS Path Prepending*, IEEE Network, Mar. 2005

[10] Donnet, B., and Bonaventure, O., *On BGP Communities* ACM SIGCOMM Computer Communication Review, vol 38. No. 2, 2008

[11] Yang, X., Clark, D., Berger, W. A., *NIRA: A new Inter-Domain Routing Architecture*, IEEE Trans. Net., vol. 15. No. 4, August 2007

[12] RIPE database, http://www.ripe.net/db/index.html

[13] Dimitropoulos, X., *et al.*, *AS Relationships: Inference and Validation,* ACM SIGCOMM Computer Communication Review, 2007

[14] Caesar, M., Rexford, J., *BGP Routing Policies in ISP networks*, IEEE Network Magazine, Nov/Dec 2005

[15] Griffin, T., Sobrinho, J., *Metarouting*, SIGCOMM'05, Aug 22-26

[16] Sobrinho, J., *An Algebraic Theory of Dynamic Network Routing*, IEE/ACM Transactions on Networking, Vol. 13, No. 5, October 2005

[17] Amaral, P., Ganhão, F., Assunção, C., Bernardo, L., Pinto, P., "Scalable multi-region routing at Inter-Domain Level", IEEE GLOBECOMM'09, Nov-Dez 2009, Honululu, USA

[18] Zhu, D., Gritter, M., Cheriton, D., "Feedback Based Routing", SIGCOMM CCR, Vol 33, No 1, pp. 71-76, January 2003

[19] Lakshminarayanan, K., Caser, M., et al, "Achieving Convergence-Free Routing using Failure-Carrying Packtes", SIGCOMM'07, Kyoto, Japan

[20] Levchenko, K., Voelker, G., Paturi, R., Savage, S., "XL: An Efficient Network Routing Algorithm", SIGCOMM'08, pp. 15-26, Seattle

[21] BGP++ Home Page. Retrieved from http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/.

[22] RIPE Routing Information Service (RIS), http://www.ripe.net/ris/

[23] Tian Bu a, Lixin Gao, Don Towsley, "On characterizing BGP routing table growth", Computer Networks issue 45 2004 pp 45-54.

[24] Juniper, "Configuring BGP to Select Multiple BGP Paths," JUNOS Software Documentation.

[25] Cisco, "BGP Best Path Selection Algorithm," Cisco Documentation, http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094431.shtml

[26] Elmokashfi, A., Kvalbein, A., and Dovrolis,. On the scalability of BGP: the roles of topology growth and update rate-limiting. ACM CoNEXT Madrid, Spain, Dec. 2008.

Pedro Amaral Received a degree in Electrical Engineering and Computers in 2001 and the M.Sc. in Computer Engineering in 2006 from *Universidade Nova de Lisboa*. He is currently working towards the Ph.D. degree at Department of Electrical Engineering, *Universidade Nova de Lisboa.* His research interests are inter-domain routing, Internet architecture, routing, traffic engineering and quality of service.

Luis Bernardo Received his Ph.D. degree in Electrical Engineering and Computers from *Instituto Superior Técnico*, Lisbon, Portugal, in 2002. He is an assistant professor at *Universidade Nova de Lisboa*, Portugal. His current research interests include MAC protocols for wireless systems, wireless sensor networks, mobile ad hoc networks, routing protocols and quality of service. He is a member of the IEE and the ACM.

Paulo Pinto Received the Ph.D. degree in Computer Science from the University of Kent, at Canterbury, and diploma in Electrical Engineering from *Instituto Superior Técnico*, Lisbon, Portugal. He is an associate professor at *Universidade Nova de Lisboa*, Portugal. His current research interests include interconnection of wireless networks, MAC protocols for wireless systems and routing protocols. He is a member of the IEE and the ACM.

Bibliography

[ABKM01] David Andersen, Hari Balakrishnan, Frans Kaaashoek, and Robert Morris. Resilient overlay networks. *ACM SIGOPS*, 2001.

[ABP08] Pedro Amaral, Luís Bernardo, and Paulo Pinto. Dtia: an inter-domain reachability architecture technical report. Technical report, September 2008.

[ABP09] Pedro Amaral, Luís Bernardo, and Paulo Pinto. Dtia - routing at the inter-domain level. Technical report, 2009.

[AGA+09] Pedro Amaral, Francisco Ganhão, Cláudio Assunção, Luís Bernardo, and Paulo Pinto. Scalable multi-region routing at inter-domain level. *Globecomm*, November 2009.

[bg309] Bgp++. see `http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/`, September 2009.

[Bon07] Olivier Bonaventure. Reconsidering the internet routing architecture. Internet draft, draf-bonaventure-irtf-rira-00.txt, 2007.

[Bra89] R. Braden. Requirements for internet hosts. communication layers. Technical report, IETF, RFC 1122, October 1989.

[bri09] Brite: Boston university representative topology generator. see `http://www.cs.bu.edu/brite/`, August 2009.

[cai09a] Caida as relationships data research project. see `http://www.caida.org/data/active/as-relationships/`, September 2009.

[cai09b] Caida data. see `http://www.caida.org/data/`, August 2009.

[CCF$^+$05] Matthew Caesar, D. Caldwell, N. Feamster, Jennifer Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a routing control platform. In *NSDI*, 2005.

[CRK89] Chunhsiang Cheng, Ralph Riley, and Srikanta P.R. Kumar. "a loop-free extended bellman-ford routing protocol without bouncing effect". *ACM SIGCOMM Computer Communication Review*, 1989.

[CS06] Reuven Cohen and Amnon Shochot. The "global-isp" paradigm. *ELSEVIER Computer Networks*, October 2006.

[Dai04] L. Daigle. Whois protocol specification. Technical report, IETF, RFC 3912, September 2004.

[Dat09] Merit Network Routing Assets Database. Internet routing database. see `ftp://ftp.ra.net/`, July 2009.

[DB08] Benoit Donnet and Olivier Bonaventure. On bgp communities. *ACM SIGCOMM Computer Communication Review*, 2008.

[DF07] Benoit Donnet and Timur Friedman. Internet topology discovery: A survey. *IEEE Communications Surveys, Volume 9, No. 4*, 4th Quarter, 2007.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik, 1, pp. 269-271*, 1959.

[DKF$^+$07] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc klaffy, and George Riley. As relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review, Volume 37, Number 1*, January 2007.

[ea99] C. Alaettinogluoglu et al. Routing policy specification language (rpsl). Technical report, IETF, RFC 2622, June 1999.

[ea02a] B. Huffaker et al. Topology discovery by active probing. In *Proc. Symp. Applications and the Internet (SAINT)*, January 2002.

[ea02b] H. Chang et al. Towards capturing representative as-level internet topologies. In *Proc. ACM SIGMETRICS*, June 2002.

[ea02c] Q. Chen et al. The origin of power laws in internet topologies revisited. In *Proc. IEEE INFOCOM'02, New York, NY*, 2002.

[ea02d] Z. M. Mao et al. Route flap damping exacerbates internet routing convergence. In *ACM SIGCOMM*, 2002.

[eig09] Cisco - eigrp. see `http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094cb7.shtml`, September 2009.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. SIGCOMM, 1999.

[FFO07a] D. Farinacci, V. Fuller, and D. Oran. Locator/id separation protocol (lisp). internet draft draft-farinacci-lisp-00.txt. Technical report, IETF Network Working Group, January 2007.

[FFO07b] D. Farinacci, V. Fuller, and D. Oran. Locator/id separation protocol (lisp). internet draft draft-farinacci-lisp-01.txt. Technical report, IETF Network Working Group, January 2007.

[Gao01] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking, Vol.9, NO. 6*, December 2001.

[gnu09] Gnu/zebra. see `http://www.zebra.org/`, September 2009.

[GP01] Timothy G. Griffin and Brian J. Premore. An experimental analysis of bgp convergence time. *IEEE International Conference on Network Protocols*, 2001.

[GS05] Timothy G. Griffin and João Luís Sobrinho. Metarouting. In *SIGCOMM'05*, 22-26 of August 2005.

[GT00] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *Proc. IEEE INFOCOM*, March 2000.

[gti09] Gt-itm: Georgia tech internetwork topology models. see `http://www.cc.gatech.edu/projects/gtitm/`, September 2009.

[IB07] Luigi Iannone and Olivier Bonaventure. On the cost of caching locator/id mappings. *ACM CoNEXT*, December 2007.

[ige09] Igen: Topology generation through network design heuristics. see http://www.info.ucl.ac.be/~bqu/igen/, September 2009.

[JI92] ork. J. Internem. "dynamics of link-state and loop-free distance-vector routing algorithms". *vol. 3*, pp. 161-188, 1992.

[KKK07] Nate Kushman, Srikanth Kandula, and Dina Katabi. Can you hear me now?! it must be bgp. *ACM SIGCOMM Computer Communication Review*, April 2007.

[LCR$^+$07] Karthik Laksminarayanan, Matthew Caesar, Murali Rangan, Tom Anderson, Scott Shenker, and Ion Stoica. Achieving convergence-free routing using failure-carrying packets. *ACM SIGCOMM Computer Communication Review*, August 2007.

[LVPS08] Kirill Levchenko, Geoffrey M. Voelker, Ramamohan Paturi, and Stefan Savage. Xl: An efficient network routing algorithm. *ACM SIGCOMM Computer Communication Review*, August 2008.

[Mal98] G. Malkin. Rip version 2. Technical report, IETF - Network Working Group, Novemver 1998.

[met] Predicting the internet's catastrophic collapse and ghost sites galore in 1996. see Infoworld, December 4, 1995.

[MKF$^+$06] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc klaffy, and Amin Vahdat. The internet as-level topology: Three data sources and one definitive metric. *ACM SIGCOMM Computer Communication Review, Volume 36, Issue 1*, 2006.

[Moy98] J. Moy. Ospf version 2. Technical report, IETF - Network Working Group, 1998.

[MW77] J. McQuillan and D.C. Walden. "the arpanet design decisions". *Computer Networks, vol. 1*, August 1977.

[NBBB98] K. Nichols, S. Blake, F. Baker, and D. Black. Rfc 2474 - definition of the differentiated services field in the ipv4 and ipv6 headers. Technical report, IETF - Network Working Group, December 1998.

[NCC09] RIPE NCC. Routing registry consistency check reports. see `http://www.ripe.net/projects/rrcc/`, July 2009.

[nsM09] ns-2 manual. see `http://www.isi.edu/nsnam/ns/ns-documentation.html`, September 2009.

[nsR09] *ns-2* - network simulator 2. `http://www.isi.edu/nsnam/ns/`, September 2009.

[nsT09] Marc grei's tutorial for the network simulator "ns". see `http://www.isi.edu/nsnam/ns/tutorial/`, September 2009.

[NSW02] R. Mahajan N. Spring and D. Wetheral. Measuring isp topologies with rocketfuel. In *Proc. ACM SIGCOMM*, August 2002.

[OBOM03] Yasuhiro Ohara, Manav Bhatia, Nakamura Osamu, and Jun Murai. Route flapping effects on ospf. In *Applications and the Internet Workshops*, 2003.

[oO09] University of Oregon. Route views, university of oregon route views project. see `http://www.routeviews.org/`, July 2009.

[ooR09] Openoffice. see `http://www.openoffice.org/`, September 2009.

[Ora] D. Oran. Rfc 1142 "osi is-is intra-domain routing protocol". Technical report, IETF.

[osi09] Osi: Open system interconnection. see `http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip`, September 2009.

[rir09] R. i. registries. see `http://www.isoc.org/briefings/021/`, July 2009.

[RLH06] Y. Rekhter, T. Li, and S. Hares. Rfc 4271 - a border gateway protocol 4 (bgp-4). Technical report, IETF - Network Working Group, 2006.

[SARK02] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *IEEE INFOCOM 2002 , New York*, June 2002.

[SCE$^+$05] Lakshminarayanan Subramanian, Matthew Caesar, Cheng Tien Ee, Mark Handley, Morley Mao, Scott Shenker, and Ion Stoica. Hlp: A next generation inter-domain routing protocol. *ACM SIGCOMM Computer Communication Review*, August 2005.

[SKM09] Amit Sahoo, Krishna Kant, and Prasant Mohapatra. Bgp convergence delay after multiple simultaneous router failures: Characterization and solutions. *Computer Communications*, 2009.

[Sob05] João Luís Sobrinho. An algebraic theory of dynamic network routing. In *IEEE/ACM Transactions on Network, Vol. 13, No.5*, October 2005.

[tan02] *Computer Networks, 4th Edition.* Prentice Hall PTR, 2002.

[tcl09] Tcl developer site. see `http://www.tcl.tk/`, September 2009.

[VCG98] C. Villamizar, R. Chandra, and R. Govindan. Rfc 2439 - bgp route flap damping. Technical report, IETF - Network Working Group, 1998.

[VPSV02] Alexei Vázquez, Romualdo Pastor-Satorras, and Alessandro Vespignani. Large-scale topological and dynamical properties of the internet. *Physical Review, Volume 65, 066130*, 2002.

[YCB07] Xiaowei Yang, David Clark, and Arthur W. Berger. Nira: A new inter-domain routing architecture. *IEEE/ACM Transactions on Networking*, 2007.

[YMBB05] Marcelo Yannuzi, Xavier Masip-Bruin, and Olivier Bonaventure. Open issues in interdomain routing: A survey. *IEEE Network*, November/December 2005.