Pedro Andrés Aranda Gutiérrez

PaRArch: An Alternative Routing Architecture for the Internet

Dissertation

submitted to the

Faculty of Electrical Engineering,
Computer Science, and Mathematics

in partial fulfillment of the requirements for the degree of

Doctor rerum naturalium (Dr. rer. nat.)

Paderborn, 15. January 2012

Referees:

Prof. Dr. rer. nat. Holger Karl, University of Paderborn, Germany
Dr. James Roberts, INRIA, Paris, France

Additional committee members:

Prof. Dr. Franz Rammig, University of Paderborn, Germany
Prof. Dr. Gerd Szwillus, University of Paderborn, Germany
Jun.-Prof. Dr. Christoph Sorge, University of Paderborn, Germany

Submission: September, 2012
Examination: 15. January, 2013
Publication: 1. February, 2013

Abstract

The BGP-4 protocol controls the routing information exchange between the independent networks or Autonomous Systems that form the Internet. It is also the main tool for controlling the traffic exchange between them. To control this traffic, routing policies are used. At the same time, the use of routing policies makes BGP-4 become metastable, making it possible for the protocol to oscillate or to produce unexpected results.

In this work, I study the general evolution of the routing tables in the core of the Internet and instances of unstable behaviour of BGP-4. In this study, I conclude that the routing table in the core of the Internet can be reduced by a third of its entries. These entries are used for local configurations traffic control between leaf networks and their providers and have little or no impact on how remote networks route traffic towards these leaf networks.

As a result, I propose PaRArch, a routing architecture that uses BGP-4 and separates the local routing policies between leaf networks and their providers from the routing information that is going to be used in the core of the Internet in two different and isolated routing planes. Using a proof of concept implementation on a virtualised network emulation environment, I show that local routes do not leak to the main routing table and that control over traffic is much tighter than by using state-of-the-art BGP-4 traffic engineering techniques.

One of the main benefits of PaRArch is that it allows the routing plane of the core network of the Internet to be controlled by an instance of BGP-4 that works without policies. This eliminates metastability from this BGP-4 instance and renders a stable and predictable core network for the Internet.

Zusammenfassung

Das Internet besteht aus einzelnen, für sich selbst agierenden und entscheidenden Teilnetzen, den sogenannten *autonomen Systemen*. Um Daten zwischen diesen autonomen Systemen auszutauschen, müssen Information zur Wegewahl (Routinginformationen) zwischen ihnen ausgetauscht werden. Diesen Austausch kontrolliert heute das so genannte Border Gateway Protokoll in Version 4 (BGP-4). Es ist damit das wichtigste Werkzeug, um den Datenverkehr zwischen den autonomen Systemen zu kontrollieren. Diese Kontrolle wird an Hand von Richtlinien, den sogenannten „Routing Policies" implementiert. Gleichzeitig sind diese „Routing Policies" dafür verantwortlich, dass BGP-4 metastabil wird und es zu Schwingungen im Netz kommen kann. Diese Schwingungen beeinträchtigen den normalen Datenfluss und können zu unvorhergesehene Pfade für den Datenaustausch im Netz führen.

In dieser Arbeit studiere ich wie sich die Anzahl der Einträge in den Routingtabellen im Internet zwischen 2000 und 2010 entwickelt hat. Ich zeige, dass sich die Routingtabellen des Internets ungefähr um ein Drittel der Einträge reduzieren lassen, weil es sich bei diesem Drittel um Einträge handelt, die sich auf lokale Konfigurationen zwischen autonomen Systemen am Rand des Internets und deren Providern zurückführen lassen. Diese Einträge haben wenig oder gar kein Einfluss auf den Datenstrom in Netzen, die topologisch entfernt von den Netzen, die sie generierten, im Internet liegen. Außerdem analysiere ich Vorfälle, wo BGP-4 unstabil wurde.

Diese Beobachtungen führen mich dazu, eine neue Routingarchitektur für das Internet, PaRArch, vorzuschlagen. Diese Architektur basiert auf BGP-4, entkoppelt aber den Austausch von Routinginformation zwischen autonomen Systemen am Rand des Internets und deren Dienstleistern vom restlichen Internet. Dazu werden zwei isolierte Datenaustauschebenen definiert. Die eine führt Routinginformationen, die nur lokal signifikant sind, während die andere die Routingdaten führt, die allgemein im Internet gebraucht werden. An Hand einer Implementation auf einem Netzemulator, zeige ich dass die lokale Routinginformation tatsächlich nicht in die allgemeine Routingebene weitergereicht wird und dass eine Kontrolle des Datenflusses wie bei anderen gängigen BGP-4 Konfigurationen möglich ist.

Einer der wichtigsten Vorteile von PaRArch ist, dass die allgemeine Routingebene

des Internets mit einer BGP-4 Instanz kontrolliert wird, die nicht von „Routing Policies" Gebrauch macht. Dadurch wird das zentrale Netz des Internet stabil und vorhersagbar, da die Ursache für die Metastabilität von BGP-4 verschwindet.

Acknowledgement

This work would not have been possible without the support and guidance of Prof. Dr. Karl from the first moment, when it crossed my mind to prepare my dissertation. I thank him specially for embarking in the adventure of guiding me from a distance of around 1900 kilometres.

I would also like to thank the MONAMI *family* for their support and all the anonymous evaluators at the different conferences for their comments –some harsh, but all constructive–. All helped me improve my work.

A significant part of this work has been possible because there are people collecting data from the Internet routing infrastructure. Without the work of team caring for RIPE's Routing Repository, a significant part of my research would not have been possible.

My family has been my inspiration during all this time. Without the vision of my grandparents and parents, who prepared me for a multilingual world at very difficult times, this work would have been much more difficult if not impossible.

Madrid, 15. September 2012

Contents

Chapter 1

Introduction

The Internet is a network of networks. It is the result of interconnecting different data packet networks that use the Internet Protocol (IP) as defined in RFC 791 [1]. Information flow in data packet networks is controlled by two functions: switching and routing. Switching implements transmitting an incoming data packet on an outgoing interface and is implemented locally at each node. Routing is a network-wide function to calculate the paths that should be followed by data packets from a source to a destination. Routers combine their knowledge about their topological neighbourhood in the network with the information they receive from other routers, locally creating a global view of the network that is used to calculate the network paths. This is a distributed process, which involves all routers in a network. In order to transmit packets reliably in a network, the routing process has to converge and yield stable configurations in all nodes. Routing protocols can be influenced in the way they calculate the paths followed by data packets by using *routing policies*.

Each of the networks which make up the Internet is known as an *Autonomous System (AS)*. ASes have clear boundaries and a central authority that defines which routing protocols are used in the network and which routing policies are applied. ASes run the *Border Gateway Protocol (BGP-4)*, defined in RFC 4271 [2], to exchange routing information with other ASes. *Internet Service Providers (ISPs)* own and operate one or more ASes. It is more coherent to use the term ISP when discussing commercial relationships between networks in the Internet and to use the term Autonomous System when discussing technical issues related with routing.

Autonomous Systems that are interconnected are said to be *peering*. These interconnections are governed by economic rules that condition the technical implementation of the peering agreement. There are basically two types of peering relationships between Autonomous Systems: client-provider and peer-to-peer. In a client-provider relationship, the AS acting as client will pay the AS acting as provider for the received traffic. Peering agreements between provider and client ASes stipulate the way the traffic volumes are measured and the price for different traffic volumes. In a peer-to-peer relationship, traffic is exchanged on a fair basis and in a balanced way between the ASes. Peering

agreements for this kind of interconnections stipulate compensations when the fairness principle is not respected by one of the parties. Another important economic factor for Internet Service Providers are the costs for setting up the network infrastructure and the costs for keeping the infrastructure functional.

The economic viability of ISPs has prompted the need to optimise both how the traffic is routed internally in an AS and how the traffic is exchanged with other ASes. To achieve traffic distributions that are (at least, approximately) optimal, *Traffic Engineering (TE)* techniques are used. TE is implemented by modifying how routers calculate the next hop for packets, i.e., their forwarding tables, implementing so-called *routing policies* in the routers' configurations.

In the present work I study inter-AS Traffic Engineering techniques. I use BGP-4 traffic traces that are being collected in a long-term effort by the European Regional Internet Registry (RIR), Réseaux IP Européens (RIPE). These BGP-4 traffic traces are stored in a routing repository that is a part of RIPE's Routing Information Service (RIPE RIS) project [3, 4]. I propose a method to identify TE techniques through their fingerprints in BGP-4 traffic traces and use the results of this study to show weak points in the Border Gateway Protocol. Finally, I propose modifications to it that yield a more robust and scalable Internet routing architecture.

I take three different approaches: my first approach is to study BGP-4 traffic traces from periods of time with no apparent dysfunctions in the Internet. For these periods of time, I propose a two-stage method to detect TE activity. It involves filtering data from the BGP-4 repositories to isolate update sequences with specific characteristics and identifying the network prefixes affected by them. Then I study the statistical distributions of the arrival times of BGP-4 updates containing these prefixes over a longer period of time. My second approach takes advantage of a serious routing incident in the summer of 2009 [5]. I study the intrinsic characteristics of the Border Gateway Protocol protocol which led to it, how an earlier storm with similar root causes was dealt with, and show how TE techniques did not only not help to control the routing storm, but actually helped to spread it through the Internet. The third and last approach I use in the present work is to study the evolution of the Internet's routing table over the last 10 years and concentrate on the prefixes which can be traced back to TE related configurations.

My research shows the following:

1. Routing policies which can be traced back to Traffic Engineering techniques propagate in the Internet core's routing tables beyond the topological neighbourhood where they are meant to control the traffic to places where they have no effect. They manifest themselves as superfluous routing table entries.
2. Routing data generated by routing policies account for one third or approximately 100,000 routes of the Internet's routing tables. These data are useless outside the scope of the routing policies.

3. The routing storms observed in 2009 were originated by session resets that were forced as a response to malformed BGP-4 packets. When a malformed BGP-4 packet is sent, the peer resets the session and the malformed BGP-4 is sent again and the session is reset again, and so on until the peering is shut down. During this process, BGP-4 includes no root cause information to help trace and correct the problem.
   Additionally, I also observed that, in order to provide a quick remedy for a BGP-4 storm in January of 2009 [6], BGP-4 was made more tolerant to some types of invalid BGP-4 data. This modification resulted in the leak of internal AS topology data, which should normally not be accessible to the outside world.

I propose an alternative behaviour for BGP-4 that yields a more robust protocol and, therefore, a more reliable Internet routing architecture. Concretely, I propose two actions:

1. To use a parallel routing table mechanism to implement routing configurations with specific purposes like implementing Service Level Agreements. Routing information that is only needed in a specific peering is kept local using this mechanism.
2. To use parallel and independent BGP-4 sessions for each of the aforementioned routing tables. The main routing table is handled by an instance of BGP-4 running on the standard TCP port, while the rest of the routing tables use alternate sessions using non-standard TCP ports. Any new feature is introduced in one of these sessions. We thereby avoid malformed packets in the main routing table and thereby, we reduce the possibility of routing storms in the Internet.

These alternatives have been implemented and tested with the Netkit network emulation environment and the open source QUAGGA routing software and will be submitted to the Internet Engineering Task Force (IETF) for consideration.

The rest of this work is structured as follows:

- Chapter 2 introduces organisation of the Internet and describes the Border Gateway Protocol BGP-4.
- Chapter 3 introduces common practises used to implement Traffic Engineering in the Internet.
- Chapter 4 discusses related work.
- Chapter 5 shows how the Autonomous System Path (AS_PATH) attribute is being used as a TE tool in the Internet and how operations windows of ISPs can be derived from the routing information stored in routing repositories.
- Chapter 6 discusses two routing storms that occurred during 2009 and that can be traced back to the BGP-4 standardisation process.
- Chapter 7 shows how the impact of BGP-4-based TE on the evolution of the Internet's Default Free Zone.
- Chapter 8 presents my alternative routing architecture proposal. It helps creating aggregated core routing tables and allows for the TE configurations that use current common practises.
- Finally, Chapter 9 presents an outlook on future work.

This work is based on the following publications:

- *Detection of Trial and Error Traffic Engineering with BGP-4* [7], presented at the Fifth International Conference on Networking and Services (ICNS 2009)
- *On the use of Trial and Error Traffic Engineering techniques in the Internet* [8] published at the Sixth International Conference on Broadband Communications, Networks and Systems (Broadnets 2009)
- *Simple Statistical Analysis Method for the Behaviour of Autonomous Systems* [9]
- *Using RFC4893 violations to reveal the topology of AS Confederations* [10], presented in The Sixth International Conference on Networking and Services (ICNS 2010)
- *Revisiting the Impact of Traffic Engineering Techniques on the Internet's Routing Table* [11], presented at the Second International Conference on Mobile Networks and Management, (MonAMI'10)
- *A Simplified Internet Routing Architecture: Removing Traffic Engineering and Security Artifacts from the Internet's Default Free Zone* [12], a full paper developed from the previous conference paper
- *Flexible Routing with Maximum Aggregation in the Internet* [13], published at the Third International Conference on Mobile Networks and Management, (MonAMI'11)

Chapter 2

Routing in the Internet

Contents

This chapter briefly describes the administrative organisation of the Internet, how Internet Service Providers (ISPs) are identified and how they are assigned addressing space. Then, the Border Gateway Protocol (BGP-4) is discussed, including a description of the Finite State Machine (FSM) that governs the protocol, the different update types and the basics of the route decision process. Finally, the basic modules in a router are presented along with the paths traversed by IP packets and BGP-4 updates in an IP router.

2.1 The Administrative Organisation of the Internet

The Internet is the interconnection of different Internet Service Providers, who operate one or more IP network infrastructures that are known as *Autonomous Systems (ASes)*. An AS, as defined in RFC 1930 [14], is "*a connected group of one or more Internet Protocol (IP) prefixes run by one or more network operators which has a single and clearly defined routing policy*". In other words, an AS is a routing domain with well-defined routing policies. BGP-4 is the inter-domain routing protocol of the Internet. It

conveys the routing information from one routing domain to another. ASes are identified by an Autonomous System Number (ASN) that, like the addressing space, is assigned by the Internet registries.

2.1.1 Internet registries

The central authority for IP address and ASN management is the Internet Assigned Numbers Authority (IANA). The IANA has divided the globe into five regions, each of which is managed by a so called Regional Internet Registry (RIR). Prefixes and ASNs are assigned by the IANA to the different RIRs in blocks. Each RIR, in turn, assigns prefixes and ASNs within its designated geographical area from their assigned block. RFC 2050 [15] describes current best practises for IP address allocation. It has been overridden in some RIRs like Asia Pacific Network Information Centre (APNIC) after assignment of the last /8 prefix assignments on the 3rd of February, 2011 [16].

2.1.2 Autonomous System Numbers

Autonomous Systems are identified by their ASN for the purpose of BGP-4 routing. The original forecast of the Internet Engineering Task Force (IETF) was that unsigned 16-bit integers would suffice to accommodate the expected number of Autonomous Systems. It reserved the range between 64512 and 65534 for *private* autonomous system identifiers that, analogously to private IP addresses, are not to be announced to the Internet. Furthermore, ASNs 0, 54272 to 64511 and 65535 were also reserved. The success of the Internet has made this initial plan insufficient. By September 2008, only ASNs between 49152 and 54271 were free [17]. 32-bit Autonomous System Numbers as specified in RFC 4893 [18] were introduced to cope with new requests for ASNs. These new 32-bit ASNs are represented either as unsigned 32-bit integers or, for the sake of improved readability, in the form *x.y*, where x and y are the most and least significant words of the ASN represented as unsigned 16-bit integers. The original 16-bit ASNs predating RFC 4893 are represented in the format *0.y*. RFC 4893 reserves the Autonomous System Numbers in the range $\{1.0, 1.65535\}$ and 65535.65535. The remainder of the space is available for allocation.

2.1.3 IP addresses and prefixes

IP addresses identify hosts in IP networks. Currently there are two versions of IP deployed in the Internet: IPv4 and IPv6. IPv4 is the mature version and uses 32-bit unsigned integers as network interface identifiers and IPv6 is the alternative version that is progressively substituting IPv4 due to the depletion of the public IPv4 address space. This process is in its final stage. The last /8 address blocks were handed over by IANA to the RIRs on the 3rd of February, 2011. IPv6 uses 128-bit unsigned integers as host identifiers. IPv4 and IPv6 are said to be two Address Families (AFs).

As with Autonomous System Numbers, the IETF has created blocks of *private* IP addresses which can only be used in the scope of private networks and not be advertised

to the Internet. RFC 1918 [19] initially defined the private address blocks for IPv4. Another specific range for *Auto-configuration* was added in RFC 3927 [20]. Currently, RFC 5735 [21] specifies all special purpose IPv4 address ranges. IP addresses not belonging to any special category are called *public* IP addresses and can only be assigned to one host at a given moment in time. IPv6 uses the same philosophy, but other terminology. RFC 4193 [22] defines *Unique Local IPv6 Unicast Addresses*, which are equivalent to IPv4 private addresses.

IP addresses are assigned in blocks, known as prefixes. Prefixes are sets of contiguous IP addresses that are designated by a base address (B) and a mask (B_M). The mask B_M has its M most significant bits set to 1 and the rest set to 0. The most common notation for a prefix is $\mathcal{P} = B/M$. Throughout this work, the following definitions are used in relation with operations on prefixes:

Definition 1 *L_{AF} is the bit length of the AF, i.e., $L_{AF} = 32$ for IPv4 and $L_{AF} = 128$ for IPv6.*

Definition 2 *Addresses and address masks can be assumed to be unsigned L_{AF}-bit long integers. If A is an address of Address Family AF, then*

$$A \in \{0, .., 2^{L_{AF}} - 1\}$$

Definition 3 *B/M represents a prefix $\mathcal{P}$, where B is the base address and $M \in \{0, .., L_{AF}\}$ is the prefix mask length.*

Definition 4 *The prefix mask B_M has its M most significant bits set:*

$$B_M = \sim (2^{L_{AF}-M} - 1)$$

where $\sim$ is the bit-wise negation.

Definition 5 *B/M represents a valid prefix $\mathcal{P}$ if and only if*

$$B \wedge B_M = B$$

Definition 6 ***Prefix matching:*** *Let $\mathcal{P} = B/M$ and A be an IP address of the same AF as B.*

$$A \in \mathcal{P} \Leftrightarrow A \wedge B_M = B$$

Definition 7 ***Subnetting:*** *Let $\mathcal{P} = B/M$. The set of addresses contained in $\mathcal{P}$ can be divided in two subsets, known as sub-networks $\mathcal{P}_1, \mathcal{P}_2$, which contain the same amount of IP addresses. The sub-networks will have a mask of length $M + 1$. They will be denoted as $\mathcal{P}_1$ and $\mathcal{P}_2$ in the rest of this work, where*

$$\mathcal{P}_1 = B/M + 1$$

and

$$\mathcal{P}_2 = B_1/M + 1; B_1 = B + 2^{L_{AF}-M-1}$$

Prefix operations in routers

The two main entities that control the routing behaviour of a router are the Routing Information Base (RIB) and the Forwarding Information Base (FIB) [23]. The RIB indicates which is the best next hop to arrive to a given prefix. It is protocol-specific, i.e., there is a RIB per routing protocol and it includes all information needed by the routing protocol to compute its best paths. The router consolidates the information coming from all RIBs in the FIB. When a packet is delivered to a router, it uses the FIB to calculate the output link the packet has to be sent through.

When an IP packet arrives at a router, the FIB is traversed and the prefix with the longest mask that satisfies Definition 6 is looked for. Prefixes in the routing table are ordered in such a way that from two prefixes with the same base address and different mask lengths $B/M_1, B/M_2$, the prefix with the longest mask length will be looked up first. A special prefix 0/0 is used to represent the *default route* used by a router to send the packets to. If this default route does not exist in the routing table, the router will not find any feasible routing entry for some destination addresses and packets destined to these addresses will be dropped. Figure 2.1 represents the process graphically.

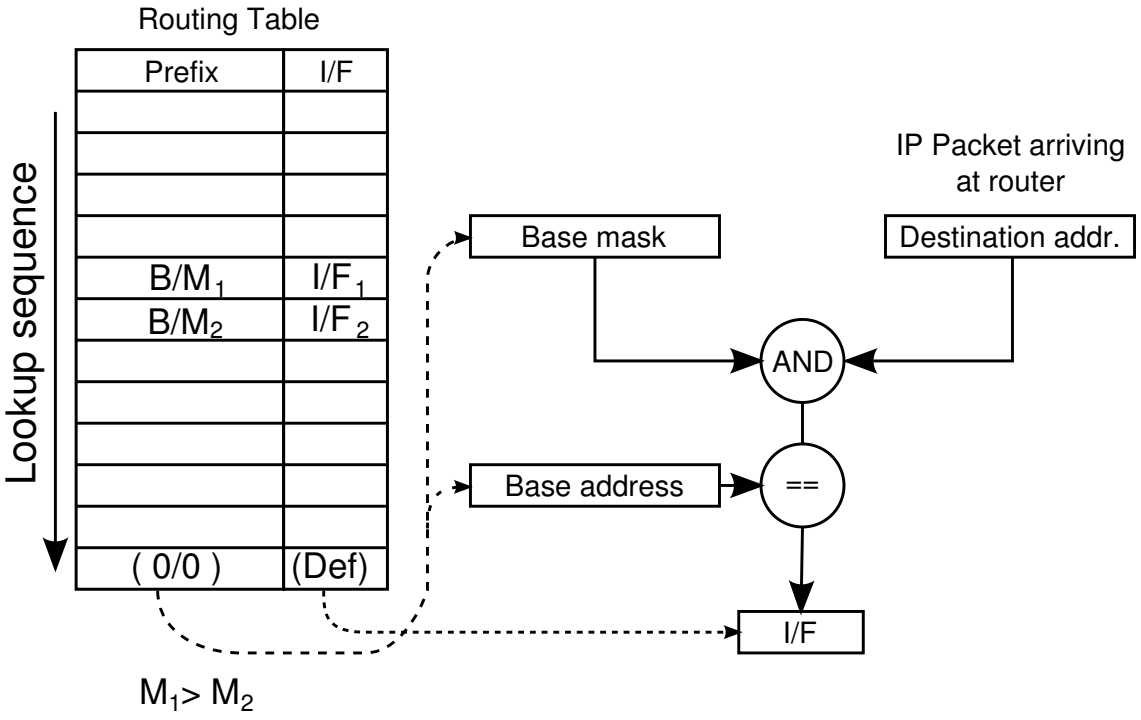


Figure 2.1: Routing table and routing process

The core of the Internet is known as the *Default Free Zone* (DFZ), because the BGP-4 routing table of the routers in this special zone of the Internet does not contain the default route 0/0. In order to guarantee that all packets can be processed, the full range of public IPv4 addresses needs to be covered by the prefixes stored in these

routers. Routing tables in the Default Free Zone (DFZ) are the longest and most resource demanding routing tables in the Internet.

2.2 The Routing Process

Routing in IP networks is a distributed process. Each router receives advertisements with reachability information regarding certain prefixes from its neighbours. This information is passed to the Route Decision Process of the routing protocol. As a result, new routes to these prefixes might be installed in the router's routing table. These new routes will be advertised to other neighbours. Routers in complex IP networks run several routing protocols in parallel, and this general principle applies for each routing protocol which is activated. The best example for this is the interaction between the Interior Gateway Protocol (IGP) or IGPs and BGP-4 in an AS. IGPs like OSPF [24] or IS-IS [25], provide quick re-convergence times in cases of failure. However, IGPs are not able to handle the amount of routing information required by the Internet's DFZ. This is what BGP-4 was designed for. Therefore, AS administrators use the IGP to control the routing within their AS and make sure that the routing information handled by the IGP, which is relevant to the whole Internet, is *redistributed* into BGP-4. BGP-4 routers follow this general principle, and also include route filters which allow to selectively modify the routing information before and after it is processed by the route decision process (see Section 2.5) and import and export functions to other intra-domain routing protocols executed by the router. These filters and import/export functions implement *routing policies*.

Figure 2.2 shows the path followed by the BGP-4 routing information in a router. As explained in Section 3.3, BGP-4 route filters compare route attributes in order to discard or further process prefixes and, eventually, modify the attributes attached to them. One example of routes which need to be discarded are routes to private prefixes in IPv4 and similar prefixes which are not routed in the IPv4 Internet [26]. Incoming updates referring to these prefixes should be ignored and outgoing updates should never be generated.

2.3 Source routing

In addition to routing by destination, the original specifications of the IPv4 and IPv6 protocols also included the possibility of *source routing*, i.e., indicating a number of nodes that needed to be traversed by a specific packet. Source routing has been disabled in Provider Networks [27]. Additionally, research on network security conducted in 2007 [28] showed that specifically crafted source routing information could result in routing loops with traffic amplification that can lead to Denial of Service (DoS) attacks. Therefore it was rapidly proposed to be deprecated for IPv4 [29] and IPv6 [30, 31]. Currently, source routing is only used in very specific scenarios [32, 33]. For all these reasons I do not consider IP Source Routing solutions in my work.

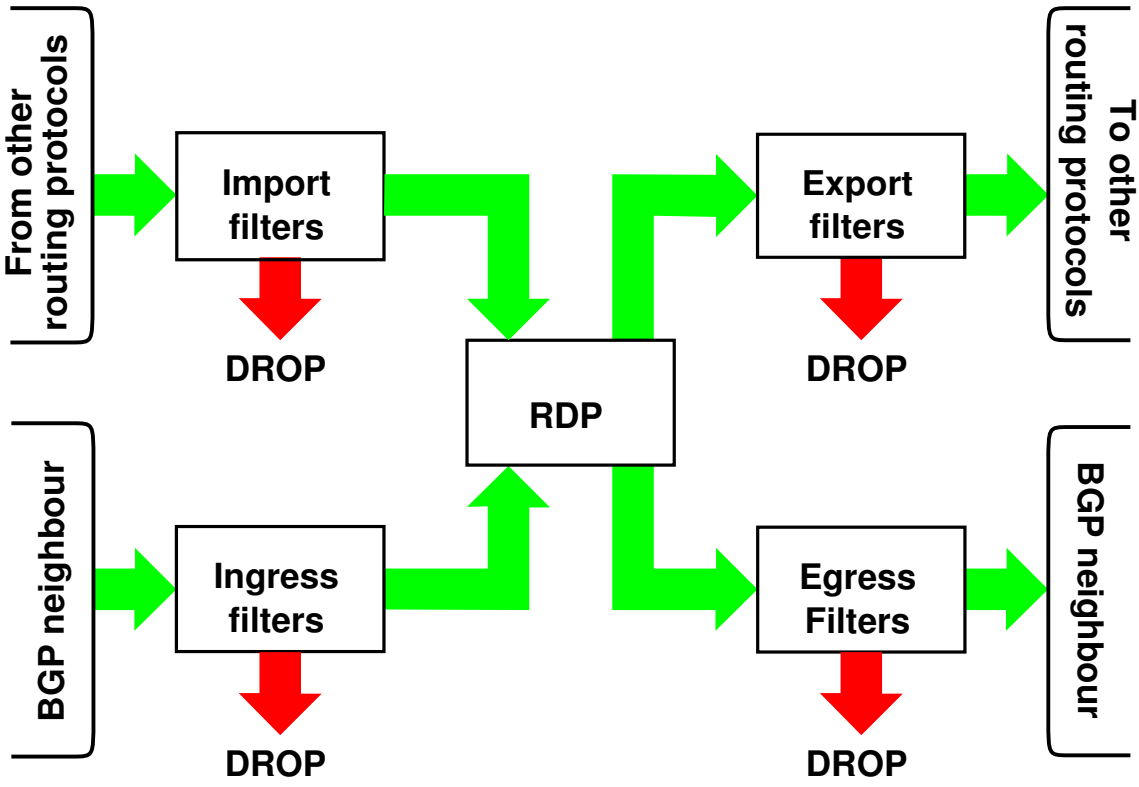


Figure 2.2: BGP-4 Routing information processing pipeline

2.4 BGP-4 protocol basics

The router or routers which connect an Autonomous System to other ASes are known as **border routers**. Routing information exchange is done by means of the *Border Gateway Protocol* (BGP-4). This subsection discusses BGP-4 as defined in RFC 4271 [2].

The two routers at the ends of a BGP-4 session are said to be *neighbours*. BGP-4 mandates the exchange of routing information to be done over a TCP session using port 179. In addition to the control mechanisms provided by TCP, BGP-4 requires explicit configuration of the neighbour's IP address and ASN, and provides mechanisms to limit the time-to-live field in the IP packets in order to provide some basic security. BGP-4 sessions between routers in different ASes are called *exterior BGP-4* (eBGP) sessions, while BGP-4 sessions between routers in the same AS are called *interior BGP-4* (iBGP) sessions.

iBGP sessions reach all routers in an AS that need to have knowledge of the Internet's routing table. These include all border routers connected to other ASes as well as selected internal routers in the infrastructure of the AS that handle transit traffic between border routers. In order to avoid routing loops, the original design of BGP-4 does not allow to redistribute routing information from one iBGP to another iBGP session. The growth of ISPs revealed a scalability problem in this design: there needs to be a full mesh of iBGP sessions between all BGP-4 speakers in an AS and this number grows as $O(n^2)$

in a full mesh. RFC 4456 [34] introduced *Route Reflectors (RRs)* as an alternative to full meshed iBGP sessions to cope with this limitation. An RR is a special router which redistributes the iBGP information to other iBGP neighbours.

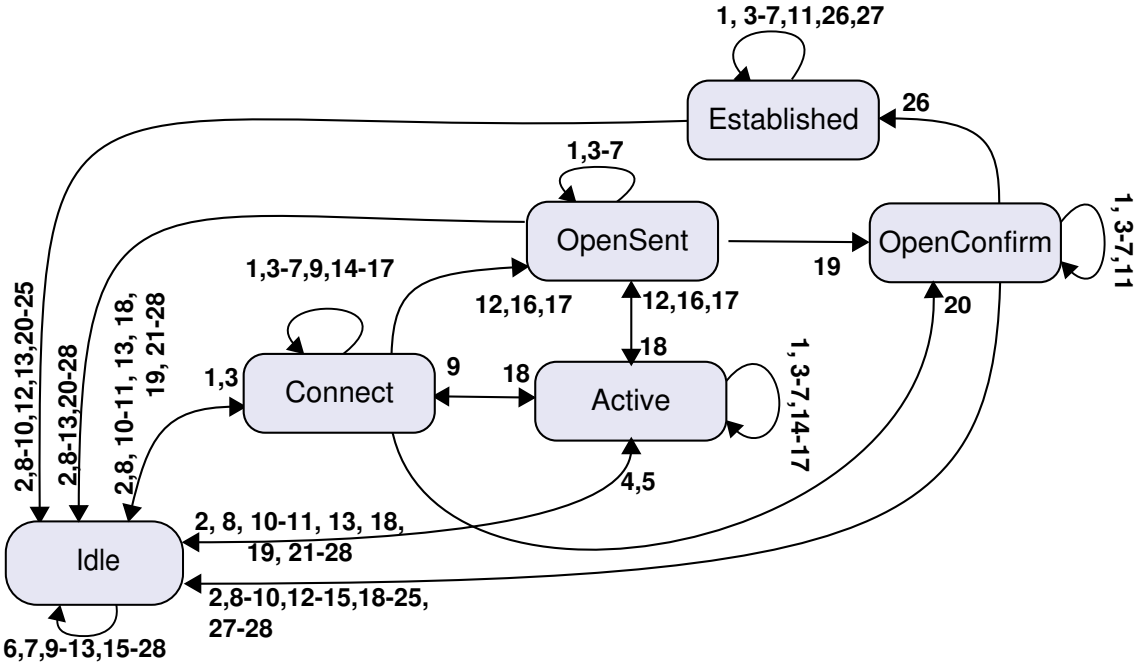


Figure 2.3: The BGP-4 Finite State Machine

Table 2.1: BGP-4 Update Types

Open
Update
Keepalive
Notification

Figure 2.3 shows the Finite State Machine controlling the BGP-4 session as defined by RFC 4271 [2]. It has 6 states. Transitions are controlled by 28 different event types that are defined in the RFC. Depending on the state of the FSM, different transitions are possible. Additionally, BGP-4 speakers exchange messages of one of the four types shown in Table 2.1. Messages of the type *Update* are only sent when the FSM is in the 'ESTABLISHED' state. They contain routing information and are referred to as *routing updates*. BGP-4 uses incremental updating: after a first massive routing information exchange to acquire the full routing tables of its neighbours, BGP-4 speakers only exchange updates to inform their neighbours of changes in the routing table. BGP-4 includes sanity-check mechanisms and when a router receives a malformed routing update, the BGP-4 session is reset, the FSM returns to the *Idle* state and the session establishment procedure has to be repeated. This process has implications on the stability of the

Internet. It is discussed in depth in Chapter 6 in the context of recent BGP-4 routing storms.

Table 2.2: Attribute categories defined in RFC 4271

Well-known mandatory
Well-known discretionary
Optional transitive
Optional non-transitive

A routing update consists of up to two fields: an *Advertisement* field containing one or more new prefixes that have appeared in the routing table and share common attributes and a *Withdraw* field for one or more prefixes that have ceased to exist in the routing table. Advertisements have attributes attached. Attributes belong to one of the four categories shown in Table 2.2. All BGP-4 speakers must recognise well-known attributes. Additionally, well-known mandatory attributes must be present in advertisements. Optional attributes may or not be recognised by all BGP-4 implementations. The attributes are used by the router in the Route Decision Process (RDP) (see Section 2.5) to decide whether to install a new route in its routing table or not. Table 2.3 shows all attributes defined in RFC 4271 and includes how attributes are used depending on the nature of the BGP-4 session. Thus, *mandatory* attributes must be used and *discretionary* attributes may or not be used. A special case is the *LOCAL_PREF* attribute, which cannot appear in updates in eBGP sessions and is mandatory for updates in iBGP sessions.

Table 2.3: BGP-4 attributes defined in RFC 4271

Attribute	eBGP	iBGP
ORIGIN	mandatory	mandatory
AS_PATH	mandatory	mandatory
NEXT_HOP	mandatory	mandatory
MULTI_EXIT_DISC	discretionary	discretionary
LOCAL_PREF	forbidden	mandatory
ATOMIC_AGGRE-GATE	discretionary	discretionary
	(depends on action to AS_PATH)	(depends on action to AS_PATH)
AGGREGATOR	discretionary	discretionary

In order to accommodate new needs, additional attributes are defined in different RFCs. Thus, for example, the AS_PATH and AGGREGATOR attributes have been extended in order to cope with the new 4-byte Autonomous System Numbers defined in RFC 4893 as discussed in Section 2.1.2. The new attributes are sent along with the old attributes. Eventually, these attributes will substitute the original attributes, once 32-bit ASN are universally supported throughout the Internet.

2.5 The Route Decision Process in BGP-4

When a router running BGP-4 receives more than one BGP-4 update for the same prefix, it has to decide whether to install it in the FIB or not. The RDP is described in RFC 4271 [2]. Figure 2.4 shows Cisco Systems' implementation of the RDP [35]. It is enhanced with information collected from other vendors (e.g., Juniper). Vendor-dependent steps are highlighted. One of these parameters is the 'WEIGHT'. The WEIGHT of a route is local to each router and thus not retransmitted. This parameter can be modified by the user and, by default, depends on the protocol that contributed the route at this step. Cisco and Juniper, for example, assign different WEIGHTs to the different protocols. Routes coming from an eBGP session are assigned a weight, routes coming from an iBGP session receive a different weight and routes which were redistributed from active IGP receive a different weight, depending on the specific protocol the route was learnt from. The multi-path decision step was introduced to allow multiple routes to a prefix to be installed in the routing table. This allows the implementation of load sharing among routes [36, 37, 38] and prevents oscillations or non-deterministic behaviour in large infrastructures with RRs [39]. The Cluster ID length comparison step is only usable in routing configurations with RRs.

Different implementations of the Border Gateway Protocol and its extensions have different default values for timers and processes specified by the RFCs. The implementation by Cisco Systems is described in [40] and a practical example of Juniper Networks' implementation of the route decision process is shown in [41]. A practical guide of how to implement communication networks mixing equipment from both vendors is provided by [42]. This book describes the differences between the two implementations. This is the major drawback of IETF as a body generating the specifications for the core protocols governing the Internet. In order to comply with its charter, which does not allow to standardise implementations, the definitions of some features may end up being very vague. This has also had a negative impact on some of the core subsystems of the Internet, like the Quality of Service components of the Differentiated Services framework in RFC 2430 [43] and RFC 4594 [44].

This chapter has discussed the basic routing principles in an Internet core router and the Border Gateway Protocol (BGP-4). These principles apply in all operations on the Internet. With the commercialisation of the Internet, different business models have arisen. The next chapter describes the basics of traffic engineering in the Internet using the mechanisms provided by BGP-4.

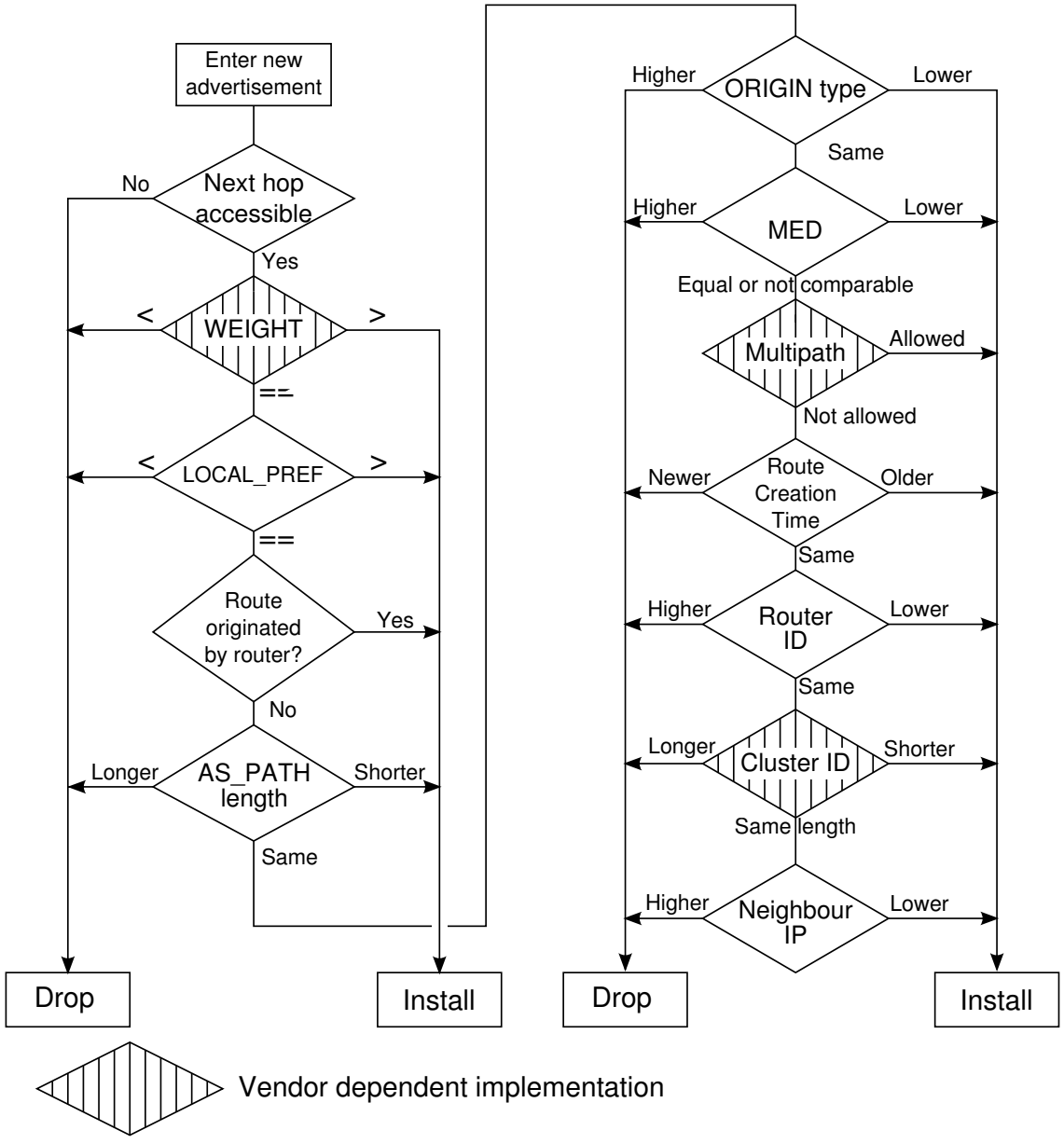


Figure 2.4: Implementation of the BGP-4 decision process by Cisco Systems

Chapter 3

Traffic Engineering using BGP-4

Contents

In the non-commercial phase of the Internet, no attention was paid to the way traffic was distributed at the interconnection links and to the costs of carrying the traffic between Autonomous Systems. With its commercialisation, this cost came into play. When the incumbent network operators started to offer Internet services, they brought in their culture and inter-operator charging models [45, 46]. This chapter analyses the impact of the commercialisation of the Internet on its routing infrastructure. As explained in Section 2.3, source routing [1] in IP networks is problematic and has been ruled out by the IETF. Therefore, traffic flows between Autonomous Systems are controlled using BGP-4 based Traffic Engineering techniques.

[1] routing using a list of hops provided by the source

This chapter describes the interconnection agreements that contain restrictions on the traffic levels, traffic pricing policies, etc. and presents the Internet's hierarchy and relationships between ASes. Then it analyses basic BGP-4 techniques used by AS administrators in order to comply with interconnection agreements they sign and techniques that can be used by an administrator to force their neighbouring ASes to comply. Most of the insights on BGP-4 dynamics have been gained by analysing BGP-4 routing traffic. This traffic has been collected at different points in the Internet thanks to the efforts of specific projects like Oregon Routeviews and RIPE's Routing Repository (RIPE RR), which are described at the end of this chapter.

3.1 Interconnection between Autonomous Systems

3.1.1 Peering relationships

There are many realms in daily life where a client pays a single provider for a service and that service can only be provided through the collaboration of several service providers. The provider chosen by the client will have to negotiate settlements with the other providers that have no direct relation with the client in order to provide that service. Internet Service Providers provide clients with access to the Internet for a fee. Internet services are spread all over the Internet and any particular ISP will need to be interconnected to other ISPs in order to be able to offer access to all Internet services to its clients.The interconnection of ISPs implies that there is some sort of financial settlement between them [46]. Financial settlements between ISPs are related with the financial settlements between traditional telecommunications operators. Regarding commercial settlements between ISPs, Huston [45, 46] and van der Berg [47] recognise three possible models of commercial settlements between ISPs. In these settlements, an AS can

1. pay another AS to access a certain set of networks or region of the Internet
2. exchange traffic with other ASes in a fair way, providing access to certain regions of the Internet, while gaining access to others in exchange.
3. get payed by other ASes for providing access to a certain region in the Internet

The rules that govern the *commercial settlements* between ISPs control which routing information is advertised on a specific BGP-4 session [46] that results in an Autonomous System assuming one of the following roles: *a*) customer, *b*) provider, *c*) peer or *d*) sibling. This classification is translated into the following rules for routing [45, 46]:

- an AS exports its own and its customer routes, but usually not its provider or peer routes to its **providers**.
- an AS exports its own and its customer routes, as well as its provider or peer routes to its **customers**.
- an AS exports its own and its customer routes, but usually not its provider or peer routes to a **peer**.
- an AS exports its own and its customer routes as well as its provider or peer routes to a **sibling**.

These rules shape the view of the AS tree perceived by each router in the Internet. They are also used by Gao [48] when trying to infer relationships between Autonomous Systems from an AS graph obtained from data extracted from BGP-4 routing tables through looking glasses [49] and similar tools.

3.1.2 Service Level Agreements

Internet Service Providers organise their interconnection through peering agreements, which include the definition of technical and economic conditions under which they exchange traffic. The technical definitions include addressing space that is made mutually accessible, the mechanisms to route traffic through the interconnection links and IP layer parameters like round-trip delay and tolerated levels of packet loss and acceptable traffic levels for the in- and outbound links. The commercial terms of a peering agreement include payment policies and Service Level Agreements (SLAs) which guarantee up-times and penalise non-conforming behaviour. Examples of current SLA for different network services including IP connectivity can be found in [50] and [51]. SLAs introduce an additional incentive for mechanisms to control the in- and outbound traffics of a network and, thus, for the implementation of Traffic Engineering (TE) techniques.

3.1.3 Autonomous System Types

The interconnection of the Autonomous Systems of the Internet is done in a hierarchical way. The *core* of the Internet is formed by a group of ASes that have a global footprint and, therefore, can reach any location without the need of hiring external connectivity. ASes at this hierarchical level in the Internet are known as **Tier 1** providers [52]. Providers directly connected to this core are known as Tier 2 providers. They, in turn, provide connectivity to other providers, and so forth. In a relationship between two ASes, the AS that is topologically nearer to the end user is said to be the *downstream* provider, while the AS that is nearer to the core of the Internet is said to be the *upstream* provider [53].

Two kinds of ASes can be distinguished: those connected to one upstream provider and those connected to several autonomous systems. ASes connected to a single upstream provider are known as *stub-ASes* and RFC 1930 [14] recommends the use of private Autonomous System Numbers assigned by the upstream provider. This kind of ASes will not appear in the Internet's core routing table as such. Stub ASes advertise a prefix or set of prefixes to the Internet. They provide connection to the Internet to end customers, i.e., they implement *Retail ISP* functionality. The use of public Autonomous System Numbers is mandatory when an AS is connected to more than one AS.

In order to be able to send traffic to the Internet, retail ISPs also need to receive routing information from their providers, but this information may be restricted to a default route as opposed to the complete Internet routing table or *full routing* that is received by transit network providers. When a retail ISP is connected through more than one link to its upstream ISP or to more than one upstream ISP, it is said to be *multi-homing*. A multi-homing ISP receives full routing from its upstream ISPs.

3.2 Automating Traffic Engineering

Routing configurations to implement TE configurations like load balancing between independent links to a major upstream ISP or load sharing between several upstream ISPs were applied from the early days of the commercial Internet. They are described in [36] and other books about BGP-4. However, predicting BGP-4 behaviour can be impossible for a network manager in a competitive environment like the Internet. Griffin and Wilfong [54] have demonstrated that BGP-4 is not always guaranteed to converge to one single solution in the presence of policies. In such situations, BGP-4 may converge to solutions other than the one intended. They also show that proving the correctness of a router configuration is an NP-hard problem.

Since the effect of configuration changes cannot always be predicted, arriving at the traffic conditions in order to comply to the SLAs signed between an AS and its peers is a process of Trial and Error: in a first step an initial routing configuration is deployed. The resulting behaviour of the AS is assessed by comparing the traffic distribution it creates in the inter-provider links to an intended one or by measuring other significant Quality of Service parameters specified in the SLA. In a third step, modifications to the initial routing configuration are computed and deployed. Then, a new cycle of measurement and reconfiguration is started. This approach is documented for academic solutions [55, 56, 57] and can be derived from the little information published for some commercial network traffic control tools like netVMG [58].

3.3 Routing policies

Routing policies are the main tool for implementing traffic engineering in BGP-4 networks. Routing policies are configured in routers and express a set of possible actions on an update that are taken depending on the route attribute values. Operators modify the BGP-4 attributes for an advertisement in order to control the routing decisions inside or outside their AS. Current implementations of BGP-4 allow most route attributes to be modified *a*) in the inbound sense before being further processed by the router, or *b*) in the outbound sense just before being sent to other routers. Filtering strategies which are present in almost all routers in the Internet can be differentiated in three classes:

1. Filters that directly influence the routing in the own autonomous system, i.e., by changing the Local Preference (LOCAL_PREF) attribute of a route depending on specific incoming route attributes.
2. Filters that pass the decision to filters in their BGP-4 peers. Routes are tagged by modifying the COMMUNITY attribute and peers take routing decisions based on the value or set of values carried by this attribute.
3. Filters that influence the routing decisions in the own and downstream autonomous systems. The attributes which are manipulated in this case are the Autonomous System Path (AS_PATH), the Multi-Exit Discriminator (MED), or both.

These techniques are described in Section 3.4 and Section 3.5.

One of the main problems faced by an AS administrator is when and how to deploy changes in an AS's routing policies because a change in the policies requires that the current routing information be re-interpreted in the light of the new policies. The initial design of BGP-4 did not foresee such a situation and recalculation was only possible with a BGP-4 session restart. This procedure affects all routes installed in a router and potentially blacked out sections of the Internet for significant periods of time [59]. RFC 2918 [60] describes the *Route Refresh Capability* for BGP-4 which allows BGP-4 neighbours to negotiate re-advertisements without the need of restarting the session. This mechanism provides the possibility of resending routes through the Route Decision Process upon modified policies and is known as the *Soft Reset Mechanism* in the Cisco Systems manuals [61]. When the route refresh capability is not implemented by a router, Cisco System routers offer the possibility of *inbound soft reconfiguration*. Routers implementing this feature construct a version of the routing table as received from the neighbour, i.e., without applying any local policies. This *clean* routing table is processed by the currently active policy filters upon request. Both mechanisms are described in [59].

3.4 Control of the inbound traffic

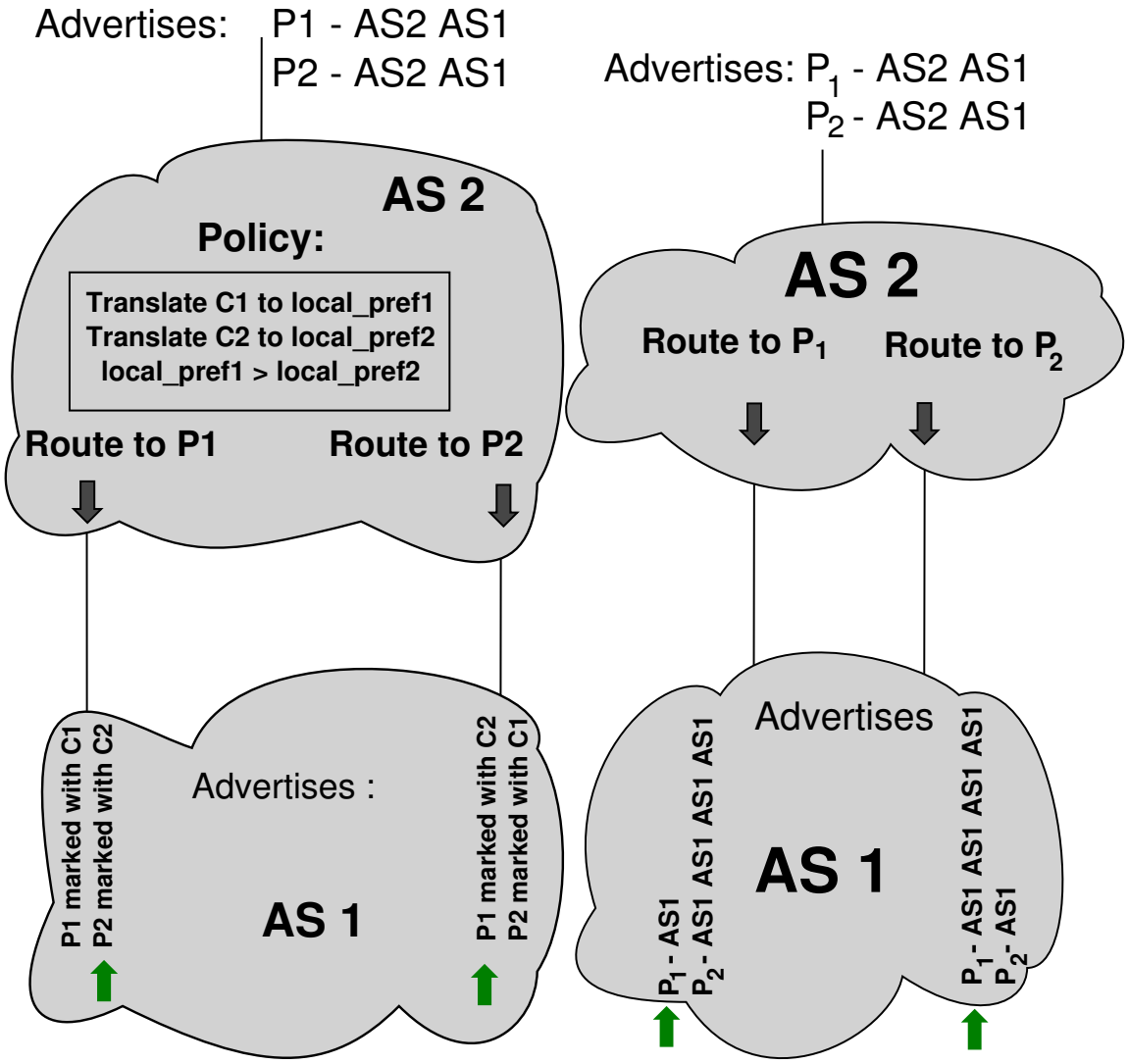
In this section, routing policies in an Autonomous System to control the inbound traffic are discussed in two scenarios. They serve as the basis for the design of the alternative BGP-4 solution for traffic engineering proposed in this work.

3.4.1 Collaborative Traffic Engineering

Current practises [36] include a marking scheme which allows the client a certain control over its inbound traffic. This technique is normally documented in interconnection agreement and allows, among others, traffic balancing. Figure 3.1a illustrates the principle behind it.

The setup is described by Halabi in his BGP-4 tutorial [62], which served as a basis for his book on BGP-4 routing in the Internet [36]. It uses the COMMUNITY attribute, as defined in RFC 1997 [63], as a marker. The client AS (AS2 in Figure 3.1a) advertises different prefixes and marks its advertisements with different values in the COMMUNITY attribute. These values are defined by the upstream AS (AS1 in Figure 3.1a). The upstream AS translates the COMMUNITY attributes into different LOCAL_PREF values within its AS and forces the routers to prefer the left link for prefix P_1 and the right link for prefix P_2. Ideally, the upstream AS should also ask its clients to send their best aggregations marked with a third community which it would use to advertise the clients further into the Internet.

A very complete survey on the use of BGP-4 communities for TE purposes is provided by Donnet and Bonaventure [64, 65]; they study the usage of communities until 2004 and how it has contributed to the Internet's routing table growth. In this article they explain the use of communities for inbound and outbound traffic control and propose a



(a) Collaborative Traffic Engineering (b) User controlled Traffic Engineering

Figure 3.1: Multi-homing to one provider

taxonomy of communities. They also identify the main drawback of communities. As a BGP-4 attribute, their meaning is local to the AS. The community attribute has four well known values, shown in Table 3.1. The community NO_EXPORT, which is used to prevent advertisements for a specific prefix to traverse the boundaries of the AS, is the most widely used well-known community [65].

This same research group has tried to standardise a use of communities that would make it easy for all ASes to understand the communities used by other ASes [66, 67]. They have proposed a new type of community, the "redistribution" community. This extended community would allow a BGP-4 speaker to influence the way a specific route should be redistributed towards other eBGP speakers it specifies.

Mnemonic	Value
NO_EXPORT	0xFFFFFF01
NO_ADVERTISE	0xFFFFFF02
NO_EXPORT_SUBCONFED	0xFFFFFF03
NOPEER	0xFFFFFF04

Table 3.1: Well known values for the COMMUNITY attribute

3.4.2 AS_PATH Prepending multi-homing scenarios

The AS_PATH attribute is an ordered list of all ASes which have been traversed by an advertisement. When a route is propagated through the Internet, each AS modifies the AS_PATH attribute by prepending its own Autonomous System Number to the received AS_PATH attribute. As explained in Section 2.5, Figure 2.4, the length of AS_PATH attribute is the third criterion used by the Route Decision Process to assign the precedence to an advertisement. The shorter the AS_PATH is, the more the route will be preferred. AS_PATH Prepending adds more than one instance of the ASN at the beginning of the AS_PATH, making it thereby artificially longer. This makes the associated advertisement less preferable to the Route Decision Process.

3.4.2.1 Multi-homing to one provider

Figure 3.1b shows a scenario where the provider (AS2) is not involved in the TE configurations of the client (AS1). This case is the most common situation, because it gives the client more freedom. The basic principle, as in the previous case, is to partition the assigned prefix or prefixes into sub-prefixes which are advertised over all links to the provider AS. The outbound policies apply different AS_PATH Prepending on the prefixes depending on the link being a primary or a backup link.

3.4.2.2 Multi-homing to more than one provider

When the main source of inbound traffic for an AS lies beyond the directly connected upstream ASes, the only reliable solution is to use AS_PATH Prepending. Figure 3.2 shows such a scenario.

In the absence of AS_PATH Prepending, a configuration like the one presented in Figure 3.2 will leave the final decision of which path to use to send traffic from AS4 to AS1 in the hands of the internal configuration of AS4. AS1 will have no control over this decision. In the configuration shown in Figure 3.2, AS4 will prefer to send traffic for prefix B/M (which is advertised by AS1) through AS2. This preference is controlled by AS1.

Figure 3.3 shows the side effects of AS_PATH Prepending. Starting from the stable configuration depicted in Figure 3.2, the two intermediate ASes AS2 and AS3 establish a direct peering. Since AS1 is a client of both Autonomous Systems, they will include it in their routing information exchange. The traffic from AS3 to AS1 is diverted

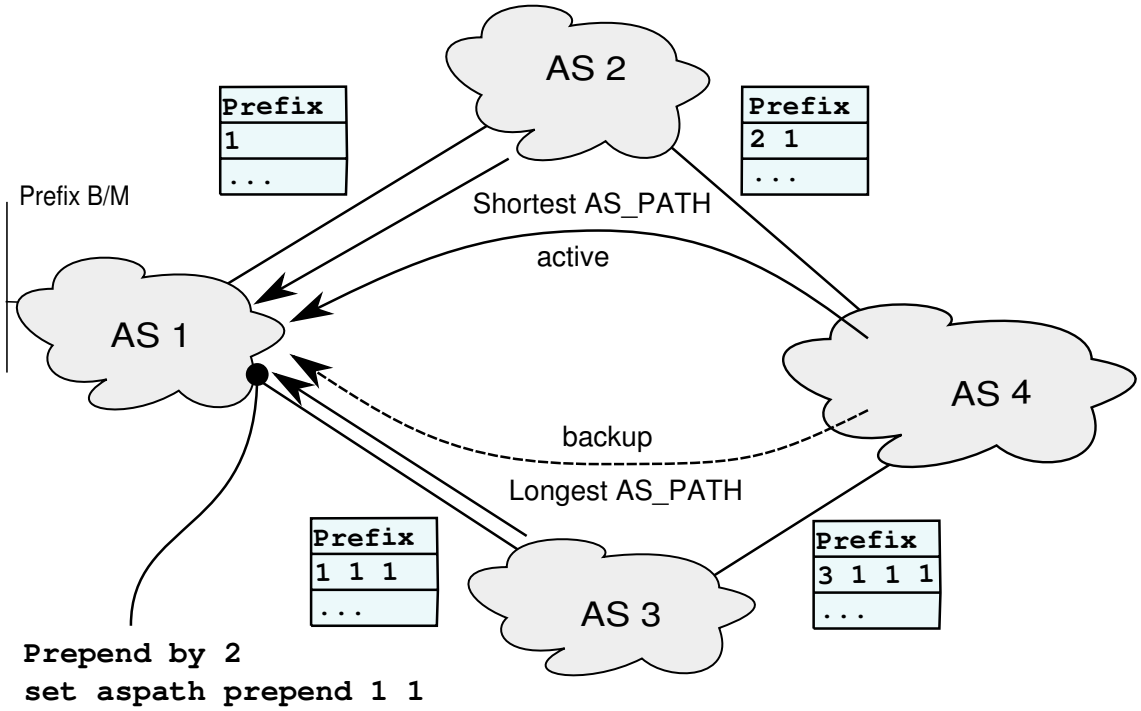


Figure 3.2: AS_PATH prepending to control traffic flows

through AS2, since the AS_PATH to AS1's prefix advertised by AS2 is shorter than the AS_PATH advertised through the direct connection. The extreme case is when AS1 is prepending just by one. AS3 will receive two advertisements with the same AS_PATH length. The RDP will have to use other attributes to select the advertisement and might eventually select the direct link between AS1 and AS3, yielding a routing configuration which is apparently correct. However, this is a hidden bug in the network. These bugs might be difficult to correct, because the network documentation will not correspond with reality.

As far as Traffic Engineering is concerned, AS_PATH Prepending is one of the most widely used techniques, as the POTAROO [68] BGP-4 studies site confirms. This technique is described in all BGP-4 manuals, e.g., [36]. AS_PATH Prepending is a popular Traffic Engineering technique, because it is able to control the behaviour of distant ASes. However, Donnet and Bonaventure point out in their study on the use of BGP-4 communities [64] that communities are being used in some ASes in order to request AS_PATH Prepending on outbound links: advertisements marked with a specific community are subject to different AS_PATH Prepending policies in different outbound links. This allows to mark the preference of the different inbound links for different network prefixes.

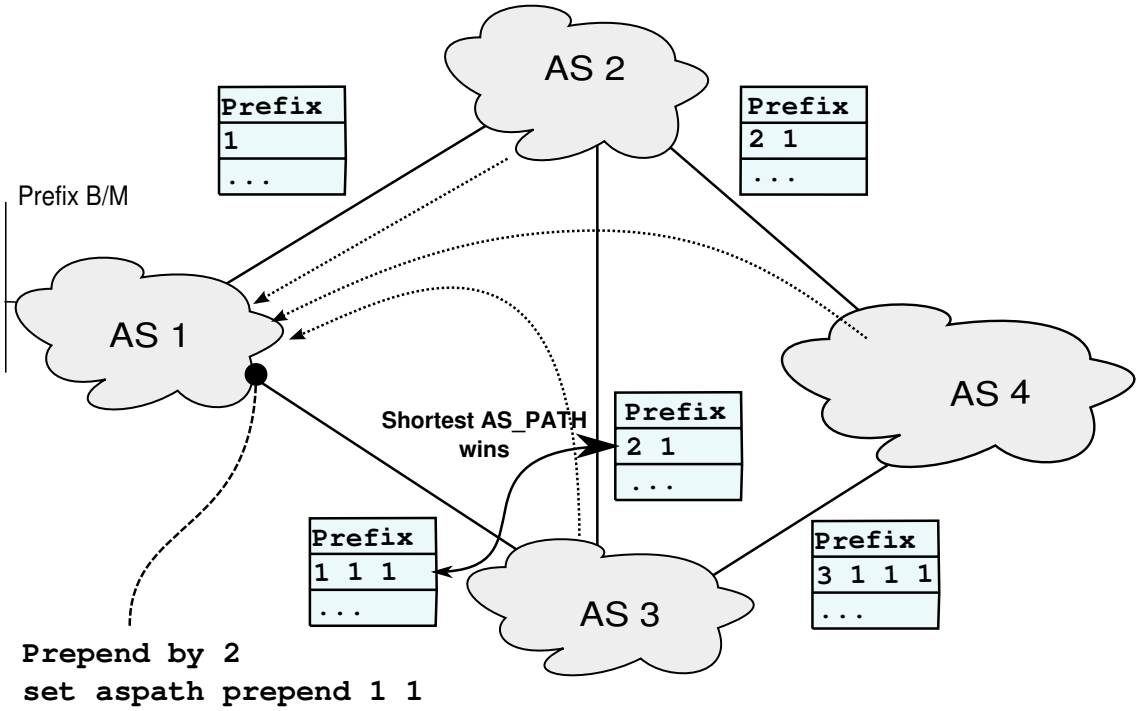


Figure 3.3: Side effects of AS_PATH prepending

3.5 Control of the outbound traffic

Controlling the inbound traffic with the techniques shown in the previous section has an effect on the advertisements generated by an Autonomous System that can be observed from the outside. Controlling the outbound traffic of an AS is implemented in a similar way. AS administrators modify the view of the Internet perceived by the AS by applying routing policies to the advertisements received from their upstream ASes. This technique is described in [69]. In most cases, the use of these techniques cannot be observed from the outside.

Tools to control the outbound traffic of an AS are described by Uhlig et al. for the specific case of a stub AS [70]. They list commercially available tools to implement Traffic Engineering and propose a device that inserts additional BGP-4 advertisements in an AS's routing infrastructure in order to control the outbound traffic. They also propose to use similar techniques to control traffic in transit ASes [55]. In their taxonomy of BGP-4 communities [64], Donnet and Bonaventure try to derive the use of the Community attribute in some configurations for outbound traffic control. While mapping communities to AS_PATH Prepending can be achieved with a relatively high level of confidence, mappings for outbound policies can only be guessed. This results in ISP administrators resorting to TE techniques based on trial-and-error, as described in Chapter 5.

3.6 Stability of BGP-4 and Routing Storms

Griffin et al. started to study the convergence properties of BGP-4 in 1999[54]. This paper presented first criteria to check whether a specific router configuration with routing policies might converge or was likely to produce divergent network conditions, i.e., oscillations. As a result of these studies, Griffin introduced the Stable Path Problem (SPP) [71]. Obradovic continued the study of the SPP and the Simple Path Vector Protocol (SPVP) [72], where he provides a first model for real-time behaviour of BGP-4 that yields an upper bound for the convergence time of BGP-4.

Several studies correlating worm attacks and increased BGP-4 activity have been published. Roughan et al. use the term **routing storm** for a "a sharp increase in the number of BGP updates exchanged between BGP routers" in [73]. In this article they conclude that a routing storm does not always impact on the data plane performance of the Internet.

More recently, Suchara et al. have resumed the study of BGP-4 abnormalities [74]. The study five sources of BGP-4 instability: route flap dampening, the Minimum Route Advertisement Interval (MRAI) timers, queueing mechanisms in the routers, clusters of routers and recent extensions to BGP-4 that were proposed to improve the convergence time and reliability. In their study, they extend the SPVP used by Griffin to capture spurious phenomena and model BGP-4 using the Dynamic Path Vector Protocol (DPVP). Their study yields the conditions under which BGP-4 converges, even in the presence of extensions.

When BGP-4 oscillates, there are different levels of impact on the Internet. In the specific case of excessive BGP-4 traffic, following effects, among others, may interfere with the performance of Internet routers:

1. **Overload** due to the level of BGP-4 traffic: the router is not able to receive and/or deliver all client traffic.
2. **Resource exhaustion**: due to the amount of BGP-4 routing messages, the router is not able to process them or to store the resulting routing information in the routing or in the switching modules.

BGP-4 routing anomalies have also been analysed by the National Institute of Standards and Technology (NIST) of the United States of America as a potential for network disruption [75]. This report also analyses other potential weaknesses and security flaws of the BGP-4 protocol.

3.7 Routing repositories

Most of the insights on how TE using BGP-4 is impacting the Internet have been obtained from extensive BGP-4 traffic collections stored in different routing repositories in the Internet. These repositories where initially established by projects like *Oregon RouteViews* [76] or *RIPE's Routing Information Service* (RIPE RIS) [3] as a joint effort of Internet Service Providers and the research community. Their initial objective was to provide a means to study the response of BGP-4 in a very large-scale deployment like

the Internet by exploring how prefixes propagated through the Internet. The BGP-4 traffic used in this work is stored in the RIPE's Routing Repository. This repository collects data from different contributing entities, including several of the major Internet Exchanges and RIPE's own Default Free Zone (DFZ).

Each entity designates a series of peers which send their BGP-4 traffic to a collecting device. Each collecting device is a modified BGP-4 router that

1. does **not** send any routing information back to its peers
2. stores all the BGP-4 updates it receives in files that cover a period of 5 minutes
3. stores every eight hours a snapshot of the routing tables of its peers.

The design of the RIPE RR is presented in [4]. As shown in Figure 3.4, the RIPE RR is highly structured. The collecting device is directly mapped to the geographical region of the Internet where the contributing entity is located. The RIPE RR files are then grouped by month and year when they were collected. This directory structure can be accessed using the Hypertext Transfer Protocol (HTTP) [77].

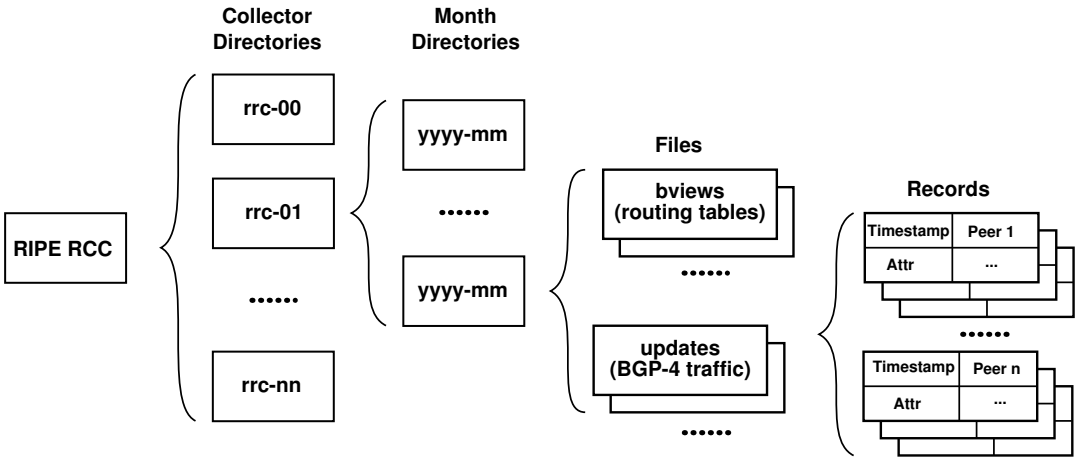


Figure 3.4: The structure of the RIPE's Routing Repository

The RIPE RR files use the so called "MRT binary format". This format was initially used in Merit's Multi-threaded Routing Toolkit (MRT) [78]. Although the programs in the MRT are completely outdated and have not been updated since 2001, the file format was adopted and fostered by the Global Routing Working Group (GROW) in the IETF. It was adopted as an RFC just before the Fall'2011 IETF meeting in Taipei as RFC 6396 [79]. The MRT file format has evolved and the RIPE RR contains files in two different formats: before the 27^{th} of March 2007, the files use the format specified in version 01 of the Internet draft specifying the MRT format. Starting that date, the files are stored in version 04 format, in order to add support for 4-byte ASNs.

The routing repositories suffer from different shortcomings. As the raw data access page of the RIPE RR [80] shows, there are two types of collectors: currently active collectors and collectors which have ceased collecting data. Additionally, as Table 3.2 shows for the case of RRC00, the collecting devices within a given collector are not always active for the whole period in which the collector is active. Another source of

uncertainty that has to be taken into account is the software on which the collecting devices rely. In 2003, Kong described different software bugs that affect the RIPE's BGP-4 collections [81]. These bugs were discovered in the BGP-4 collector frontend, which is implemented using the Zebra [82] routing software suite or its open-sourced fork Quagga [83]. Recently, Cheng et al. have studied the routing repositories [84] and developed methods to detect BGP-4 session restarts in monitors. They use the session restarts as an indicator for potential data incompleteness and apply their method to data collectors of the RIPE RR and the Oregon Routeviews. Their Web site [85] provides information regarding BGP-4 session problems for some of the BGP-4 monitors of both projects. Although the routing repositories are far from perfect, they provide a valuable source of data for the study of BGP-4. All data used in this work to analyse the different trends and threats in the Internet can be found in the RIPE RR.

This chapter has discussed the basics of traffic engineering in the Internet using the mechanisms provided by BGP-4. It has also discussed the current level of understanding of the causes for BGP-4 oscillation and routing storms. In the light of this recently gained insight regarding the conditions for convergence and stability of BGP-4, some of the Current Practises in Traffic Engineering using BGP-4 will need to be revised. This insight is also reflected in the architecture I propose in Chapter 8, where I create a basic, Internet-wide routing plane that is isolated from a local routing plane, where policies are applied.

Table 3.2: Lifetime of IPv4 collector peers for RRC00 of the RIPE-RIS Database

Peer	Start	End
193.0.0.56	01/01/2001	
195.47.235.100	10/05/2006	
168.209.255.123	01/03/2008	
217.64.144.1	01/05/2009	
91.103.24.2	01/07/2010	
46.227.200.68	01/03/2011	
12.0.1.63	23/12/2003	
193.136.5.1	25/02/2007	
218.189.6.2	01/06/2008	
203.119.76.3	01/08/2009	
202.12.28.1	01/11/2010	
212.25.27.44	01/03/2011	
202.12.28.190	01/01/2001	01/02/2011
12.127.0.121	01/01/2001	01/02/2011
193.148.15.34	01/01/2001	01/02/2011
203.37.255.126	01/01/2001	01/02/2011
195.66.224.112	01/01/2001	01/02/2011
206.251.0.85	01/01/2001	01/02/2011
195.211.29.254	01/03/2001	01/02/2011
202.12.29.64	30/11/2001	01/02/2011
212.20.151.234	01/09/2001	01/02/2011
194.109.197.245	30/10/2003	01/02/2011
213.179.39.65	29/11/2005	01/02/2011
213.200.87.254	30/01/2007	01/06/2008
203.119.0.116	01/07/2007	01/02/2011
145.125.80.5	01/10/2008	01/11/2009
208.51.134.248	01/10/2004	01/12/2007
129.250.0.232	01/01/2001	01/02/2011
192.65.184.3	01/01/2001	01/02/2011
212.20.151.253	01/01/2001	01/02/2011
134.222.87.12	01/01/2001	01/02/2011
193.148.15.85	01/01/2001	01/02/2011
212.47.190.1	01/01/2001	01/02/2011
64.211.147.146	01/09/2001	01/02/2011
192.205.31.33	28/02/2002	01/02/2011
195.69.144.34	26/09/2003	01/02/2011
168.209.255.2	30/03/2004	01/02/2011
193.138.164.1	19/04/2006	01/02/2011
145.125.80.62	01/04/2007	01/02/2011
91.103.24.1	01/04/2008	01/02/2011
195.28.164.125	01/05/2009	01/02/2011

Chapter 4

Related work

Contents

This chapter discusses other published research in the main fields covered by my work. I separately discuss routing architectures and specific error response mechanisms proposed for BGP-4. I also include an analysis of open source BGP-4 implementations I considered for my prototype implementation and some notes on available parsers for

BGP-4 traffic stored in route collectors, since I have spent some time developing my own analysis tools and all data on the evolution of the Internet I include in this work are generated by these tools.

4.1 Routing Architectures

The Traffic Engineering solutions based on BGP-4 presented in the previous chapter are static solutions which take advantage of basic techniques like address space fragmentation and attribute manipulation. Current state of the art solutions for TE in the Internet take advantage of these basic techniques but go beyond their mere application. Some of the problems that advanced techniques try to address are the depletion of the IPv4 addressing space, the growth of the Internet's routing table, and mobility. This section compiles efforts done in this field and compares them with the solutions I propose in this work.

4.1.1 Routing table compression solutions

One of the problems I address with my proposal is the growth of the routing table in the Internet's DFZ. It has been addressed by different talks in Network Operator forums like SANOG-6 in 2005. Smith gave a talk on the aggregation problem [86] explaining that there are different causes for the current state. These range from technical education that needs a complete overhaul and lack of training of the teams in charge of network operation to economic aspects like pressure to deliver services on a day-to-day basis or lack of stability of these teams. Figure 4.1 shows the evolution of the number of routing entries in the Internet's Default Free Zone between January 2001 and December 2010. Besides the sheer number of entries, each entry has a significant amount of attributes associated to it. The result is that storage space needed to accommodate the routing table of the Internet's Default Free Zone has grown significantly.

Despite this, Fall et al. [87] argue that there is no sound technological reason to compress routing tables in routers. Moore's Law is driving cost-effective scaling of hardware performance in excess of the Internet's growth, making major reductions in routing table sizes potentially unnecessary. Their router model takes computation power and memory size requirements into account and they come to the conclusion that new multi-core processor and high speed memory architectures are sufficient to keep up with the pace of growth of the Internet. While their line of argument might stand from a purely processor-oriented approach, it forgets that keeping well-aggregated routing tables helps reducing the risk of undesired routes being introduced into them and reduces the effort to debug them in case of failures.

I have explored two kinds of the routing table compression algorithms to compare them with my approach:

1. compression algorithms specific to routing tables. These try to achieve the smallest information structure representing the mapping between prefixes and their associated next hops.

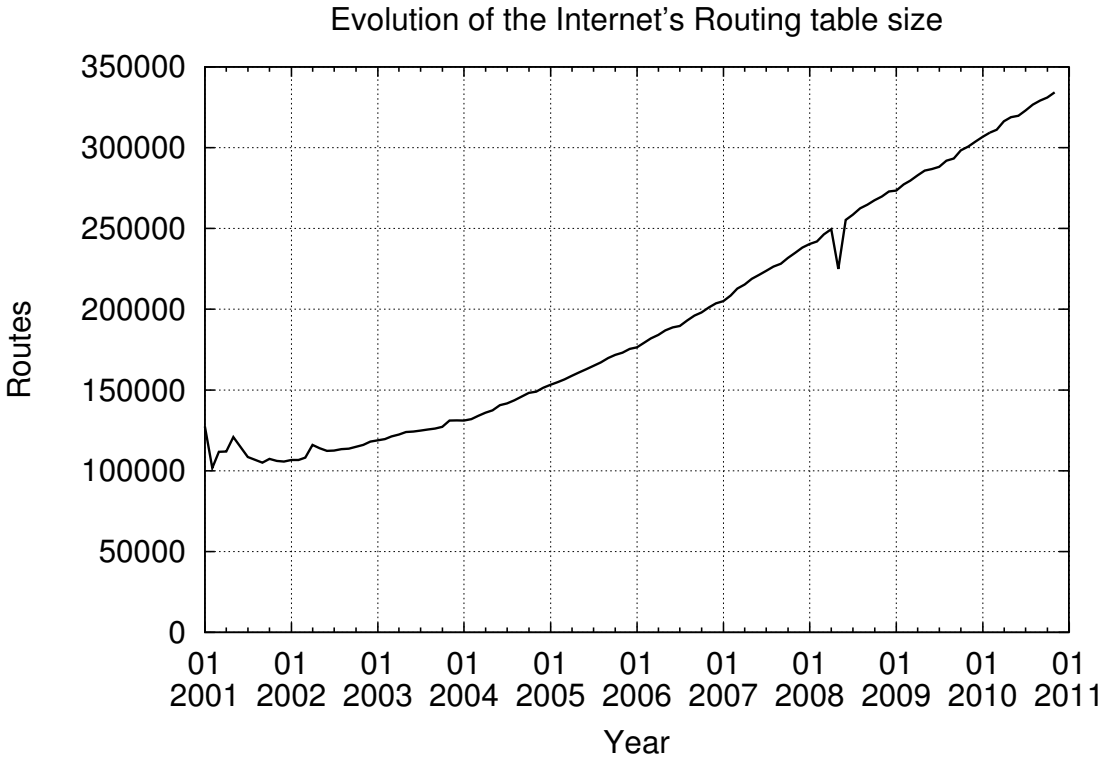


Figure 4.1: Evolution of the Number of Routing Entries in the Internet's Default Free Zone derived from data stored in RIPE's Route Repository

2. general data structure compression algorithms that can be applied to the information stored in the routing table. These take into account more factors (i.e., fields of the data structure) than the mere prefix-to-next-hop mapping in their computations.

Routing table compression algorithms like the Optimal Routing Table Constructor or ORTC [88] proposed by Draves et al. produce relevant levels of compression for longest prefix match databases relating prefixes to their next hop. However, the authors propose the use of BGP-4 to exchange routing information and the use of ORTC locally on each router in such a way that it does not affect BGP-4. This shows that ORTC is a FIB compression algorithm. In 1999, when this paper was written, the distinction between the Routing Information Base and the Forwarding Information Base in a router had not been introduced. The continued use of BGP-4 as is does not alleviate the main problem addressed by my work: the spreading of local configurations to regions of the Internet where they do not offer any additional information to the Route Decision Process and only bloat the routing tables. The situation ten years later had not changed dramatically. Ballani et al. [89] aimed at reducing the size of the RIB at specific routers in the Internet. However, they conclude that, for their purposes, FIB compression is a more realistic objective.

Generalised compression algorithms like the one proposed by Suri, Sandholm and Warkhede [90] take into account rules which apply to a combination of the source and destination address fields. Their main application field is switching devices for layers above IP, like Layer-4 to Layer-7 switches. In their reduction to one dimension, they introduce the concept of background prefix to mimic *longest prefix matching*. By doing so, they violate one of the main principles in Internet routing, i.e., that transit ASes should not create new prefixes nor suppress prefixes generated by other ASes. In my work, the responsibility of advertising prefixes lies in the AS that generates them. Their scope is fixed by selecting different advertisement paths: best aggregations through the global BGP-4 path and TE prefixes using the TE extension.

SMALTA [91] describes an interesting solution for routing table compression that its authors claim is compatible with BGP-4. This proposal has also been proposed to the IETF [92]. In order to maintain backward compatibility with all routing protocols and achieve compression at FIB level, the proposed algorithm needs to store an intermediate state which is used to derive the changes needed in the FIB when a routing update arrives at the router. This implies that extra storage and computing resources are needed in the routers implementing SMALTA. This proposal does not address the problem of RIB growth in the Internet's DFZ due to de-aggregation. In fact, a well-aggregated RIB would lessen the benefits of SMALTA. Additionally, SMALTA does not keep routing information related to configurations with limited topological scope, e.g., a traffic-balancing setup between two providers, from spreading through the Internet.

4.1.2 Alternative routing architectures

In order to cope with the explosive growth of the Internet in general and of the routing table in the Default Free Zone, there have been several proposals of modified routing architectures for the Internet in the last years. A common element in all of them is the anticipation of the deployment of IPv6. Although the lifetime of IPv4, as of writing this work, is very limited, there are still a significant number of open issues regarding a global IPv6 deployment strategy. With respect to routing, the currently proposed IPv6 inter-domain scenarios exclude the possibility of multi-homing. The proposals examined in this section all try to simplify the routing mechanisms to cope with the growth of the Internet's DFZ routing table, but do not examine the reasons for this growth. In addition to proposing algorithms to compress the 'routing table', meaning either the FIB or the RIB, some authors have gone further and proposed radically new approaches to inter-domain routing. Some approaches claim to have beneficial impact on Traffic Engineering inside the AS or ISP infrastructure in general. These approaches have been too radical to be fully deployed in the Internet's core.

Separating routing and forwarding in different devices was proposed by Feamster et al. [93]. The authors introduce a new level of routing devices in a separate control plane and make those devices control the routing in the realm of an AS, while the current routers implement a data plane which is specialised in forwarding packets. They claim that Traffic Engineering functions inside the AS can be implemented easier and in a more consistent way. However, they do not address inter-AS traffic engineering

and do not study the impact of their platform on the size of the routing table in the DFZ in the Internet.

An attempt to address the problem of inter-domain TE in Multi-Protocol Label Switching (MPLS) or Generalised MPLS (G-MPLS) networks is provided by the Path Computation Element (PCE) [94]. The PCE implements constraint-based path computation for a domain and provides the interfaces and mechanisms to extend it to a multi-domain environment, e.g., an AS Confederation or several ASes belonging to different providers. RFC 4655 describes a method to implement TE beyond a network domain, but is restricted to MPLS networks. It presents the additional drawback of introducing a single point of failure in the network. The PCE concept integrates well with inter-AS TE implementations for MPLS networks as outlined by RFC 4216 [95]. TE extensions for interior routing protocols are defined in RFC 3630 [96] for OSPF [24] and RFC 5305 [97] for IS-IS RFC 1195. None of the aforementioned approaches provides TE extensions that can be used by BGP-4. The effectiveness of all methods is increased when the granularity of the routing table is increased. In the case of the OSPF and IS-IS TE extensions, the growth of the routing table is initially localised in the domain covered by the interior routing protocol. However, these routes can be injected into the BGP-4 routing tables easily through route redistribution. Once these routes are installed in the BGP-4 table, they can easily spread to the Internet's Default Free Routing Zone. In my proposal, these routes are redistributed to the additional BGP-4 session used to exchange local routes and not to the main BGP-4 session. Thus, they are not propagated to the routing tables in the DFZ.

Cohen and Shochot [98] analyse the shortcomings of BGP-4 and propose an alternative routing overlay to which ASes can connect using BGP-4. This overlay would hide all the routing processes at a global level. The proposal enumerates the routing algorithms needed in such an overlay and the potential benefits with regards to new services that could be offered. However, it does not include a detailed design of the internals of the proposed architecture.

Virtual Aggregation (VA) [89] as a means to compress the FIB or the RIB in the Default Free Zone of the Internet is discussed by Ballani et al.. The routing space is divided into virtual aggregated prefixes and routers within an ISP are designated as aggregation points for these virtual prefixes. These aggregation points direct the traffic towards the border routers which the global Internet routing information would designate as next hops within the network of the ISP. Although this proposal can significantly reduce the size of the FIB in most routers and may ease some aspects of TE within the ISP, it does not help avoid de-aggregation as a means of implementing TE configurations between two ASes. Virtual prefixes are also proposed by Zhang et al. with their Core Router-Integrated Overlay (CRIO) [99] as a means to shrink the routing table. In their simulations, they reduce table size by an order of magnitude at the price of allowing intra-AS traffic to be routed outside the AS.When they constrain the intra-AS traffic to be routed inside the AS, the reduction obtained is less. In this case, they only achieve a reduction by a factor of 5.

VA has gained momentum in the Global Routing Working Group (GROW) of the IETF. Internet drafts on simple VA strategies [100], auto-configuration for VA [101]

and strategies to selectively suppress FIB entries when using VA [102] have been lately discussed with the general objective of producing an IETF-sanctioned VA framework. The authors claim that their solution can be incrementally and independently deployed in ASes throughout the Internet and that it neither requires changes in BGP-4 nor in the forwarding mechanisms of MPLS routers if and where they are used. They also claim that deploying the VA framework in an AS is transparent to the rest of the Internet. The additional elements they introduce to the Internet's routing architecture are what they call FIB-Installing Router (FIR) and FIB-Suppressing Router (FSR), which are responsible to generate the virtually aggregated FIBs for the AS and to suppress it when communicating with other ASes, respectively. This architecture is more complex when compared to mine and introduces an additional level of complexity when managing the network: routers inside an AS have routing information that might not be consistent with the Internet routing tables, making debugging tools deployed in the Internet (e.g., looking glasses) more difficult to use. With my proposals, the main routing tables are consistent with the Internet routing tables and those tools can be used.

CRIO [99] proposes an overlay network of core routers that use tunnelling techniques like IP-GRE [103] or MPLS between ISPs and virtual prefixes to cope with routing table growth. Tunnelling provides a clean separation between topology and addressing. Additionally, it reduces the dynamics of the routing protocol because the backbone topology is very stable and most of the dynamics in BGP-4 are introduced by changes in the peripheral topology of the network. The authors also claim that CRIO allows multi-homing to be implemented without additionally burdening the routing table of the core network and show it by applying the architecture to the Internet and to MPLS-Virtual Private Networks (VPNs) [104] with hub-and-spoke topology. Both exploit all the benefits of the CRIO architecture optimally. However, CRIO needs a new naming system to cope with the separation between network core and peripheral networks and this may be the main blocking factor for its deployment.

NIRA [105] is the acronym for a New Internet Routing Architecture proposed by Yang in 2003. The paper outlines a proposal for routing in an IPv6 network which takes into account that the evolution of the Internet from IPv4 to IPv6 should not disturb current inter-AS relationships proposed by Gao [48]. One of the objectives of NIRA is to empower the users to determine the way the packets are forwarded in a potential "New Internet". Yang claims that this level of user empowerment would steer competition, would impose economic discipline in ISPs and foster innovation. The main contributions of NIRA are a hierarchical provider-rooted addressing scheme to identify hosts and the need for end users to keep topological information of the transit networks she or he uses or intends to use. Computation of end-to-end routes is done by the traffic source based on topological information regarding the destination. This information is collected on demand and the author claims that it is thereby not necessary for all users to perform this computations. NIRA uses a hierarchical addressing scheme similar to the one proposed for the IPv6 Internet, where domains in the "core" of the network delegate addressing space to "downstream" domains which are connected to them. These, in turn, delegate addresses to their downstream domains, etc. The aspect that deserves most criticism in the NIRA proposal is the fact that packets have to carry the route specification in the

header. This represents an additional overhead on the header structure, which is not quantified in their work. Additionally, the same threats described for *source routing* in Section 2.3 apply here.

NIRA proposes a disruptive, clean-slate approach that implies a lot of non-backward compatible changes to the Internet. Some concepts are taken from proposals for the IPv6 Internet and may, therefore, be common practise in the near future. Others, however, introduce significant overhead in the communication itself or in the end devices. TE is implemented at a high price: the end devices control the path followed by the packets and therefore need to implement path computation algorithms and keep state in order to send packets. My proposal does not involve any changes in the end device or in the IP protocol and reduces the impact of TE on the network.

SIRA [106] is a proposal for a scalable Internet routing architecture proposed by Zhang et al.. In their argumentation, they identify the conflict between customers requesting Provider Independent (PI) addressing space for freedom and providers wanting to allocate Provider Assigned (PA) addressing space for aggregation and scalability purposes. This already happens, since some customers with large enough networks can request address space from the RIRs independently. They also identify that the size of the routing table depends on the product of the number of ISPs by the number of Points of Presence (POPs) per ISP by the number of customers connected to them. The other challenge for Internet scalability the authors identify is the update traffic generated by TE. When analysing the key factors for the scalability problems faced by the Internet today, they argue that the failure to differentiate between provider and customer networks is the main factor that is behind the problems.

To overcome these problems, they propose to create a core or backbone network they call Global Transit Network (GTN) that interconnects client networks. Individual hosts are identified by their host address within the client network and the point of entry into the network. However, they also state that a migration to such an architecture faces several problems. The first is that they need to establish a new mapping service for their architecture, similar to today's Domain Name System (DNS) as defined in RFC 1591 [107]. They also identify that another weak point of their proposal is that the links between the providers and their customers are neither in the provider nor in the customer infrastructure. Any failure will therefore need a special handling procedure. The main challenge of this architecture is how to code the addresses of the end hosts. Although the transition to IPv6 will provide a larger addressing space with room to code GTN and customer addresses, the way of coding the addresses for hosts in a multi-homed network is not addressed.

Separation of the core and the edge is also proposed by Jen et al., who also analyse the problem of multi-homed sites or networks in the future IPv6 Internet [108]. They claim that the current line of thought of assigning hosts in multi-homed networks one address per upstream provider is not the way to go, because many networks have already been assigned Provider Independent prefixes and transitioning to an Internet with Provider Assigned prefixes only will therefore be impossible.

Locator-Identifier Separation has been under discussion since a workshop on scalability of the Internet held by the Internet Architecture Board (IAB) in October, 2006

in Amsterdam. With current architectures, the IP address is used both as Routing Locator (RLOC) and Endpoint Identifier (EID). This negatively impacts scalability and functions like multi-homing and mobility. In order to overcome this problem, different implementation proposals for the so-called "Locator/ID (Loc/ID) split" have been presented [109, 110]. All proposals assume that the core infrastructure is efficiently aggregated, something that is not the case today, as I show in Chapter 7. The Loc/ID split can be implemented in the end systems or in the border routers. A strategy proposed to implement the Loc/ID split is to "map-and-encapsulate". From a routing point of view, this strategy proposes that the IGP handles the EIDs of a routing domain and the Exterior Gateway Protocol (EGP) handles RLOCs. The IGP is completely decoupled from the EGP, which is assumed to be BGP-4. These proposals do not take the influence of the translation between EID and RLOC into account. The standardisation state of the different proposals varies. The Locator/ID Split Protocol (LISP) [109], which implies no modifications in the host protocol stack, is supposed to be proposed as an RFC by mid 2012. The complexity of the whole LISP framework and the migration to IPv6 after the depletion of the IPv4 address space have caused some redesign in the protocol. Another proposal, the Host Identity Protocol [110] has reached the RFC status and is being proposed in the context of IPv6 and as a migration tool in the transition from IPv4 to IPv6. However, all Loc/ID solutions exhibit several architectural drawbacks [111]. e.g., all solutions rely on BGP-4 to carry the information and would be vulnerable to bogus route injection to divert traffic. My proposal also allows to construct well aggregated routing tables for the Internet's core, which is one of the main benefits promised by Loc/ID solutions, but does not require any additional translation mechanisms or patches in the network stack.

4.1.3 Traffic Engineering solutions based on BGP-4

In addition to alternative routing architectures which try to address the flaws of the current Internet routing architecture or implement inter-domain TE solutions, there has been interest in the academic and the commercial world to overcome the current limitations of BGP-4 while keeping it as the routing protocol of choice. These solutions, which are applicable to either transit or stub ASes, deal mostly with traffic optimisation. They accept the fact that they have to introduce additional information to the BGP-4 infrastructure in order to accomplish their goals and try to minimise their impact.

4.1.3.1 Academic studies and TE solutions based on BGP-4

The interest of the academic community in BGP-4 has always been great and the problem of TE using BGP-4 has been discussed under different perspectives. Regarding the design of actual TE tools for the Internet, the research group around S. Uhlig and O. Bonaventure has been very active. Based on their studies of BGP-4 Communities [112, 113], on their own proposals of how to use this BGP-4 attribute [66], and on a general performance evaluation of BGP-4 TE solutions [113], they have proposed their own BGP-4 TE solutions. When controlling the outbound traffic in stub ASes [70] they

propose to inject additional BGP-4 advertisements to control the view of the Internet perceived by the border routers. These advertisements are kept local in the AS and are not re-distributed into the Internet. They represent a small fraction of the "normal" BGP-4 traffic handled by the border routers and imply a negligible extra load for them in terms of extra processing power and memory needed to handle them. In order to generate this additional BGP-4 information, they express the target traffic distributions as an objective function and use an evolutionary algorithm that allows to both optimise the objective function and limit the number of extra iBGP messages needed.

An evolution of this principle is applied by Uhlig and Bonaventure in their *Tweak-it* tool [55]. This tool is based on sending additional BGP-4 updates that are sent to the ingress routers of a transit AS to control the inter-domain traffic over time. To this end, it computes the steady-state view of BGP-4 routing inside the AS and the traffic demands of the AS with a BGP-4 simulator and then applies a multiple-objectives evolutionary heuristic that can deal with multiple conflicting objectives that can occur in real networks.

While both solutions tend to minimise the BGP-4 traffic they introduce, they do not achieve any reduction of the size of the routing tables in the DFZ.

4.1.3.2 Commercially available TE solutions

On the commercial front, there have been some efforts to produce technologies that control the outbound traffic of an Autonomous System. These solutions are implemented by or for larger ISPs who claim to provide better SLAs to clients than their competitors. They use BGP-4 because it provides a well-known standard interface to clients. Several companies like netVmg and Sockeye, acquired [58] by Internap [114] for their own products [115], or Route-Science [116] offer or have offered "route control" services or products that influence how a multi-homed site chooses the access link to its providers. They complement the routing information and influence the next hop selection process. Therefore, these technological solutions work for the outbound traffic only. The solution I propose is conceived for controlling the inbound traffic of an AS. It can be complemented by any of these solutions, although the incremental benefits of controlling the outbound traffic should be carefully weighted against the additional management burden put on the AS. Although the internal implementation details are confidential, all solutions claim to be based on a control loop that adjusts the routing information according to some Quality of Service (QoS) objectives. In a first phase, a certain routing configuration is deployed on the infrastructure. Then QoS probes are activated, measurement data is collected and evaluated, and eventually corrections on the routing configuration are flushed into the network.

4.1.4 Higher-layer traffic engineering solutions

In addition to pure BGP-4 developments, there have been lately some efforts in the IETF to offload TE tasks from routing and move them from the network to the terminals. These solutions move away from the IP layer and try to control the traffic at the

application layer. In this line of thought, there is a lot of activity around the Application-Layer Traffic Optimisation (ALTO) architecture proposed in RFC 5693 [117]. ALTO is an effort to control peer-to-peer (P2P) traffic in order to reduce peering costs for the ISPs by localising it as much as possible within the ASes. In order to do so, the ISPs participate in the P2P overlay and deploy so-called "tracker" devices. These devices help the end terminals decide where to retrieve the contents and direct the terminals towards those content sources that produce a lower strain on their network, for example, by optimising peering costs. In recent proposals, the trackers use routing information coming from the IGPs and BGP-4 to calculate the optimal content source.

These solutions radically differ from my approach in that they involve the end device. User intervention is needed in order to install an ALTO-compliant P2P client in their terminals and this is the weakest point. Since a significant amount of users will not be convinced about the solution, a mixed ecosystem is to be expected and the benefits in terms of gains from optimised traffic versus complexity of managing the solution and unexpected interactions in the network are still to be further investigated. Simulations show promising results [118] but feedback from realistic trials is still missing. My proposal is completely transparent to the user and does not need any sort of user intervention and yields a much more controllable network.

4.1.5 Debugging Network Configurations

My work is also significant when it comes to operating the Internet. As shown in Section 7.1, fragmentation of the addressing space is significant. Fragmentation makes the overall operation of the network more difficult because it increases the amount of information which has to be processed in case of a failure. There have been efforts to produce automated network debugging solutions or tools to help the operator debug complex network failures [119]. The aim of my work is to clean up the Internet's DFZ routing table, in order to make the problem which has to be tackled by these systems smaller and more manageable.

The other problem arising from complex routing tables is that they make the Internet more susceptible to attacks like prefix hijacking [120], as shown by projects like INTERSECTION [121]. A cleaner routing table in the DFZ would also allow simpler and quicker implementations for tools that detect prefix hijacking like PHAS [122] and simplify best common practises to handle illegal advertisements [26]. This problem is described in a more general way by Feamster et al. [123] when analysing control plane security properties of BGP-4.

In addition to addressing the operative problems of debugging possibly erroneous network configurations, my work also addresses Internet storms related with the reaction of BGP-4 to malformed packets. The related work in this field is discussed in the next section.

4.2 Alternative BGP-4 error handling

As presented in Chapter 2, BGP-4 reacts very drastically when errors are detected in updates: it resets the BGP-4 session. This is at the origin of BGP-4 routing storms detected in 2009: a peer was advertising a prefix with a malformed attribute. This caused a session reset after which routing information was exchanged again. The prefix with malformed attributes was received again and the cycle started all over again. To the outside world, this incident appears as a route flap for all prefixes advertised by the peer advertising malformed routing information. Currently, there are two different philosophies how to handle updates with malformed updates while mitigating the risk of routing storms: either denial (i.e., not recognising that the packet is malformed) or partial reset (i.e., only resetting the BGP-4 session for the specific Address Family where the malformed packet was detected). Currently, infrastructures with MPLS-based Virtual Private Networks (VPNs) and IPv6 Internet services running over the same infrastructure as IPv4 are common. This means that they use a common BGP-4 session to exchange routing information and errors in one Address Family affect the others. Therefore, there has also been an attempt to isolate the different Address Families in different BGP-4 sessions to avoid that an error in a given Address Family affects the whole routing infrastructure and service offerings of an Autonomous System.

4.2.1 Denial: handling confederation data in the AS4_PATH attribute

The transition from 2-byte to 4-byte ASNs has been at the root of at least one big routing storm in 2009. The results from the analysis by Watanabe [6] show that updates where the 4-byte AS_PATH (AS4_PATH) attribute contained one or more AS_CONFED_SEQUENCE field were interpreted by some routers as invalid and the affected BGP-4 sessions were reset. This corresponds to the behaviour that had been defined originally in RFC 4893 [18]: *"To prevent the possible propagation of confederation path segments outside of a confederation, the path segment types AS_ CONFED_ SEQUENCE and AS_ CONFED_ SET from RFC 3065 [124] are declared invalid for the AS4_ PATH attribute."*

As a response to this incident, the behaviour of a router when receiving this kind of packets was changed in a new version of RFC 4893 [125]. The consequence of this change was that the AS confederation information was leaked to the Internet. This incident is analysed in more detail in Section 6.1. Furthermore, with different Internet Service Providers using different operating systems and even different versions of the same operating system within the network, the risk that malformed packets can induce new routing storms is not completely banned.

4.2.2 Error detection for optional transitive attributes: 'treat as withdraw'

Another attempt to treat malformed BGP-4 updates was presented to the IETF [126]. This Internet Draft specifies the 'treat as withdraw' behaviour for optional transitive attributes only and does not define any backward notification mechanism to peers that originate or relay the incorrect information. It just requires that the peer discovering the malformed attribute or attributes should have logging facilities to record the offending instances. In my work, 'treat as withdraw' is always applied and I propose to send error notifications to upstream peers in addition to logging the events that might be triggered by the router detecting the error. This increases the probability of detecting it.

4.2.3 Enhancing the inter-protocol isolation in Multiprotocol BGP-4 environments

BGP-4 is being used as the inter-domain routing protocol of choice for protocols other than IPv4. This has been possible with the Multiprotocol Extension for BGP-4 (MP-BGP) specification. MP-BGP transports the routing information of *all* address families over a common BGP-4 session. This results in massive disruption to all communication when an error condition, like an invalid BGP-4 update, forces the BGP-4 session to be reset.

In the year 2005, Nalawade, Patel et al. proposed a soft notification mechanism in case of reception of corrupted packets [127]. This proposal describes a mechanism that allows to notify a remote peer of an error-condition or an event without resetting or terminating the BGP-4 session. However, the proposal only provides the ability to soft-reset for a particular AF without disrupting the other AFs carried by the BGP-4 session. The main drawback of this proposal is the complex mechanism proposed to handle soft notification messages, which almost always results in a BGP-4 session reset.

Another strategy to overcome this problem was proposed to GROW in the IETF in the "Multi-session BGP-4" draft [128]. Multi-session BGP-4 proposes a mechanism that allows multiple BGP-4 sessions to exist between two BGP-4 speakers. Each isolated BGP-4 session can then carry the routing information of a specific Address Family. Although this approach would provide finer-grained fault management and isolation, it would not suppress the routing storms which have occurred during 2009, because they do not specify any way to avoid a faulty packet to be leaked into the Internet. In my practical implementation, I go a step further and use independent daemons (and hence protocol sessions) to isolate TE activity from the regular Internet routing tables. However, multi-session BGP-4 would also be a valid option to transport the TE-related routing information within the proposed architecture.

4.3 Comparison between my work and the related work

I present, as many other people have done, an attempt to improve BGP-4 in order to provide a more reliable Internet. The differences between other approaches and mine, presented in Section 4.1 and Section 4.2 can be condensed in the following comparison table:

Table 4.1: Comparison between the related work and my proposal

Proposal	Argument Drawback	My answer
Fall et al. [87]	No need for table compression, Moore's Law provides computing power to accommodate the growth of the routing tables	Well aggregated tables reduce the risk of misconfiguration and the effort debugging faulty routing tables.
Optimal Routing Table Constructor [88]	Use BGP-4 as is to remain backwards compatible and use ORTC locally in the router to reduce the FIB	Aggregating adequately at the AS level reduces BGP-4 information exchange and helps generating well aggregated routing tables in the DFZ.
Suri, Sandholm and Warkhede [90]	When applied to BGP-4, this algorithm generates new aggregations in intermediate ASes, leading to potential routing loops.	Aggregation is a responsibility of the ASes that advertise prefixes to the Internet and not of intermediate ASes in order to avoid routing loops.
SMALTA [91]	Achieves optimal FIB compression but does not help reducing the size of the routing table	Well-aggregated routing tables produce the same benefits as resource-intensive FIB compression algorithms

Continued on next page

Table 4.1: Comparison between the related work and my proposal

Proposal	Argument Drawback	My answer
Feamster et al. [93]	Separating routing and forwarding in different devices would make intra-AS Traffic Engineering easier but does not address the inter-AS TE case and impact is on routing table size is not known	I address inter-AS TE configurations and provide an estimate of the reduction of the routing table size in the DFZ
PCE [94]	Specialised for MPLS networks, introduces a single point of failure and is tailored for IGPs. Routes for TE that leak into BGP-4 make the routing tables grow.	Designed for IP. Keeps routes associated to TE isolated from main routing table.
VA [89]	No means to avoid de-aggregation as a means for TE	TE tools provided while keeping the DFZ well aggregated.
CRIO [99]	Could route intra-AS traffic through external ASes	Keeps intra-AS traffic inside the AS.
NIRA [105]	New packet format: data packets carry route specification in the header. Severe security risks exhibited by IP source routing	No modification in the packet format.
NIRA [105]	Disruptive, clean slate and non backward-compatible	Backward-compatible, evolutionary approach
SIRA [106]	Multi-homing clients are not addressed	Architecture for multi-homed ASes

Continued on next page

Table 4.1: Comparison between the related work and my proposal

Proposal	Argument Drawback	My answer
LISP [109]	Transition from IPv4 to IPv6 not addressed correctly	IP version independent
TE for stub-ASes [70] and *Tweak-it* [55]	Does not address the growth of the routing tables in the DFZ	I help to aggregate the routing tables in the DFZ better.

4.4 BGP-4 protocol and routing storm analysis

In my work, I perform BGP-4 update sequence analysis. I observe several situations were the update traffic is abnormally high and explore which the techniques applied by network administrators can be identified in the BGP-4 update traffic. These results help me identifying weak points of the protocol that I need to overcome in my architecture.

Dolev et al. study the robustness of the Internet and the BGP-4 protocol [129]. They compare the different techniques used by ASes and conclude that BGP-4 resilient AS interconnection is mainly achieved by multi-homing followed by the use of a main and a backup link.

Programs crafted with malicious intent have caused disruptions in the Internet infrastructure. Li et al. [130] analysed these disruptions and identified patterns of BGP-4 traffic they could link with different documented attacks and were able to detect new attacks on the Internet's infrastructure using those patterns. Similarily, they were able to predict the effect of natural disasters on the BGP-4 infrastructure based on previous disruptions.

A general BGP-4 protocol security analysis was provided by Kuhn et al. in [75] for the National Institute of Standards and Technology. In this paper, they analyse the protocol and list the potential sources of instability that are intrinsic to it.

Li et al. study BGP-4 protocol dynamics in [131]. This study shows that in the first decade of the 21st century, BGP-4 was more stable than previously, identifies ASes that act as main BGP-4 traffic sources and shows that BGP-4 traffic generated by ASes is not related with their size.

Huang et al. [132] propose a method to detect network disruptions using BGP-4 traffic, using network configuration information in addition to the BGP-4 traffic.

Zhang, Quoitin and Zhou have recently published a comparative study of the evolution of the IPv4 and IPv6 Internets [133]. They provide some insight into the evolution of the number of ASes that are active on both networks. Since the objective of my analysis is to provide a case for an evolution of the access infrastructure to parallel routing

architectures, my analysis of the Internet evolution has concentrated on the evolution of the leaf ASes and their use of traffic engineering techniques like AS_PATH Prepending.

Regarding BGP-4 analysis, Khosla et al. propose to group BGP-4 updates in sequences they call "molecules" in order to analyse BGP-4 prefix behaviour [134]. This work was published two years after my initial papers on Trial-and-Error Traffic Engineering patterns [7].

Renesys [135] is a private company that has specialised in BGP-4 consulting. They maintain a blog where experts analyse routing incidents, e.g., [136, 5]. I use this information to confirm the existence of major BGP-4 events and contrast their findings with my own.

4.5 Implementation alternatives

I have implemented a proof-of-concept of my routing architecture on a limited scale, in order to confirm that it is implementable and reasonably safe to deploy at a minimal scale in the Internet. This section describes the different alternatives with regards to emulation environments, simulation environments and BGP-4 source code bases I have considered for my work.

4.5.1 Network emulation environments

Network emulation environments provide a testbed for debugging implementations of routing protocol suites. They are virtualised environments and provide an independent execution environment for each emulated machine. Their main weak point is the way they emulate network links and introducing delay or jitter as in real-world deployments increases the amount of resources they demand from the host system.

Network emulation environments considered for my work can be grouped in two categories: emulators running locally and remote emulation testbeds. Remote emulation testbeds like Emulab [137] in the U.S.A. or German-lab [138] provide network infrastructures to conduct networking experiments. By providing full control over network nodes, experiments that imply high node load benefit from them. However, they require an Internet connection in order to conduct the experiments and I knew that I would not have Internet access for longer periods when performing this work. The focus of my work was to implement the routing architecture and test the functionality, not the performance.

Therefore, I decided to go for an emulation environment I could use locally on my machine.

4.5.1.1 Netkit

Netkit [139] uses User-Mode Linux [140] as a virtualisation technology and runs Linux router images based on Quagga [83] only.

4.5.1.2 GNS3

GNS3 can use different virtualisation frameworks like QEMU [141], VirtualBox [142] to execute guest operating systems like Linux or Windows and Dynamips [143] for selected images of major network equipment vendors like Cisco or Juniper.

4.5.1.3 Selection of the emulation environment

I finally chose **Netkit** because I had prior experience with it. GNS3 is certainly attractive because it is able to execute actual vendor operating systems, but since I introduce new features in BGP-4, I would have needed access to the source code for these images and this is currently not possible.

4.5.2 Open source BGP-4 implementations

In order to implement the proof-of-concept of the routing architecture I propose, I had two possibilities:

1. implement BGP-4 from scratch. This would have meant a new development with a significant amount in debugging in order to provide a solid implementation that would not falsify the results.
2. reuse an existing implementation. For this purpose, there are different open source router implementations that provide implementations of the BGP-4 protocol, which have been proven in real-word deployments.

The objective of my work is to implement an architecture that uses BGP-4 in a new way, in order to reduce the routing table size. It has to be evolutionary and backwards compatible. Therefore it was not necessary to re-implement BGP-4 from scratch, but rather add the required features to an existing implementation.

Following implementations were available when I started my work:

4.5.2.1 Merit's Multi-threaded Routing Toolkit

Merit's Multi-threaded Routing Toolkit [78] was one of the first implementations of a routing protocol suite that could be executed on off-the-shelf equipment. Development ceased in the early 2000's and it's presence in the Web disappeared 2010. The most significant and lasting result of this effort is the MRT binary format specification, which is maintained and enhanced by the Global Routing Working Group (GROW) of the IETF.

4.5.2.2 Zebra

Zebra [82] is an open-source implementation of a router supporting different routing protocols like RIP [144], OSPF [24] and BGP-4. This routing suite uses a command-line interface that is inspired by Cisco Systems' command-line interface. It is implemented using the C programming language. The last stable version of the source code was released September of 2005.

The Zebra routing suite is based on a central daemon (*zebra*) that communicates with different daemons that implement routing protocols on one side and with the kernel routing tables. The communication with the routing daemons is based on a simple protocol, known as the Zebra protocol [145]. The interaction with the kernel routing tables is implemented with the Netlink protocol. Figure 4.2 shows a block diagram of the different components in the Zebra routing protocol suite.

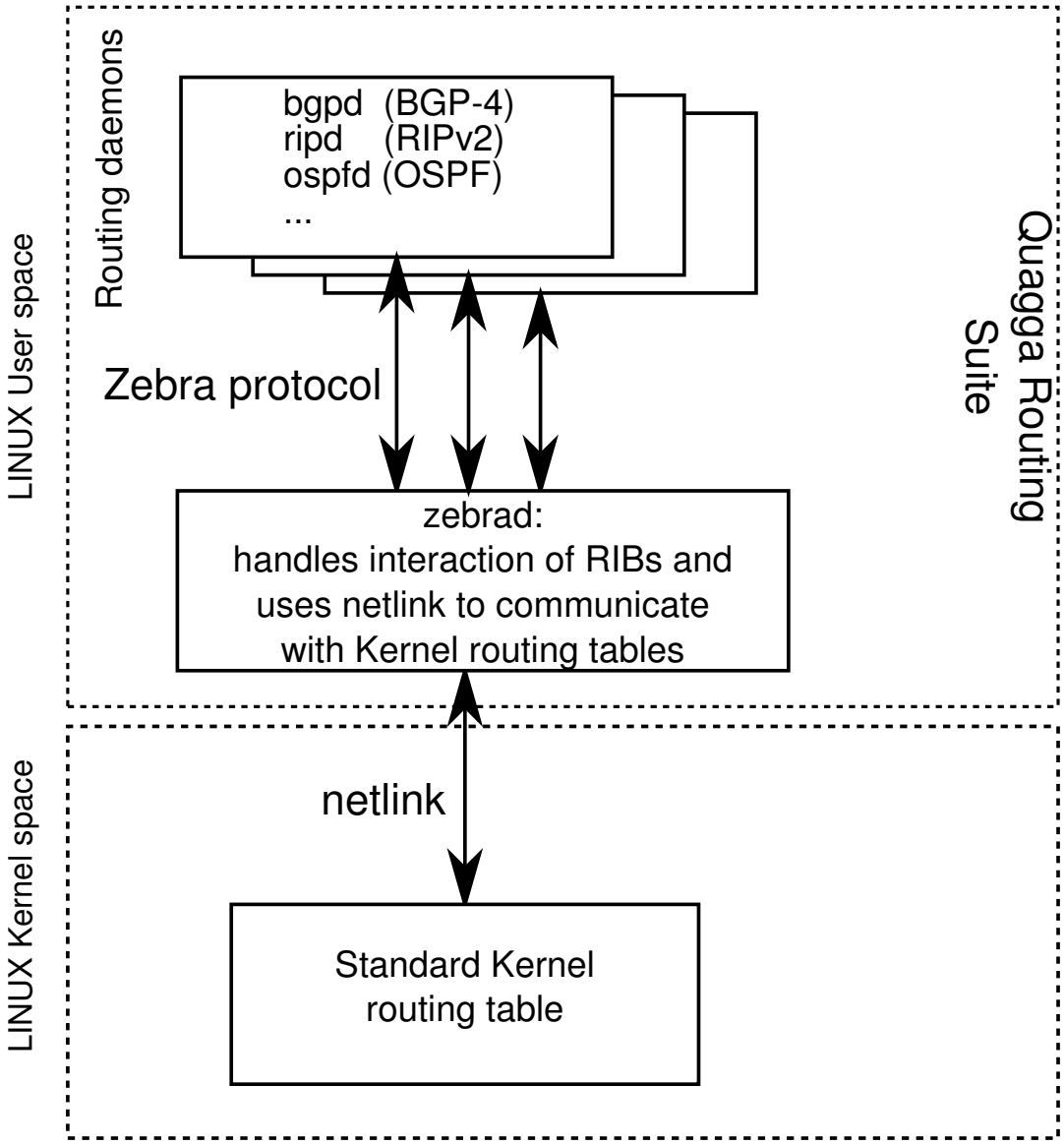


Figure 4.2: High-Level View of the Zebra/Quagga routing protocol suite on deployed on a Linux platform

4.5.2.3 Quagga

Quagga [83] is a fork of the Zebra routing suite. This project is currently active and has an extended base of supported routing protocols. It has an active community of maintainers and developers who correct implementation errors. This code base is deployed in many routers in the Internet, and runs on Solaris, BSD and Linux based Operating Systems.

4.5.2.4 OpenBGPD

Zebra and Quagga have been mainly developed by the Linux and Solaris communities and are licensed under the GPL license. This license is incompatible with the OpenBSD operating system. A community of programmers developed a BGP-4 routing daemon with free licensing and stressing security and robustness. This development can be downloaded from the OpenBGPD [146] project's website. This daemon is an independent effort, i.e., no routing protocol suite, although the companion project OpenOSPF, also available from the OpenBGPD site, is developing an OSPF implementation under the same model.

4.5.2.5 XORP

The eXtensible Open Router Platform [147] is another project that is developing a router platform that can run on commodity hardware over a number of Operating Systems. It implements additional functionality like GRE tunnelling [103]. This routing suite has GPL licensing and provides a command-line interface akin to the JunOS-command line interface (CLI) [148].

4.5.2.6 The Bird Internet Routing Daemon

The Bird Internet Routing Daemon [149] is another Linux-oriented routing protocol suite developed by the Charles University in Prague, Czech Republic. It implements multiple routing tables and has a command-line interface akin to the JUNOS CLI. This development started 2005 and was dormant for about 4 years. The development activity ramped up after I had already implemented a first working prototype of my parallel routing architecture.

4.5.2.7 Selection criteria

When I planned the development, I was looking for an open source project that was alive, provided a reliable BGP-4 implementation in a code base that could be executed on a Linux environment. Table 4.2 shows the different strengths and weaknesses of the examined BGP-4 implementations. Based on this table, and due to my experience in operating Cisco network equipment, I finally chose to use Quagga.

Table 4.2: Comparison between the different routing suites

Routing Suite	Strengths	Weaknesses
MRT	Very simple code	Incomplete BGP-4 implementation Old code base Abandoned project
Zebra	Complete BGP-4 implementation Well-known development environment Cisco CLI	Closed source
Quagga	Complete BGP-4 implementation Live project, Open Source Well-known environment Cisco CLI	
OpenBGPD	Stable code Linux support Some experience	Incomplete implementation Focusing on FreeBSD
XORP	Apparently feature complete Live project, Open Source	Unknown environment Juniper CLI
BIRD	Apparently feature complete Live project, Open Source	Unknown environment Juniper CLI

4.5.3 MRT binary format parsers

I have developed the software to parse the MRT binary format files from scratch, using the Java programming language and the reference documents circulated in the Global Routing Working Group in the IETF. Réseaux IP Européens (RIPE) provides a reference implementation of an MRT binary format parser in C [150]. When I started my work, this implementation was not quite stable and threw exceptions when parsing IPv6 related information.

There is also an MRT format parser written in Perl by Rossi et al. The MDFMT toolkit [151] was started after I had already worked intensively on my Java toolkit. A

version of libbgpdump written in Python [152] was released when I started working on my parser. It was even less mature than libbgpdump and seems to be abandoned after it was released, since no updates have been released after January, 2007.

Chapter 5

BGP-4 Update Sequence Analysis

5.1 Introduction

This chapter collects my work on BGP-4 update sequence analysis. This work was published in ICNS'2009 [7], Broadnets'2009 [8] and ICNS'2010 [9].

I started this work to find traces of the activity of automatic routing optimisation products like netVmg (integrated at least in the Internap service offering [58]) and of human intervention on the routing infrastructure of the Internet. This work led to discovering instances of instability that affected some prefixes. These instabilities or *routing storms* are undesired situations from the network operations point of view and are therefore good candidates to contain update sequences that show how network operators work to mitigate them. Using these update sequences, I could also show how the affected ASes tried to route around the instability by artificially increasing the AS_PATH length, i.e., using AS_PATH Prepending techniques. Further on, I looked at the arrival times of updates for a prefix in general. This helped me single out cases of ASes with a specific operations time window and develop techniques to determine this time window.

5.2 AS_PATH Prepend Sequence Analysis

The view of the Internet from one specific AS is limited and it is impossible to compute the network paths between all possible traffic sources and destinations from the viewpoint of a specific AS [70]. For this reason, it is also impossible to predict accurately how a routing configuration applied in an AS will affect the overall routing behaviour in the Internet. Therefore, the routing policies that translate into routing configurations at the AS-level are applied on a trial-and-error basis: based on current conditions, a policy is designed and applied and the resulting traffic profiles are observed. If the expected traffic profile is not met after allowing the routing a certain time to settle, the policies are changed [57]. This operational procedure applied by an AS that uses AS_PATH Prepending to control its inbound traffic can be observed from the outside as a sequence of updates for certain prefixes that are sent every time the routing policies

are modified. The AS_PATH attribute in these updates will have fluctuating length, depending on how many times the Autonomous System Number is prepended. When using AS_PATH Prepending to control the traffic, an AS will withdraw a prefix from the Internet under the following conditions:

- when a policy leads to an unintended routing configuration and needs to be removed
- when the prefix is not needed for a given configuration
- when a prefix has been reassigned to another AS

The first two conditions normally lead to unwanted traffic distributions in the inter-domain links.

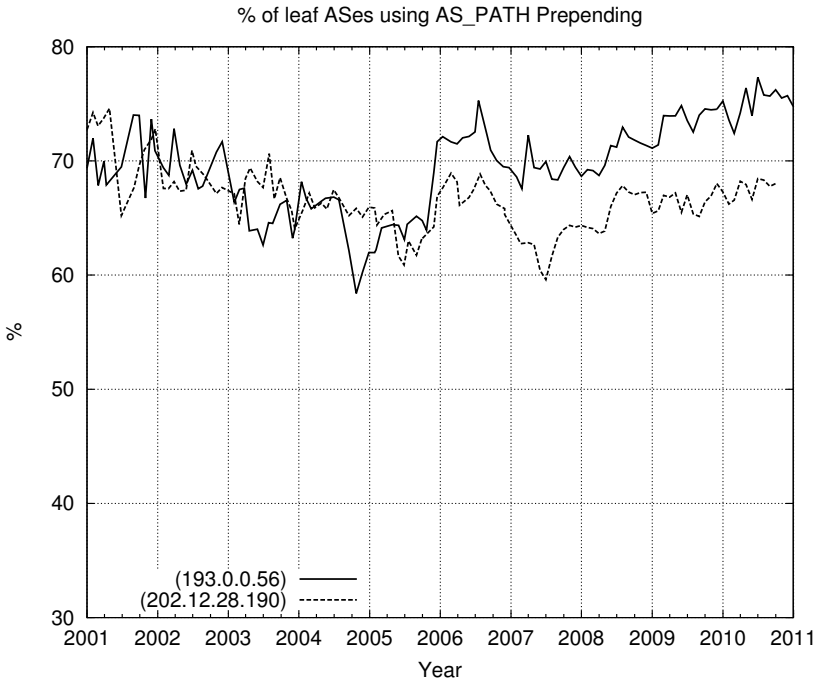


Figure 5.1: Evolution of the use of AS_PATH Prepending by leaf ASes

AS_PATH Prepending is presented in all BGP-4 reference manuals (e.g., [40, 153, 36]). It significantly contributes to the growth of the Internet's routing tables [39]. Figure 5.1 shows that more than 60% of the leaf ASes were using AS_PATH Prepending in the period between 2001 and 2011. These data were extracted from DFZ routing tables contributed to the RIPE RR by two collectors that were active during that period. This figure shows the lower bound for the actual use of AS_PATH Prepending, because the BGP-4 route decision mechanisms selects advertisements with the shortest AS_PATH and might have suppressed advertisements with AS_PATH Prepending.

5.2.1 Side-effects of Traffic Engineering

The Internet is an eco-system of collaborating and yet competing ISPs. When applying the aforementioned techniques in the Internet, conflicts between ISPs will arise. These conflicts trigger advertisements and counter-advertisements as each AS acts in its own interest trying to neutralise the negative impact of a foreign policy change and to optimise the utilisation of its resources [154, 155].

The response of the BGP-4 protocol introduces additional complexity to the problem. As observed by T. Griffin in RFC 4264 [156], the behaviour of BGP-4 is not completely stable and there are interactions between routing policies in interconnected autonomous systems which lead to unpredictable responses, which can not be debugged. They are discovered by a network operator, but he is not able to manipulate the network to make it return to the expected state, neither by changing the policies used by BGP-4 in his AS nor by other more drastic means like disabling and re-enabling an interface Even in the absence of these interactions, Bush and Griffin [157] show that due to the timing mechanisms in BGP-4, a stimulus consisting of a single advertisement in a simple network with several ASes may trigger different BGP-4 update exchange sequences between the ASes until the system settles. In fact, Griffin et al. demonstrate, that the effect of applying routing policies on the configuration of the BGP-4 protocol in a router is non-deterministic and may lead to unpredictable network states [54]. They demonstrate that policies render the BGP-4 protocol metastable. This means that under some circumstances, BGP-4 will oscillate, producing so-called *routing storms*. Routing storms are characterised by abnormally high update traffic and impact the experienced QoS. Therefore, network operators will need to eliminate them. They are good candidates to find BGP-4 update sequences that show how this is done.

Additionally, there is debate regarding route flap dampening in RFC 2439. Route flap dampening has caused major disruption in the Internet [157]. It is generally accepted as being harmful [158] and has been deprecated by RIPE [159] because it can further destabilise the network. Despite this, some BGP-4 speakers in the Internet continue to use it. All these effects have to be taken into account when analysing BGP-4 update traffic stored in the RIPE RR or other repositories.

5.2.2 Preselection Algorithm for BGP-4 Update Sequences

The overall objective of my work is to propose an architecture that decouples routing policies from the global BGP-4 routing infrastructure. By doing so, the main routing infrastructure uses a policy-free and, therefore, stable and deterministic BGP-4 [54]. My study of BGP-4 update sequences was aimed at sequences that can be linked to different mechanisms used to implement Traffic Engineering using BGP-4. Once identified, I could integrate these techniques into the architecture in a way that does not affect the stability of the Internet's BGP-4 routing plane.

I started by examining sequences of BGP-4 updates referring to a specific prefix that arrive at the RIPE RR and always follow the same path. In order to simplify, I use the term Canonical AS_PATH for the sequence of ASes traversed by an update:

Definition 8 *The **Canonical AS_PATH** is the sequence of ASes traversed by an advertisement. It can be derived from the AS_PATH attribute by removing all duplicated ASes from it (see Figure 5.2).*

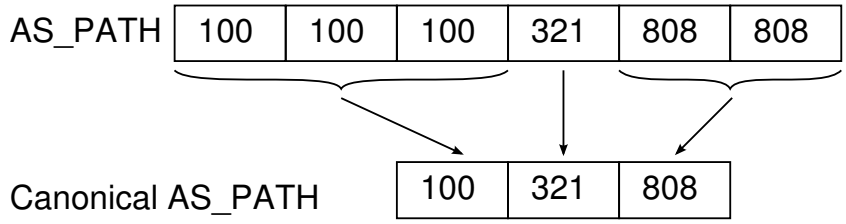
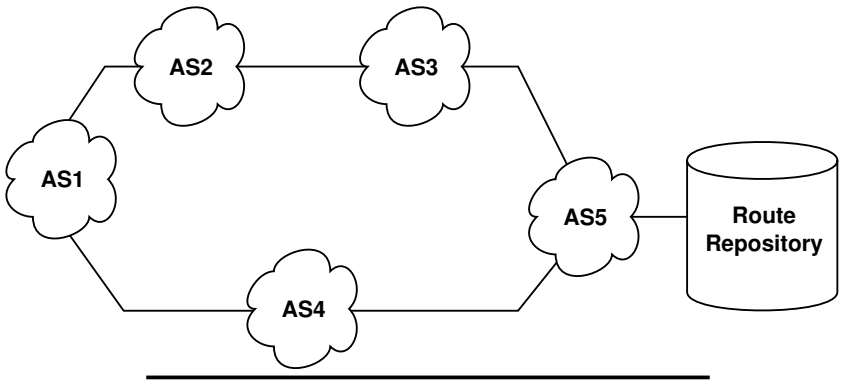


Figure 5.2: Relationship between AS_PATH and Canonical AS_PATH

The sequences of advertisements I studied share the same Canonical AS_PATH but have fluctuating AS_PATH. These sequences show what prepending policies are used by the ASes during the lifespan of the sequence. As Figure 5.3 shows in a simple example, sequences where AS_PATH Prepending cause a change in the Canonical AS_PATH do not yield any information regarding the routing policies.



Policy in link AS1-AS2	Policy in link AS1-AS4	AS_PATH in Route Repository
no prepend	no prepend	AS1-AS4-AS5
no prepend	prepend by 2	AS1-AS2-AS3-AS5

Figure 5.3: Change in AS_PATH Prepending policy that is not reflected in the AS_PATH stored in a route repository

In order to fulfil the requirement of arriving through the same path to the RIPE RR, the sequences of BGP-4 updates I filtered out for study have the following characteristics:

- they refer to the same prefix
- they are contributed by the peer to the RIPE RR,

- they are consecutive in time, and
- they are
  - advertisements with the same Canonical AS_PATH, or
  - withdraws

The advertisements contain information about the different policies applied by the ASes along the path. The withdraws are originated by events like:

- routes that were manually withdrawn
- oscillating routes that were put on hold by route flap dampening as defined in RFC 2439.
- policy changes that were applied to the network and activated by the soft reconfiguration mechanism [61]

All these events are significant when trying to understand the mechanisms involved in network management and their impact on the Internet's routing tables. In order to make the new architecture more likely to be accepted by the network management community, these mechanisms need to be mapped into my proposal.

Figure 5.4 shows the sequence detector I used. It is implemented by a BGP-4 update sequencer shown in Algorithm 1 followed by an evaluator, that narrows down the characteristics of the BGP-4 sequences kept for further analysis.

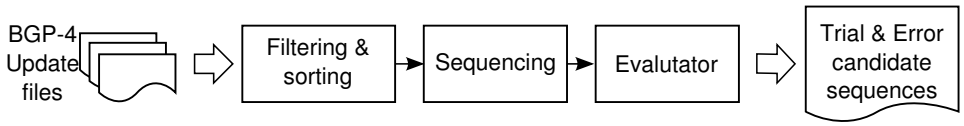


Figure 5.4: Sequence detector: logical blocks

The evaluator I have used for all my published work is shown in Algorithm 2. It flags BGP-4 update sequences that have two or more advertisements with different AS_PATH length. With this evaluator, I get rid of sequences where the AS_PATH does not fluctuate, because no information of the AS_PATH Prepending policies can be obtained from them. Additionally, it includes an inter-arrival time threshold. This threshold helps filtering out different phenomena. For instance, implementations of the Route Refresh Capability [60] are based on sending a withdraw and a new advertisement for the routes that need to be refreshed in the same BGP-4 update or very close in time; when a route oscillates, the interval between routing updates is influenced by the protocol timers; finally, when an operator or a machine applies policies, the time between two consecutive updates tends to be several times the BGP-4 protocol timers. Table 5.1 shows the threshold values used depending on which phenomenon needs to be analysed.

Table 5.1: Threshold ranges used to analyse different phenomena

Threshold interval	Phenomenon under study
0 s to 5 s	Route refresh capability
15 s to 1 min	BGP-4 protocol timers
5 min or more	Network maintenance or network management (e.g., traffic engineering, etc.)

Data: all updates collected by a specific device during a time window [t0,t1]

Sort by prefix, peer and time in this order;
foreach *prefix* **do**
  foreach *peer* **do**
    $canonical\_as\_path \leftarrow \emptyset$;
    $collector \leftarrow \emptyset$;
    foreach *update* **do**
      `/* skip all withdraws at the beginning */`
      if $canonical\_as\_path == \emptyset$ **then**
        if *IsAdvertisement(update)* **then**
          $canonical\_as\_path \leftarrow$ getCano(update);
      if $collector == \emptyset \parallel$
      `/* get all withdraws and updates sharing the canonical path */`
      IsWithdraw(update) $\parallel$
      getCano(update) $== canonical\_as\_path$ **then**
        $collector \leftarrow collector \cup update$;
      else
        `/* when the canonical path changes, evaluate what you collected */`
        evaluate(collector);
        `/* and start all over again */`
        $collector \leftarrow update$;
        $canonical\_as\_path \leftarrow$ getCano(update);

Algorithm 1: The AS_PATH Prepending sequencer

Data: a candidate update sequence
Data: Δt_{min}
Result: true if candidate is Trial and Error Traffic Engineering (Trial and Error TE)

```
fluctuated ← false;
adv_counter ← 0;
last_aspath ← ∅;
last_TS ← 0;
foreach update in candidate do
    if IsAdvertisement(update) then
        adv_counter ← adv_counter + 1;
        if last_aspath ≠ ∅ then
            if last_aspath ≠getASPath(update) then
                fluctuated ← true;
        last_aspath ←getASPath(update);
        if Δt_min > 0 then
            /* Take into account a minimum threshold for the sequence
               interarrival time.  If the Route Refresh Capability is
               activated, a withdraw and an advertisement might arrive
               at the same time.                                      */
            if last_TS ≠ 0 and last_TS ≠TimeStamp(update) then
                if TimeStamp(update)−last_TS < Δt_min then
                    /* Inter-arrival time out of range.              */
                    return false
            last_TS ← TimeStamp(update);
return fluctuated == true and adv_counter > 2
```

Algorithm 2: The basic AS_PATH Prepending evaluator

5.2.3 Traffic Engineering with BGP-4: Defensive AS_PATH Prepending

In order to test the detection algorithm with the simple evaluator, I took the data collected by the RR from peers in the London Internet Exchange (LINX) during May 2007. This selection was motivated because there are publicly available traffic profiles for that particular month [160]. These traffic profiles clearly show a period of high and a period of low data traffic. The period of low traffic is also known as *off-peak hour*. During May 2007, the off-peak hour was during the early morning, i.e., from 04:00 to 08:00 local time. The off-peak hour is optimally suited for network maintenance because the impact on customers is minimised. This time window was my first target when looking for traces of maintenance operations in the RIPE RR.

I divided the data sets in time windows of 4 hours, i.e., the duration of the off-peak

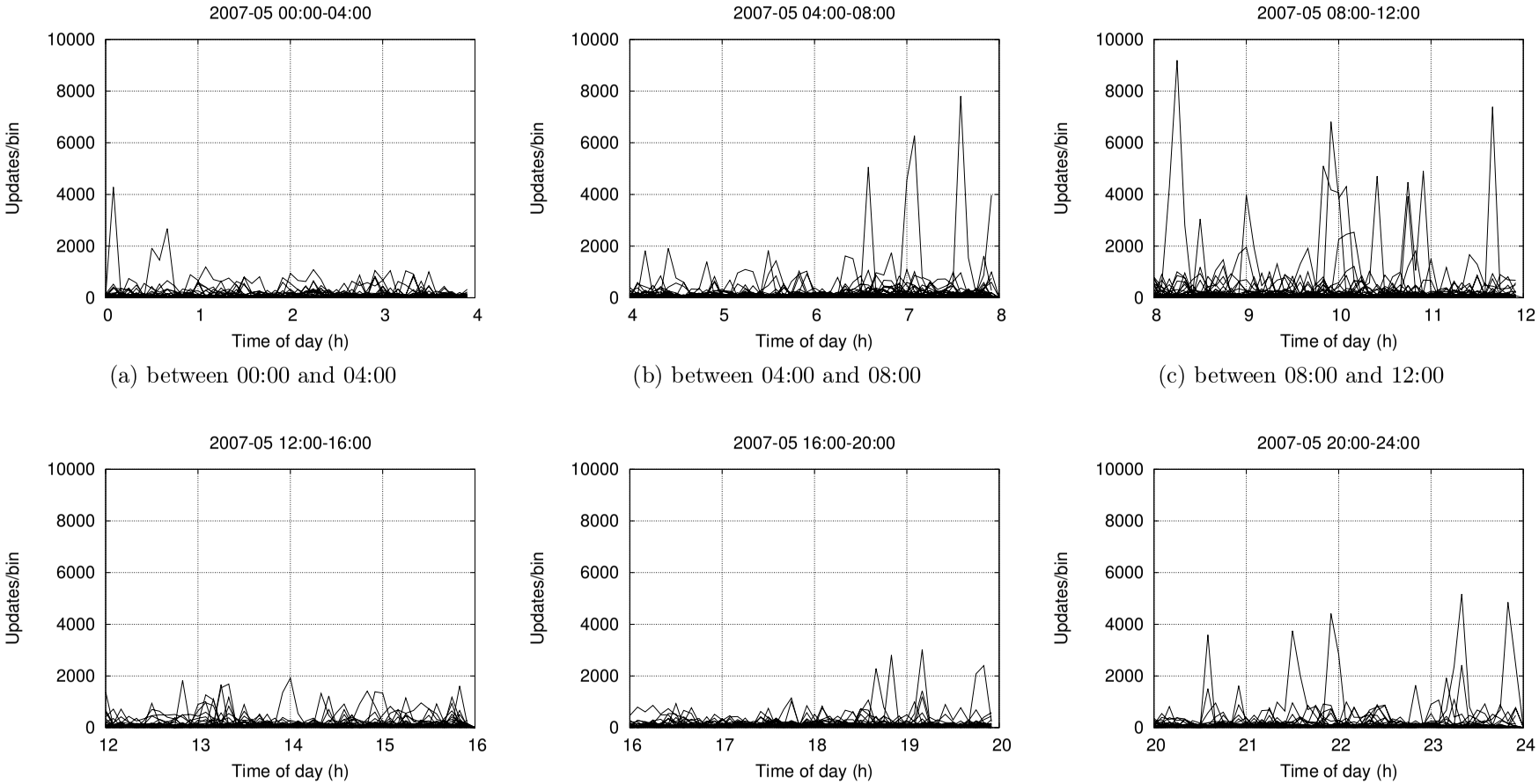
period, starting at 00:00 GMT. The six resulting time windows approximately cover the off-peak period in the different time zones: for South America it is four hours earlier than in Europe, the US east coast it is shifted 6 hours, the west coast is shifted between 8 and 9 hours, and the Asia/Pacific rim, taking Tokyo as a reference, is shifted 8 hours later than the US west coast.

I used a five minute bin to count the number of updates. This bin corresponds to the time window stored in each of the update files in the RRs. It is also used by major vendors to collect long trend statistics in network equipment [153] and allowed me to compare the routing data with traffic data sampled with the same frequency.

Figure 5.5 shows the evolution of the number of BGP-4 updates per 5 minute bin detected by the BGP-4 update selection algorithm. It superimposes the graphs for all days of the month. In the time window between 00:00 GMT and 04:00 GMT, sustained activity with sporadic larger peaks can be observed. This activity corresponds to activity in Europe, because the prefixes involved in it were assigned in that period of time to European ASes.

Significant peaks in the most active bin are detected in the time window between 04:00 GMT and 08:00 GMT, shown in Figure 5.5b. With the exception of these peaks, the activity for this time window can be traced back to operations in South America. The activity in the next time window, shown in Figure 5.5c, can be linked to operations in North America. The next time window shows relatively low activity. The last two time windows show some more activity.

Uhlig et al. measure a background traffic of around 100 updates per minute as EGP sustained traffic in the traces they use for their outbound TE system for stub ASes [70]. It must be noted that this figure represents the total number of updates per minute in one AS. The figures presented here only take into account sequences which the TE detector has selected but contain the BGP-4 traffic from **all** the ASes which contribute to the LINX. Uhlig's measurements are confirmed in Figure 5.6. When taking the number of updates per bin on for a specific point in time, the upper end of the 95% confidence interval for the samples during May 2007 stays below 500 BGP-4 updates per bin, i.e., the 100 updates per minute figure measured by Uhlig. Only Figure 5.6e has larger periods of time where this does not happen. This result shows that applying the BGP-4 update selection algorithm blindly on data stored in a BGP-4 routing repository helps detecting abnormal situations.



(a) between 00:00 and 04:00

(b) between 04:00 and 08:00

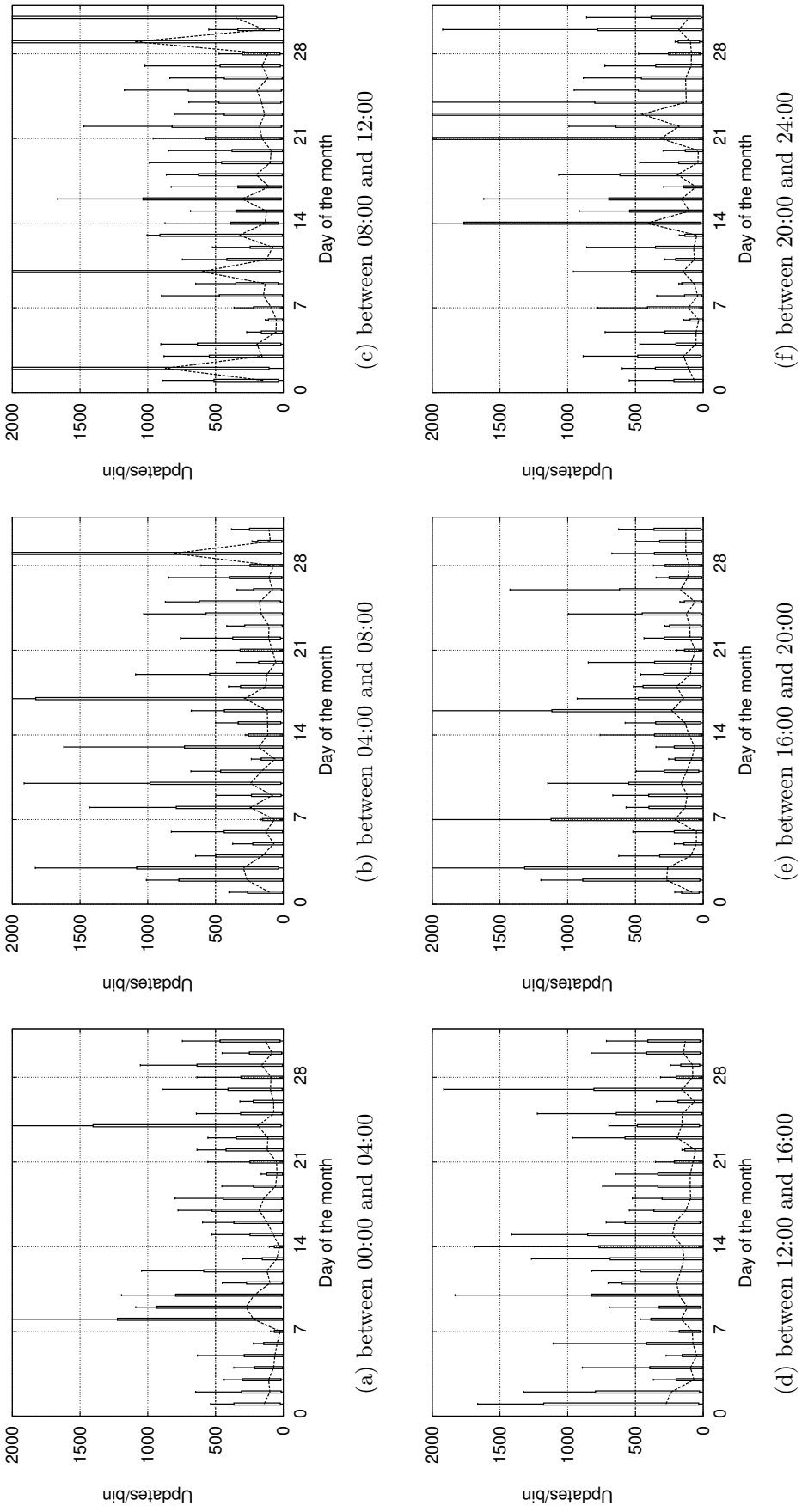
(c) between 08:00 and 12:00

(d) between 12:00 and 16:00

(e) between 16:00 and 20:00

(f) between 20:00 and 24:00

Figure 5.5: Filtered TE profiles for May, 2007



(a) between 00:00 and 04:00 (b) between 04:00 and 08:00 (c) between 08:00 and 12:00

(d) between 12:00 and 16:00 (e) between 16:00 and 20:00 (f) between 20:00 and 24:00

Figure 5.6: Mean and 95% confidence intervals for May 2007

The algorithm proves to be very efficient when analysing a limited set of prefixes. This scenario represents the use case of an Autonomous System checking how the prefixes it advertises are seen in the Internet. The peaks shown in Figures 5.5b and 5.5c can be dated to the 29 of May, 2007, as confirmed in the peaks on the mean BGP-4 traffic in Figures 5.6b and 5.6c. I collected the prefixes involved in the peaks from the update sequences that yielded Figure 5.5 and applied the sequence detection algorithm to advertisements involving this set of prefixes only. The result is shown in Figure 5.7. It shows that these prefixes produced a significant amount of BGP-4 traffic in a delimited time window. The peak traffic is two orders of magnitude above Uhlig's average BGP-4 traffic for 100 updates/min or 500 updates per bin.

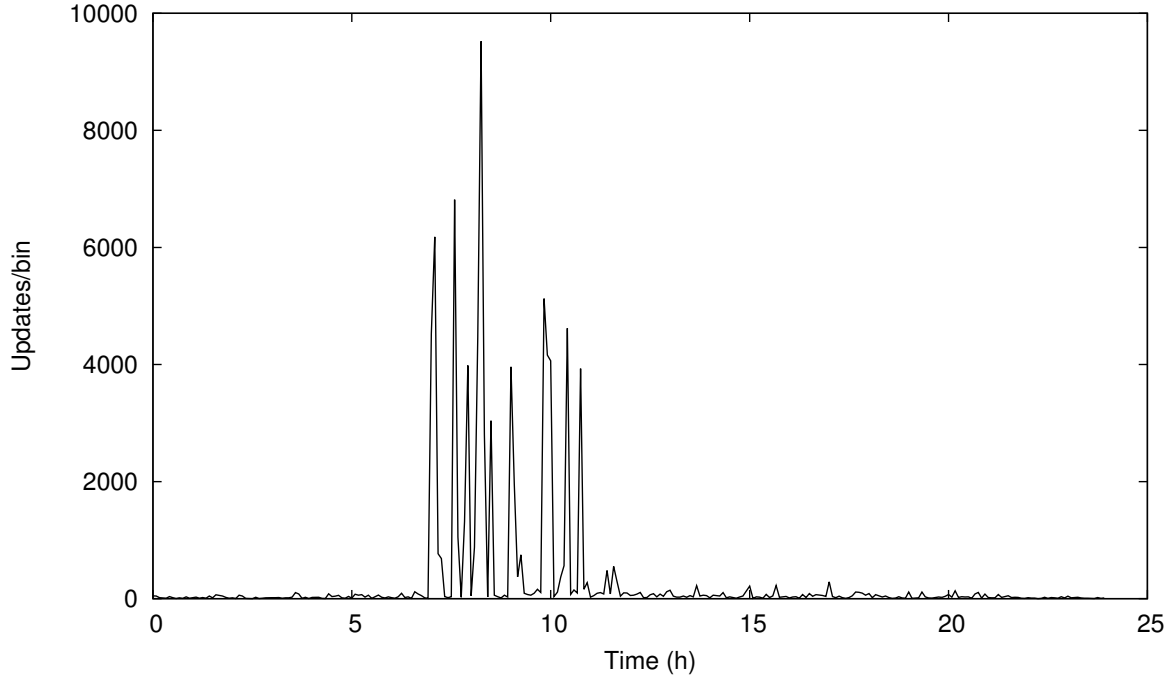


Figure 5.7: Profile for the 29 of May, 2007 for the subset of prefixes involved in the outlier

In order to further evaluate this traffic peak, I studied the traffic contributed by peer 202.12.28.190 to traffic collector RRC00 of the RIPE RR the day after, i.e., the 30 of May, 2007. Figure 5.8 shows the frequencies of the count of updates per 5 minute bin. The distribution is skewed with regards to a normal distribution. Different alternative distributions were tested with GNU-R [161]. Listing 5.1 shows that the hypothesis of a logarithmic normal distribution for the BGP-4 traffic cannot be rejected ($p > 0.05$).

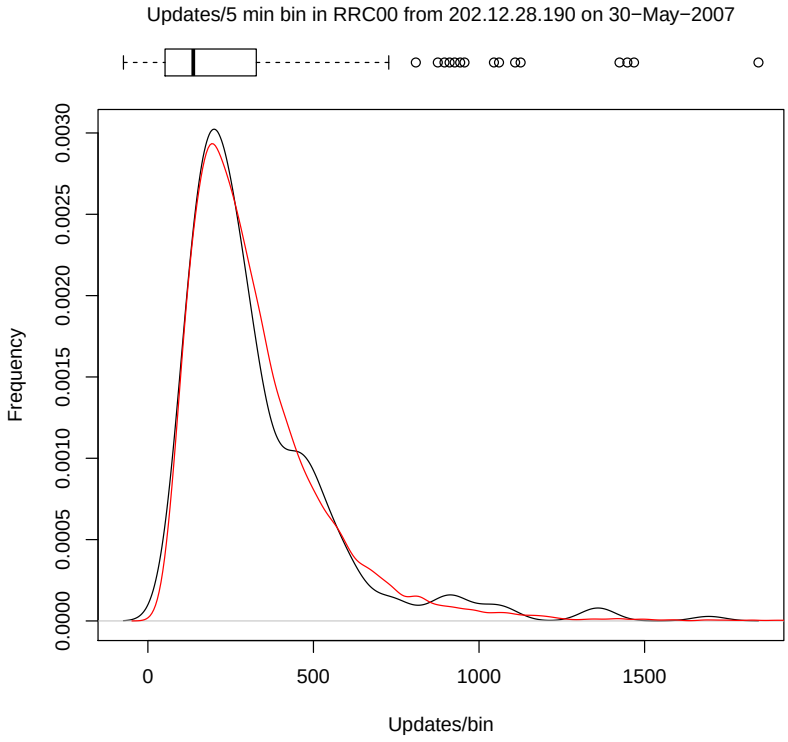


Figure 5.8: Frequency distribution of the BGP-4 traffic

Listing 5.1: GNU-R fit to a log-normal of the BGP-4 traffic

```
> summary(total)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   76.0   182.5   254.0   328.6   413.2  1694.0
> summary(log(total))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.331   5.207   5.537   5.607   6.024   7.435
> fitdistr(total,"log-normal")
    meanlog        sdlog
  5.60652164   0.59046248
 (0.03479334) (0.02460260)
> ml<-mean(log(total))
> sdl<-sd(log(total))
> ks.test(total,"plnorm",meanlog=ml,sdlog=sdl)

        One-sample Kolmogorov-Smirnov test
```

```
data:  total
D = 0.0562, p-value = 0.3225
alternative hypothesis: two-sided
```

Additionally, I also studied the empirical Cumulative Distribution Function (eCDF) of the number of updates per bin contributed by 202.12.28.190 during May of 2007. This function is shown in Figure 5.9. It shows that the probability of the number of updates in a bin being 1000 or more is very small. Table 5.2 shows that the probability that more than 3000 updates were collected during a time bin is very low. Therefore, the peaks in Figure 5.11 can be considered outliers.

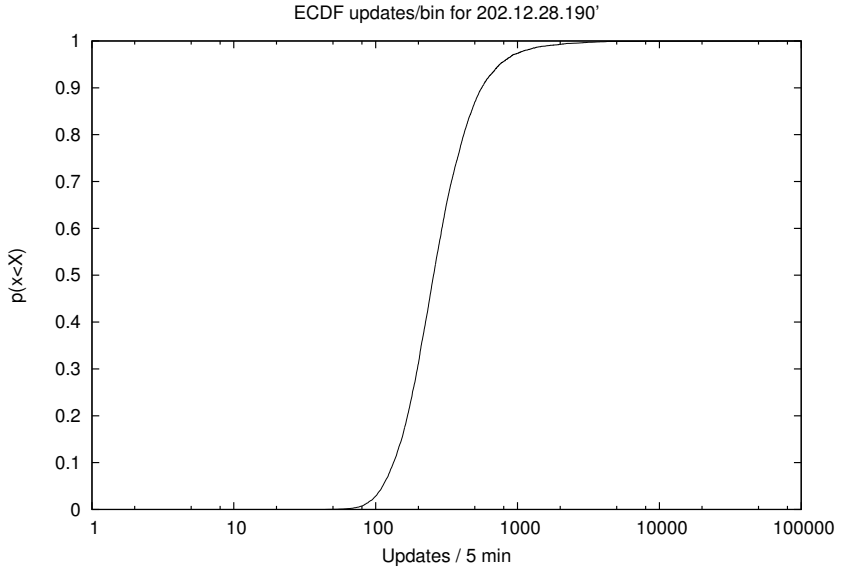


Figure 5.9: ECDF of the number of updates per bin

Table 5.2: Probability of collecting a large number of updates in a time bin

N (Updates per bin)	$p(n < N)$
3000	0.003360
4000	0.001680
5000	0.001008
6000	0.000896
7000	0.000784
8000	0.000672
9000	0.000560
10000	0.000448

Figure 5.11 shows a 3D expansion of Figure 5.7. I was obtained by sorting the advertisements by the length of their AS_PATH attribute. One axis represents the time of the day, another the length of the AS_PATH attribute, and the third the number of updates per bin. The graphs are drawn for a fixed AS_PATH length. The graph for AS_PATH length 0 shows the number of withdraws and the graph for AS_PATH length 18 shows the total number of updates collected for the set of prefixes under study.

Figure 5.10 shows the same graph projected on the *xy* plane. This gives a better view of the periods of time where advertisements with a given AS_PATH length could be observed. The horizontal lines in the graph delimit the period of time with unusually high activity.

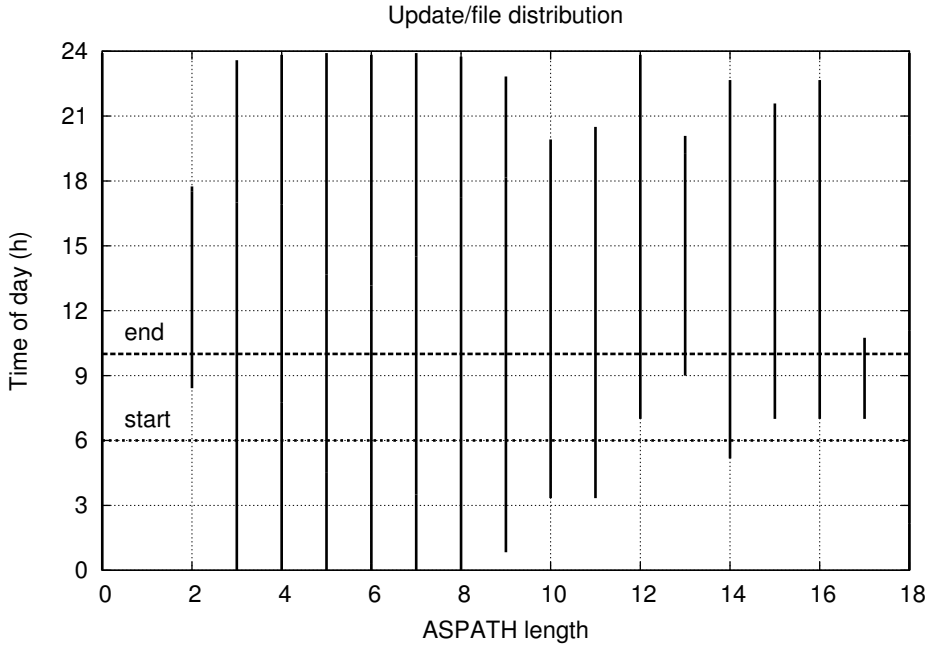


Figure 5.10: Different AS_PATH lengths in time

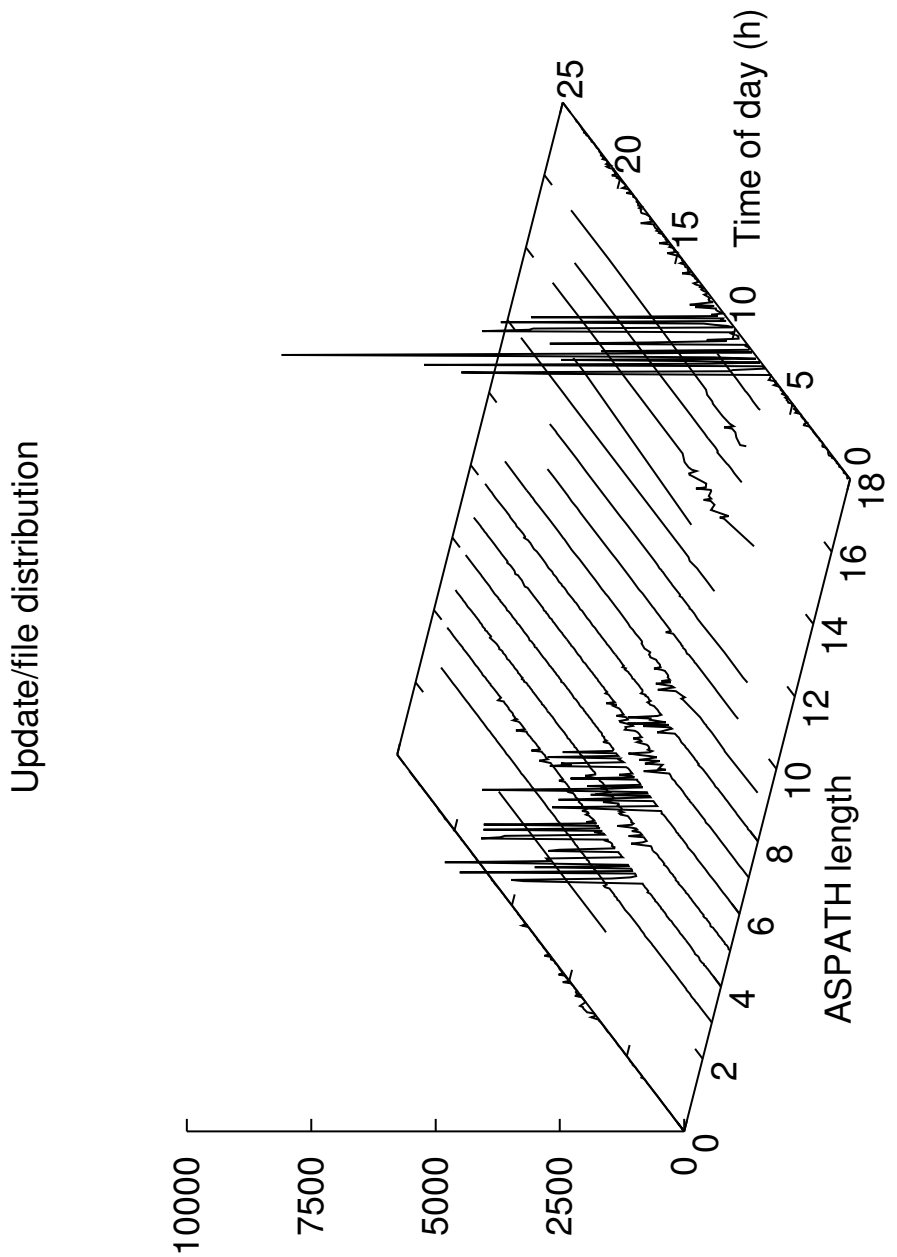


Figure 5.11: Decomposition of the profile in Figure 5.7 by AS_PATH lengths

From both figures we see that

1. the event produces the highest number of updates for AS_PATHs attributes with lengths 4 and 6. The graphs follow the profile of the aggregated traffic. Figure 5.12 shows that they account for approximately 70% of the events.
2. AS_PATH lengths shorter or equal to 9 and with lengths 10 and 11 are present during the whole observation period or appear very much in advance with respect to the period of maximum activity. Except for the above mentioned lengths 4 and 6, the traffic is not very significant.
3. AS_PATH lengths 12, 15, 16 and 17 appear after the activity ramps up. As shown further down, they are used to assign unstable paths a very low priority. They show an attempt to divert the incoming traffic to paths that are more stable, in the hope that the AS_PATH attribute with which they are advertised is shorter than the AS_PATH of the unstable path.

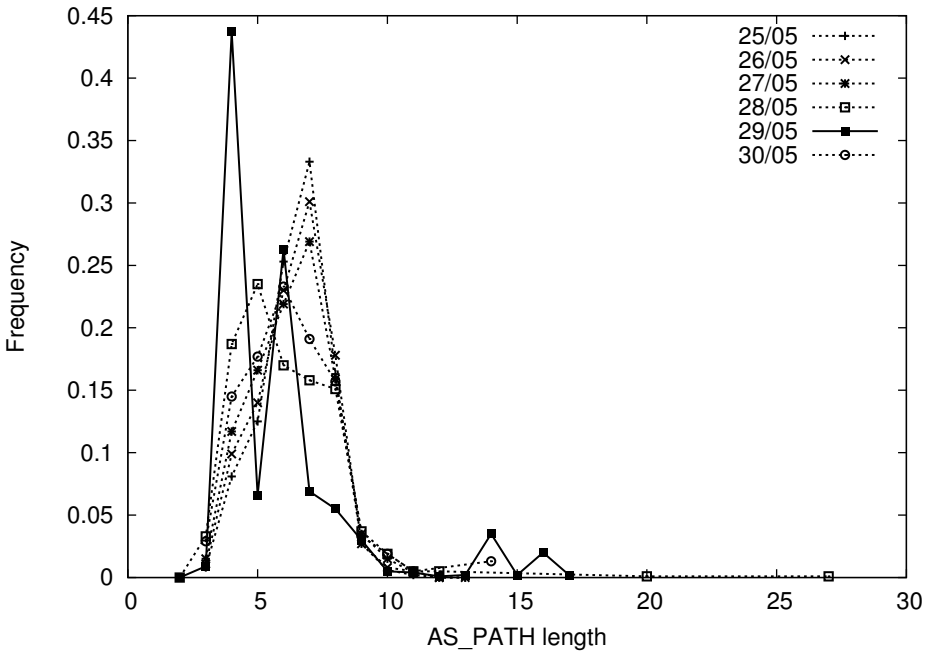


Figure 5.12: AS_PATH lengths frequency distribution, during and after the storm

Figure 5.12 and Figure 5.13 show the frequencies of the lengths of the AS_PATH attribute and the Canonical AS_PATH, respectively. Figure 5.13 shows that ASes which are 4 hops away (i.e., Canonical AS_PATH length is 4) account for more than 80% of the events, whereas the rest of the days the activity is mainly distributed among ASes that are between 3 and 5 hops away. The advertisements need to be further analysed to detect the source of the instability.

Figure 5.12 shows that the AS_PATH length fluctuates between 2 and 8. The day of the incident, the distribution has two main peaks at when AS_PATH length is 4 or 6 and two additional peaks for when the AS_PATH length is 14 and 16. The first two peaks together show that most instabilities result from ASes in the path which

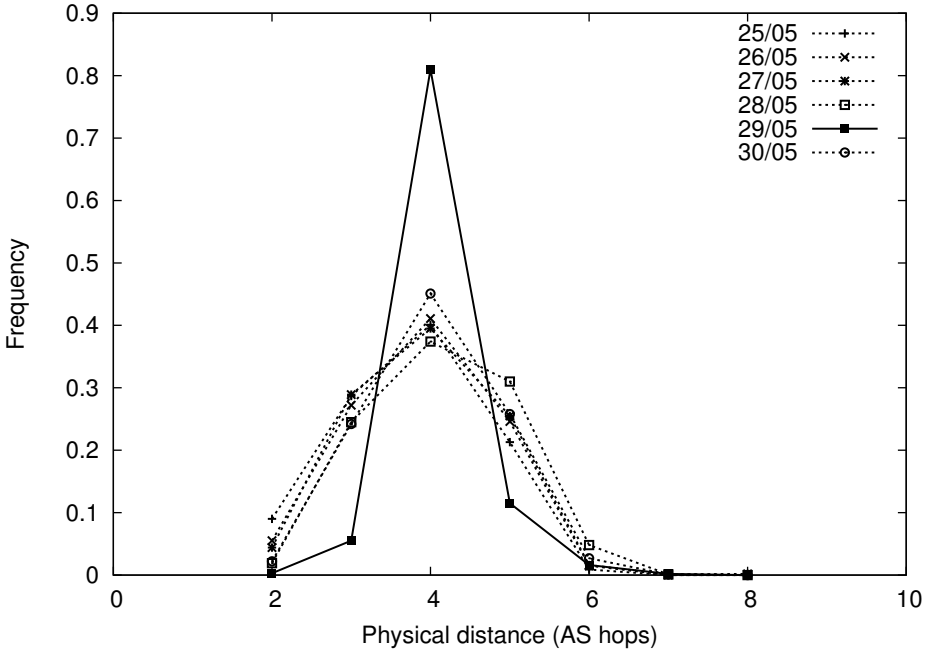


Figure 5.13: Canonical AS_PATH probability before, during and after the storm

toggle between announcing without AS_PATH Prepending and prepending by 2. This behaviour is also reflected in the beginning of the sequences shown in Listing 5.2. The other two peaks result from AS14 prepending by 10, in order to reduce the preference of the path.

Defensive AS_PATH Prepending can be explained as follows: ASes prefer stable paths because the QoS parameters stay within predictable limits [162]. Stable paths also reduce the risk of processor overload in the routers as a consequence of continuous path computations. Therefore, AS administrators fine-tune the routing policies they deploy, making unstable paths less preferable than stable paths. This is achieved by making unstable paths artificially longer than stable paths using AS_PATH Prepending. The amount of prepending introduced on the advertisements has to guarantee that the unstable path will be always appear longer than any stable alternative.

Listing 5.2: Defensive AS_PATH Prepending: anonymized advertisements for a specific prefix received in the RIPE RR

```
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 51 51 14|...
```

```
...|A|...|7 2 51 14 14 14 14 14 14 14 14 14 14 14|...
...|A|...|7 2 51 51 51 14 14 14 14 14 14 14 14 14 14 14|...
...|A|...|7 2 51 14 14 14 14 14 14 14 14 14 14 14|...
...|A|...|7 2 51 51 51 14 14 14 14 14 14 14 14 14 14 14|...
...
```

Listing 5.2 shows an excerpt of the output of the analysis program. All records correspond to the same prefix. The '|A|' field indicates that the BGP-4 updates were advertisements. Autonomous System Numbers in the AS_PATH attribute have been changed in order to provide some level of anonymity. It shows an update group where AS51 is oscillating and an update group that exhibits a defensive AS_PATH Prepending pattern. The source of instability is AS51. It is toggling between advertisements without AS_PATH Prepending and advertisements prepended by 2. The end of the sequence corresponds to AS14 entering defensive mode: it starts to prepend by 10, in order to make this path less preferable compared to other, more stable paths. This behaviour explains the peaks for AS_PATH lengths 14 and 16 in Figure 5.12 the day of the incident. The day after, a fraction of the updates still have an AS_PATH length of 14. They correspond to ISPs which prefer to maintain the defensive re-routing until there is a reasonable level of certainty that the path is stable again.

Table 5.3: AS_PATH Prepending behaviour (excerpt 1 of 2)

AS	fake hops	before	during	after
14	10	2	258	30
16	4		1120	
16	5		1043	
51	2	6	2818	37

Table 5.4: AS_PATH Prepending behaviour (excerpt 2 of 2)

AS	fake hops	before	during	after
11	2			3
11	3	98	108	95
11	4	20	15	3
60	1		2	2
60	2	12	22	13
60	3	115	156	130
60	4	127	178	142
67	3	91	67	60
67	6	60	32	37

Finally, the AS_PATH Prepending behaviour of all ASes was studied. Tables 5.3 and 5.4 show excerpts of the full result. Table 5.3 shows Autonomous Sys-

tems that behave depending on whether the BGP-4 update traffic is low or high. Only one Autonomous System (AS14) is applying defensive AS_PATH Prepending during and after the incident by introducing 10 fake hops into the AS_PATH. Two additional Autonomous Systems are involved in significant activity: AS51 introduces 2 fake hops in a significant amount of events and AS16 toggles between introducing 4 and 5 fake hops.

Table 5.4 shows ASes that use the same policies when the BGP-4 update traffic is low and when it is high. AS11 mainly prepends by 3 and 4, AS60 has policies which prepend by 2, 3 and 4 and AS67 consistently prepends by 3 and 6. The fact that AS11 starts prepend by 2 and that AS60 starts prepending by one can be considered a result of the increase in BGP-4 traffic.

5.3 Studying Provider Behaviour

As explained in Section 3.1.2, ISPs sign extensive interconnection contracts, which include SLAs. In some cases, these SLAs define acceptable service time windows, where invasive network maintenance can be performed. These time windows normally occur when the impact on the end-user Quality of Experience (QoE) of such maintenance operations is minimal. Network maintenance involves either changes in the physical network or changes in its routing configuration. Both cases trigger BGP-4 update traffic while BGP-4 copes with them. Since BGP-4 tries to optimise traffic by grouping prefixes with the same attributes into updates, studying BGP-4 update traffic may be inconclusive for purposes like SLA dispute. The tools I have developed to understand how ISPs operate their networks are finer grained and use the prefix information contained in the BGP-4 updates.

5.3.1 Update arrival time distribution

In order to understand the behaviour of a prefix over time, historical BGP-4 data are needed. These data can come from public routing repositories, like RIPE's Routing Repository [3] or the Oregon Routeviews project [76] or from one or more private routing repositories installed by Internet Service Provider in their infrastructures.

The method I propose to analyse the prefix activity and relate it to the operation policy of a provider is to

1. select the update traffic files for the period under study
2. select the updates related to the prefix under study, and
3. to compute and plot the empirical Cumulative Distribution Function of the prefix update arrival time-of-day, and
4. compare with information from other sources like the Internet Routing Registries

I then complement this analysis with information about the geographical location and the nature of the AS provider coming from other sources like the Internet Routing Registries, to put the results in perspective.

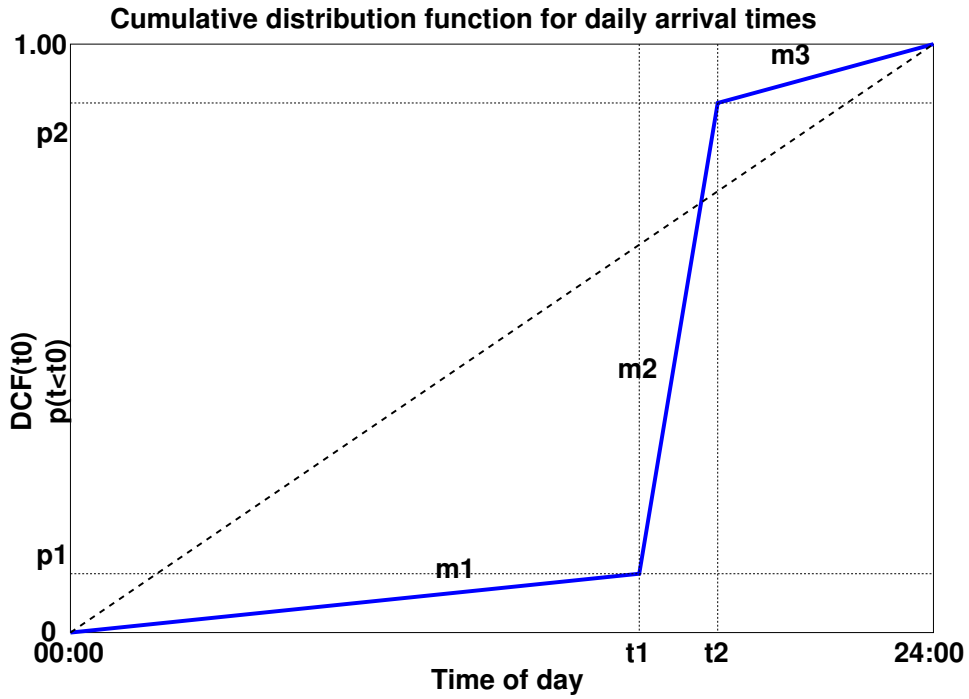


Figure 5.14: Idealised update arrival time CDF

Figure 5.14 shows a simple idealised Cumulative Distribution Function (CDF) for update arrival times with three specific time windows. In all three time windows, the distribution of arrival times is uniform. The slopes m_1, m_2 and m_3 denote different levels of activity and are proportional to the arrival frequency in the specific time interval. In the case of Figure 5.14, the interval of maximum activity is $t \in \{t_1, t_2\}$. In order to minimise the impact of network operations on their customers, some network operators define operation windows, where manipulations on the network are allowed [163, 164].

There are two extreme cases, from a mere Operations & Management point of view:

- No operations time window: the prefix is updated randomly at any time of the day.
- Well defined operations time window: the prefix is only updated during one or more well-defined time intervals; in the latter case, the number of intervals is usually small.

Even if the SLA signed with a client does not include maintenance time windows, the operator may choose to restrict operations to specific time windows in order to protect network integrity, the SLAs signed with other clients, etc.

5.3.1.1 No predefined operations window

When an AS has no predefined operations time window, updates will be distributed uniformly during the whole day. The empirical Cumulative Distribution Function will be close to the diagonal line starting at probability 0 at time 0:00 and ending at proba-

bility 1.00 at time 24:00. This reference line is shown in Figure 5.14 and in all graphics in this paper. It gives a visual reference for the deviation from this uniformly random behaviour.

5.3.1.2 Well-defined operations window

When all updates occur in a well-defined operations window $t \in \{t_1, t_2\}$, the curve will have two horizontal segments for $t < t_1$ and $t > t_2$. Supposing, again, that the events will be uniformly distributed during the operations window, the central segment will be a straight line between $(t_1, 0.00)$ and $(t_2, 1.00)$. Figure 5.14 shows a more realistic situation, where a small fraction of the events arrive before and after the operations window, while the majority arrives in the operations window (t_1, t_2).

5.3.2 Analysis of AS behaviour

Figure 5.15 shows the behaviour for different prefixes as recorded by RIPE RR at the London Internet Exchange (LINX) during 2007. In order to provide a better insight, the BGP-4 updates were classified into four different categories:

1. Prefix withdraws
2. Prefix advertisement without prepending
3. Prefix advertisement with prepending by the originating AS
4. Prefix advertisement with prepending by other ASes in the AS_PATH

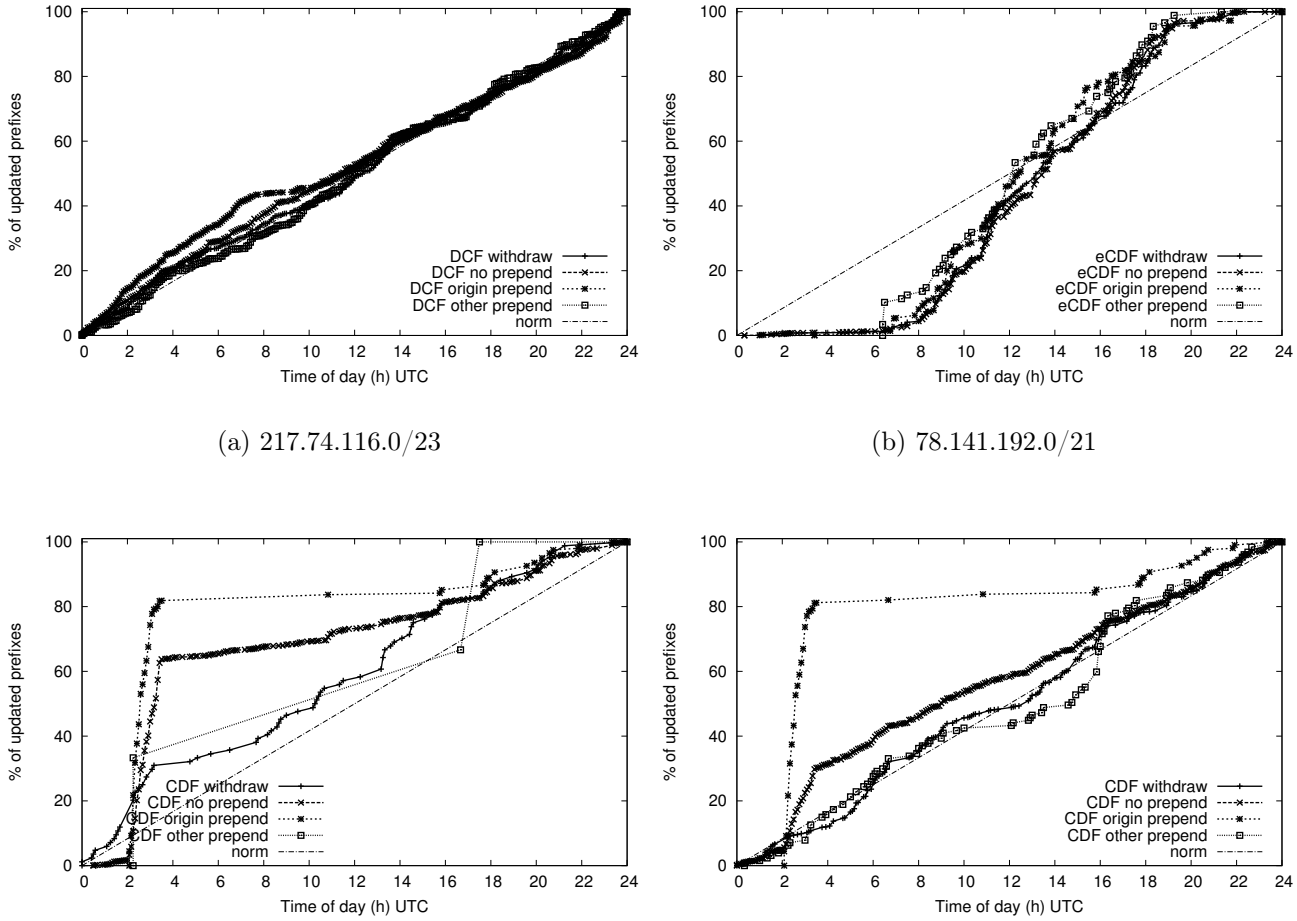
The relation between four categories determines the impact of local and foreign policies on the path between the originating AS and the routing repository that is used as reference point.

5.3.2.1 Uniform distribution

Figure 5.15a shows a prefix that exhibits a nearly uniform distribution for all four categories. Using GNU-R [161] to analyse the arrival time series, Pearson's Chi-squared test was applied to the daily arrival times of the four categories. Each time series was compared with a uniformly distributed distribution for $t \in \{0s, 86400s\}$ with the same number of samples. As shown in Table 5.5, all p-values are greater than 0.05, which means that the approximation by a uniform distribution is plausible.

Distribution	p-value
Without AS_PATH Prepending	0.2689
AS_PATH Prepending by originating AS	0.2724
AS_PATH Prepending by others	0.2799
Prefix withdraws	0.2851

Table 5.5: Statistical analysis for Figure 5.15a



(a) 217.74.116.0/23

(b) 78.141.192.0/21

(c) 12.160.186.0/24 between March and May 2007

(d) 12.160.186.0/24 during 2007

Figure 5.15: Behaviour of different prefixes during 2007

The fact that all empirical Cumulative Distribution Functions are approximately uniform points towards a prefix which is not subject to any form of maintenance. All advertisements and withdraws occur at random as the result of protocol interactions.

5.3.2.2 Wide operations window

Figure 5.15b shows the behaviour of a prefix which has an operations window of 14 hours, approximately between 06:00 UTC and 20:00 UTC. These values were obtained with the regression utilities of GNU-R. Additionally, the different update categories were matched against comparable uniform distributions between the same times. Table 5.6 confirms that the hypothesis of a uniform distribution cannot be rejected because the p-values are significantly high ($p > 0.05$). In this case, the main contributors to AS activity are Advertisements without AS_PATH Prepending, Advertisements where the originating

AS is prepending and Withdraws. There is only a small amount of Advertisements with AS_PATH Prepending by ASes other than the originator and therefore, these were excluded. This analysis confirms that the activity recorded by the RIPE RR is originated in the AS that had been assigned the prefix during the period of time under study.

Distribution	p-value
Without AS_PATH Prepending	0.2842
AS_PATH Prepending by originating AS	0.2796
Prefix withdraws	0.2579

Table 5.6: Statistical analysis for Figure 5.15b

Listing 5.3: whois information for 78.141.192.0/21

```
route:    78.141.192.0/21
descr:    FON Wireless Autonomous System
          Network
origin:   AS43636
mnt-by:   FON-NET-MNT

address: FON Wireless LTD
address: Avda. Bruselas 7.
address: 28108 Alcobendas
address: Madrid. Spain
```

Listing 5.3 shows the whois [165] information about the prefix under study. Its profile, shown in Figure 5.15b, is compatible with the following facts about this AS:

1. The AS is located in Spain, which uses the CET/CEST timezone (UTC+1h in winter and UTC+2h in summer). The time window shown in Figure 5.15b covers office hours in Madrid (compare with `http://www.ripe.net/projects/ris/rawdata.html` [80] for average office hours in Amsterdam, which lies in the same timezone).
2. The clients of this AS are residential customers, which will make most **interactive** use of its connectivity in the time window between 20:00 UTC and 06:00 UTC. Outside this period of time, the impact of the QoS on the Quality of Experience is not as significant, since the users are not using their Internet connection interactively i.e., Web surfing, but rather for background tasks which they pay less attention to like p2p file exchange.
3. The network operators tend to operate the network during their work hours and do not need to program operations during night hours like in the case of ISPs which have a significant number of business customers.

The fact that nearly all updates are originated by policies deployed by the originating AS strengthens the hypothesis that the BGP-4 traffic is linked with operations of the originating AS.

5.3.2.3 Narrow operations window

Using the generalised method to detect BGP-4 sequences that can be traced back to network operation presented in the previous section, I applied the "Soft Session Restart" detection and "Large Update Inter-arrival Times" heuristics for further analysis. Both help discriminating between sequences that are triggered by external, *intelligent* entities and mere protocol interaction. They take advantage of the fact that a deployment-observation-refinement cycle for TE purposes is affected by the convergence time of the routing protocol.

The heuristic to select sequences with large inter-arrival times ($\Delta_t > 4\,\text{min}$) returned several positives from prefix 12.160.186.0/24 during 2007. Browsing the RIPE RR for updates originated by 12.160.186.0/24 during that year reveals that the prefix only used AS_PATH Prepending between March and May of that year. Figure 5.15c shows its behaviour during that period. It exhibits a well defined and narrow window between 02:00 UTC and 04:00 UTC for AS_PATH Prepending by the originating AS, where more than 80% of the updates appear. This behaviour is confirmed by the empirical Cumulative Distribution Function for the behaviour of this prefix during whole year shown in Figure 5.15d. Examining Figure 5.15d, prefixes without AS_PATH Prepending show a small window of activity at the same times. This window confirms that between March and May 2007, the activity of the AS mainly consisted in sequences interlacing advertisements without AS_PATH Prepending and advertisements with AS_PATH Prepending by the originating AS. During the rest of the year, the advertisements without AS_PATH Prepending are randomly distributed. The curve for the updates were other ASes are prepending is inconclusive in both cases because of the small number of events.

Listing 5.4: WHOIS information for AS21703

```
> whois 12.160.186.0

AT&T WorldNet Services ATT (NET-12-0-0-0-1)
     12.0.0.0 - 12.255.255.255
NETWORKS AND MORE NETWORKS75-186
  (NET-12-160-186-0-1)
     12.160.186.0 - 12.160.186.255

> whois AS21703

OrgName:    Networks \& More! Inc.
OrgID:      NETWOR-390
Address:    24 Highland Bend
City:       Island Heights
StateProv:  NJ
PostalCode: 08732
Country:    US
```

Listing 5.4 shows the information provided by the whois service for the prefix and AS. It confirms that the prefix is allocated to "Networks & More Inc.". This provider is located on the Atlantic coast of the United States of America. Translating CET to EST, the time window for the AS_PATH Prepending activity shown in Figure 5.15c translates to a time window between 10:00pm and 12:00pm. The main activity of this provider, as stated in its Web page [166], is to provide enhanced connectivity to schools, public

libraries and similar institutions. Therefore, the observed time window is compatible with network maintenance operations, since the potential impact on QoS would not be perceived by customers.

5.4 Conclusion

In this chapter, I have shown that it is possible to trace operative procedures on the Internet. I have used publicly available data to show how the routing tables in the Default Free Zone are inflated by additional routing entries that are used either for Traffic Engineering purposes or as self-defence against unstable routing information.

In the next chapter, I examine two major routing storms that happened in 2009. These routing storms are rooted in the definition of the BGP-4 protocol and how it treats malformed BGP-4 updates. I use the results from both chapters to propose an alternate behaviour for BGP-4 in my alternate Internet routing architecture.

Chapter 6

Sources of Instability in BGP-4

Contents

The standardisation process followed by the IETF allows an evolution model that accommodates quickly to changes. This is achieved by introducing incremental changes or enhancements on protocol definitions and maintaining backward compatibility. One example is the Multiprotocol Extension for BGP-4 specification [167], which enables the introduction of new services like MPLS-VPN [104] and the migration from IPv4 to IPv6 [168]. Introducing IPv6 has been necessary, because the address space in IPv4 is near exhaustion: the success of the Internet has been such, that a 32-bit host identifier not enough to identify all devices that connect to it. To cope with this problem, a new version IP with 128-bit long address fields is being deployed.

Something similar is happening at the Internet Service Provider (ISP) level. The original BGP-4 definition in RFC 1654 foresaw unsigned 16-bit Autonomous System Numbers (ASNs) to identify the different ISPs in the Internet. Foreseeing a depletion to the ASN space, the Internet Assigned Numbers Authority (IANA) extended the ASN field to 32 bits in November 2006 and, currently, 32-bit ASNs are being allocated to new ISPs. There are some residual allocations of 16-bit ASNs and the final exhaustion date of the 16-bit ASN address space has been predicted for 2013, as shown in G. Houston's ASN analysis page `http://www.potaroo.net` [17]. The way these 32-bit ASNs are to be handled by BGP-4 is specified in RFC 4893 [18]. Since the number all network elements in the Internet are affected by this change, RFC 4893 opts for the coexistence of 2-byte and 4-byte ASNs and proposes the use of new BGP-4 attributes to carry 4-byte ASNs, that are ignored by legacy equipment. This ensures interworking between new

equipment that needs to be deployed in ASes that are assigned 4-byte ASNs and legacy equipment that only understands 2-byte ASNs. One of the attributes that is impacted by the new 4-byte ASNs is the AS_PATH attribute. In order to propagate AS_PATHs with 4-byte ASNs through Autonomous Systems that are not prepared for 4-byte ASNs, the AS4_PATH was introduced.

This chapter analyses two routing storms that can be traced back to software problems with coping with the AS4_PATH attribute. It is based on work I have published in the The Sixth International Conference on Networking and Services [10].

6.1 Routing incidents linked to 4-byte ASNs

Autonomous System Confederations were originally defined in RFC 1965 [169][1]. They are used to provide better scalability to BGP-4 and a smooth transition period during ISP mergers. An AS Confederation hides the topology of a region composed of ASes in the Internet and makes it appear as a single AS. The part of the AS_PATH that describes the path followed by an advertisement within a confederation is coded in specific AS_PATH segments known as AS_CONFED_SEQ and AS_CONFED_SET. These segments must be substituted with the ASN for the AS Confederation when the advertisement leaves it. While this is clearly defined for the case of the AS_PATH attribute, the situation in the case of the AS4_PATH is ambiguous.

RFC 4893 prohibits the AS_CONFED_SEQ and AS_CONFED_SET segments in the AS4_PATH. The first routing storm of 2009 was caused by a router that started breaking this rule, as described by Watanabe [6] in a talk at the IRS in 2009. This routing storm was caused by a BGP-4 speaker that started leaking AS_CONFED_SEQ segments in some announcements. Its peer implemented RFC 4893 literally and marked those advertisements as illegal. As a consequence, and following RFC 4271, it restarted the BGP-4 session. When the session was reestablished, the advertising peer sent the "wrong" AS_CONFED_SEQ segments again and provoked an oscillation. Apparently, as Watanabe describes in his presentation, the illegal advertisements were leaked to other BGP-4 sessions. An avalanche effect reinforced the routing storm. In order to avoid this problem, equipment manufacturers started to silently suppress the AS_CONFED_SEQ or AS_CONFED_SET segments in AS4_PATH attributes. But this has not solved the problem completely. However, this way of solving the problem has had a side-effect. Some AS Confederations that are in the process of transitioning to 4-byte ASNs may leak routing information that partially reveals their internal topology as I showed in [10].

Softening the rules on how to treat AS Confederation segments has not avoided further routing storms linked with the AS4_PATH attribute. In August 2009, a second major routing storm occurred. The best root-cause analysis for this storm is presented by the Renesys routing blog [5]. It confirms the storm was caused by a malformed AS4_PATH attribute. This blog entry identifies CNCI (AS9354) as the AS that started the incident, confirms that the cause was the presence of BGP-4 updates that contained

[1] The currently active RFC is RFC 5065 [170]

an empty AS4_PATH attribute, and presents a graph of the BGP-4 update traffic variation induced by the storm.

6.1.1 RFC4893 violations revealing internal AS Confederation structure

Analysing data related with this incident, I discovered that the 17th[a] nd the 18th of August, 2009 the collector associated with AS13237 in the LINX of the RIPE RR contributed BGP-4 updates violating RFC 4893. Further analysis of this data revealed that this AS is an Autonomous System Confederation and exposed its internal configuration partially. Figure 6.1 shows the evolution in time of count of updates violating RFC 4893 and its linear regression fit calculated with the fitting function of GNU-Plot (see Table 6.1). Since the asymptotic standard error for the slope is very small, the linear approximation is plausible. The slope is expressed in RFC 4893 violations per hour, which translates into a frequency of

$$f = \frac{4.62}{3600\text{s}} \approx \frac{1}{780\text{s}}$$

or a bogus update injected by the peer every 780 seconds during the period of observation. These bogus updates may lead to routing information loss or BGP-4 session tear-downs [6].

Table 6.1: Linear fit provided by GNU-Plot

linear fit:	y = m*x + b	
m	4.615 +/- 0.033	0.71%
b	-3.614 +/- 0.924	25.56%

By inspecting the AS4_PATH attribute of the bogus updates, topological information from inside the AS Confederation could be obtained. Listing 6.1 shows an excerpt of the output of my analysis software. It clearly shows that AS13237 was not replacing its AS Confederation information when sending the information to the collector. Additionally, this listing shows that it has chosen to use private Autonomous System Numbers in the confederation. Figure 6.2 shows the partial routing tree inside AS13237 revealed by the bogus AS4_PATH attributes during the observation period. It includes the ASes in the confederation and the first hop behind them. It shows the internal structure of AS13237 with some level of detail, like the multi-homing relationships with AS1299, AS16150, AS9002 and AS174. This level of detail is normally hidden by the AS Confederation and only appears in the incorrect AS4_PATH attribute. This level of detail can neither be reached when RFC 4893 is implemented correctly and the AS_PATH attribute is the only source of information.

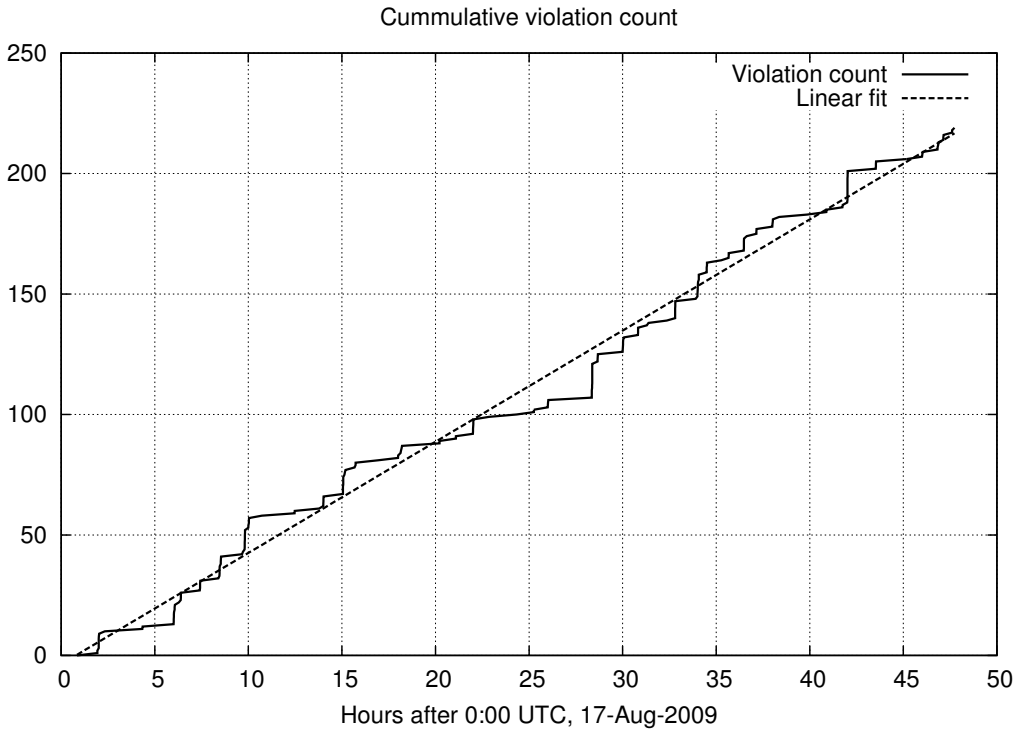


Figure 6.1: RFC 4893 violations during the 17th and 18th of August, 2009

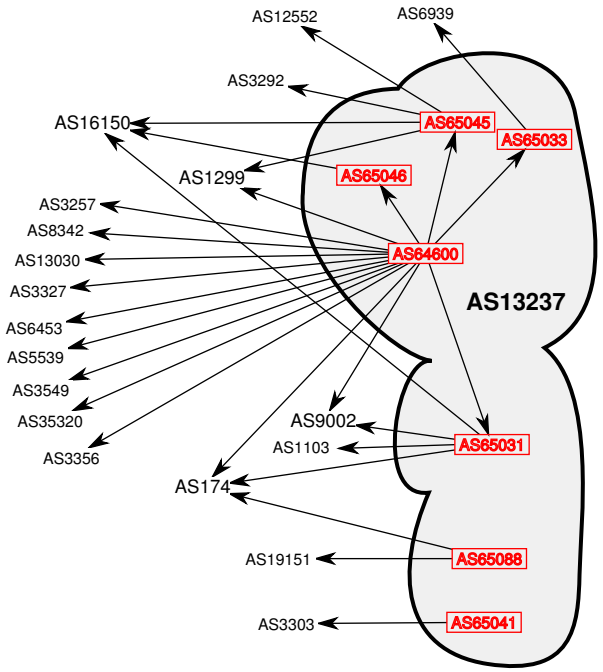


Figure 6.2: AS Confederation topology revealed by the bogus BGP-4 updates

Listing 6.1: RFC4893 violations detected with my analysis software (excerpt)

```
RFC4893 violation @ 1250489010 from (AS13237;195.66.224.99)
 with AS4PATH containing asConfedSequence
 while trying to modify: 13237 3549 9002 40965 AS_TRANS
  with 4 byte ASPATH:
   { 64600 } 3549 9002 40965 3.196

RFC4893 violation @ 1250489012 from (AS13237;195.66.224.99)
 with AS4PATH containing asConfedSequence
 while trying to modify: 13237 3549 9002 40965 AS_TRANS
  with 4 byte ASPATH:
  { 64600 } 3549 9002 40965 3.196
```

6.1.2 Induced routing instabilities

The routing storm of August 2009 reveals another weak point of BGP-4. As explained before, the storm happened because an AS Confederation started to use 32-bit ASN and confederation-internal data were advertised outside the AS Confederation. These data were flagged as invalid routing data and the BGP-4 session was repeatedly reset, inducing an oscillation. Additionally, these data were leaked beyond the first faulty link, other BGP-4 sessions started to oscillate and unrelated prefixes started to flap [5].

This routing storm is reflected in the BGP-4 repositories around the world, including the LINX. Figure 6.3 shows the aggregated update traffic profile collected by all LINX collectors for the 17^{th} and the 18^{th} of August 2009. During this period, the LINX collector received BGP-4 updates from 64 collecting devices installed in 58 Autonomous Systems. Figure 6.3 confirms the data from the Renesys blog [5] during the incident, as far as the first traffic peak is concerned. However, the Renesys curve returns to a quiet state at approximately 23:00 UTC, while the LINX data have a second peak at around 05:00 UTC of the 18^{th} of August. The traffic during this second peak was approximately 50,000 updates per 5 minute bin or 10,000 updates per minute.

Figure 6.4 shows the eCDF of the update arrival times for one of the prefixes that were affected by the incident and Table 6.2 shows the results of GNUPlot's [171] linear fitting routine applied to these data while producing the plot.

Table 6.2: GNU Plot fit results

Final set of parameters	Asymptotic Standard Error
m = 15.2496 +/- 0.1209	(0.7928%)
b = -261.093 +/- 2.473	(0.947%)

These fit results allow to compute the time the prefix was affected by the storm:

$$\Delta T = \frac{100.0}{m} = \frac{100.0}{15.25} = 6\text{h } 33\text{min}$$

and the instant the storm started affecting the prefix:

$$t_0 = \frac{-b}{m} = \frac{261.093}{15.2496} = 17\text{h :}07\text{min UTC}$$

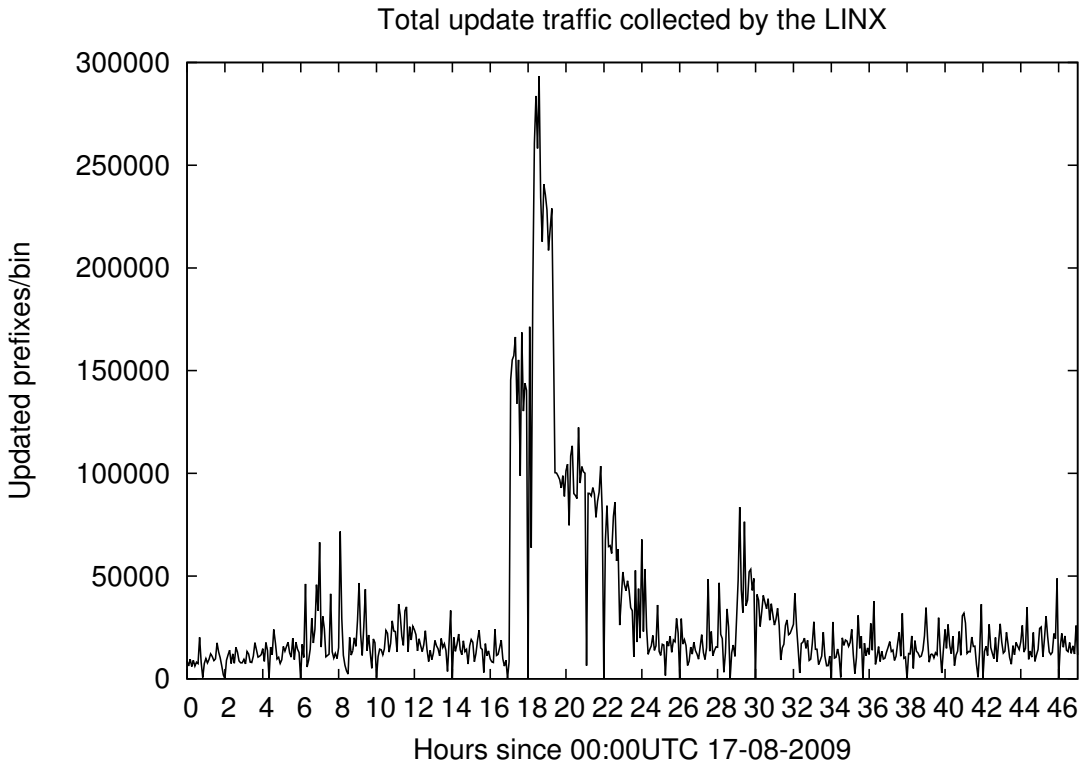


Figure 6.3: Traffic profile of the London Internet Exchange during the August, 2009 incident; time bin=5 min

The fact that the linear fit produces such good results is due to the nature of the error handling mechanism in BGP-4: when an error is detected, the session is restarted and the session restart timer has a fixed value. The slope of the linear fit is the mean time between two session resets of between a session reset and the moment the buggy update is processed by the peer (causing the next session reset).

After processing the data collected by the LINX during the routing storm in an aggregated manner, I studied them collector by collector. The results show that the LINX collectors can be classified into three groups:

1. Collectors associated to Autonomous Systems that were not affected by the storm.
2. Collectors associated to Autonomous Systems that were directly affected by the storm.
3. Collectors associated to Autonomous Systems that were indirectly affected by the storm.

There are several collectors that fall under the first category. During the storm, They exhibit a traffic profile similar to that of the days preceding the storm or after it. The second category includes all collectors that have a different traffic profile during the traffic storm. One example is shown in Figure 6.5. In this case, the traffic profile is a superposition of the traffic profile they exhibited shortly before the incident and the

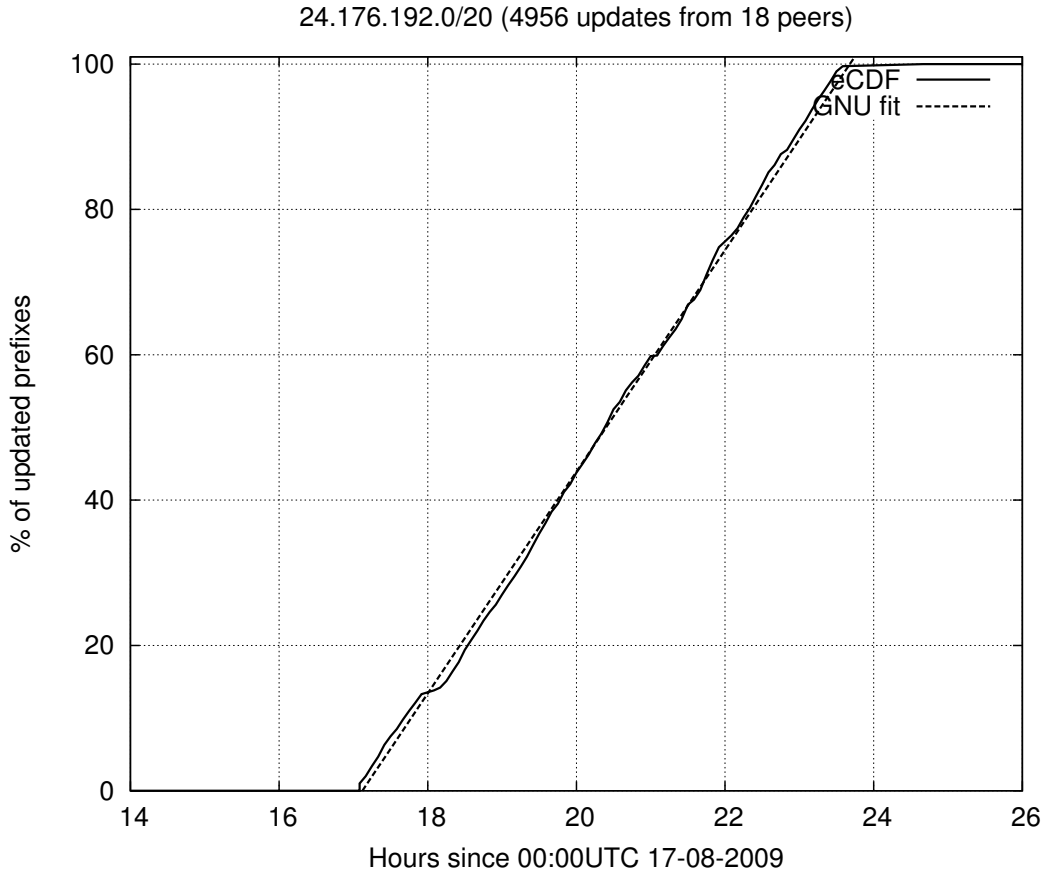
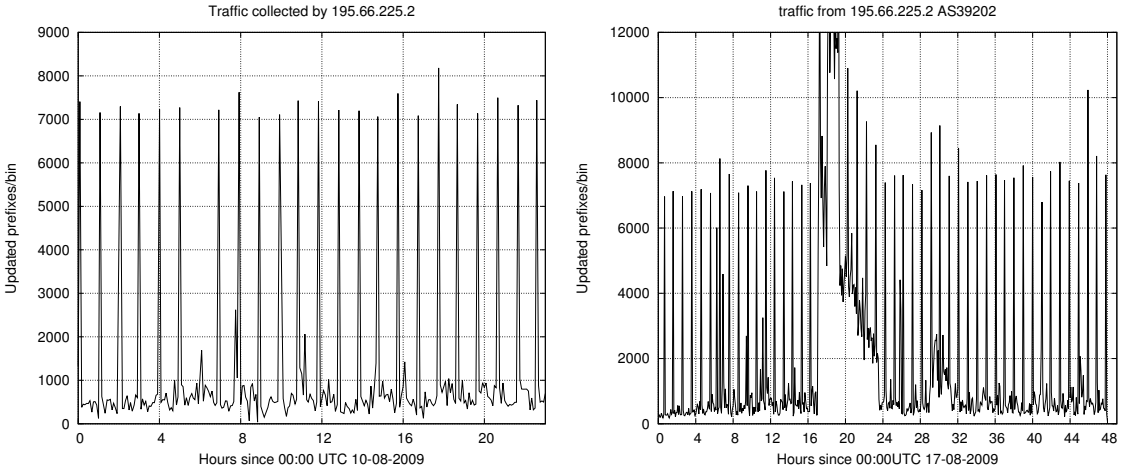


Figure 6.4: Empirical Cumulative Distribution Function of BGP-4 update arrivals for a specific prefix at the London Internet Exchange during the incident

traffic profile from the Renesys curves. This happens because the LINX is topologically far away from the origin of the incident.

There is one collector that falls under the third category. Its traffic profile is depicted in Figure 6.6. This collector detected very low BGP-4 traffic before the storm. During the storm, it exhibits a significant increase in traffic. However, the traffic profile is not compatible with the Renesys blog profile. It starts approximately one hour after the storm and is significantly shorter.

In order to understand the cause for the behaviour of this particular collector, I started by grouping the updates by prefix and studying the resulting update sequences. Figure 6.7 shows the eCDF of the update sequence lengths. More than 50% of them are more than 140 updates long. These long sequences are compatible with a routing storm; they come from links that were affected by session resets for a longer period of time. These sequences only appear during the main traffic peak. This fact suggests that they were induced by operations in other ASes during the routing storm and were suppressed manually, i.e., disabling the routing session with the oscillating neighbour or with routing policies, by local operations in the affected AS.



(a) Traffic before the incident (b) Traffic during the incident

Figure 6.5: Traffic superposition for collector 195.66.225.2

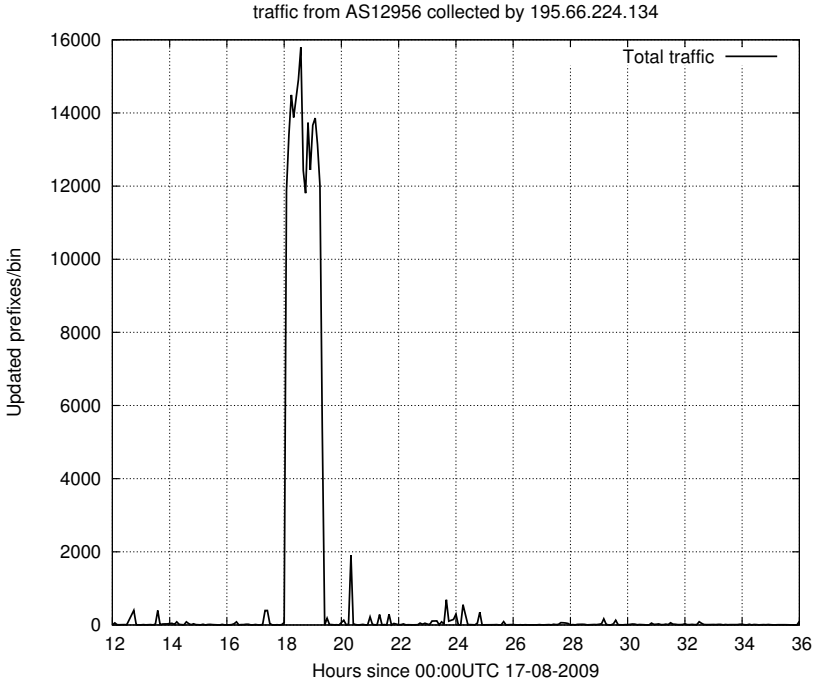


Figure 6.6: Traffic collected from AS12956 during the storm

6.2 Conclusion

This chapter has examined the effect of flaws of the BGP-4 protocol on the operation of the Internet. I have used publicly available data to show how buggy routing information and the error handling mechanisms of BGP-4 are creating sources of unexpected insta-

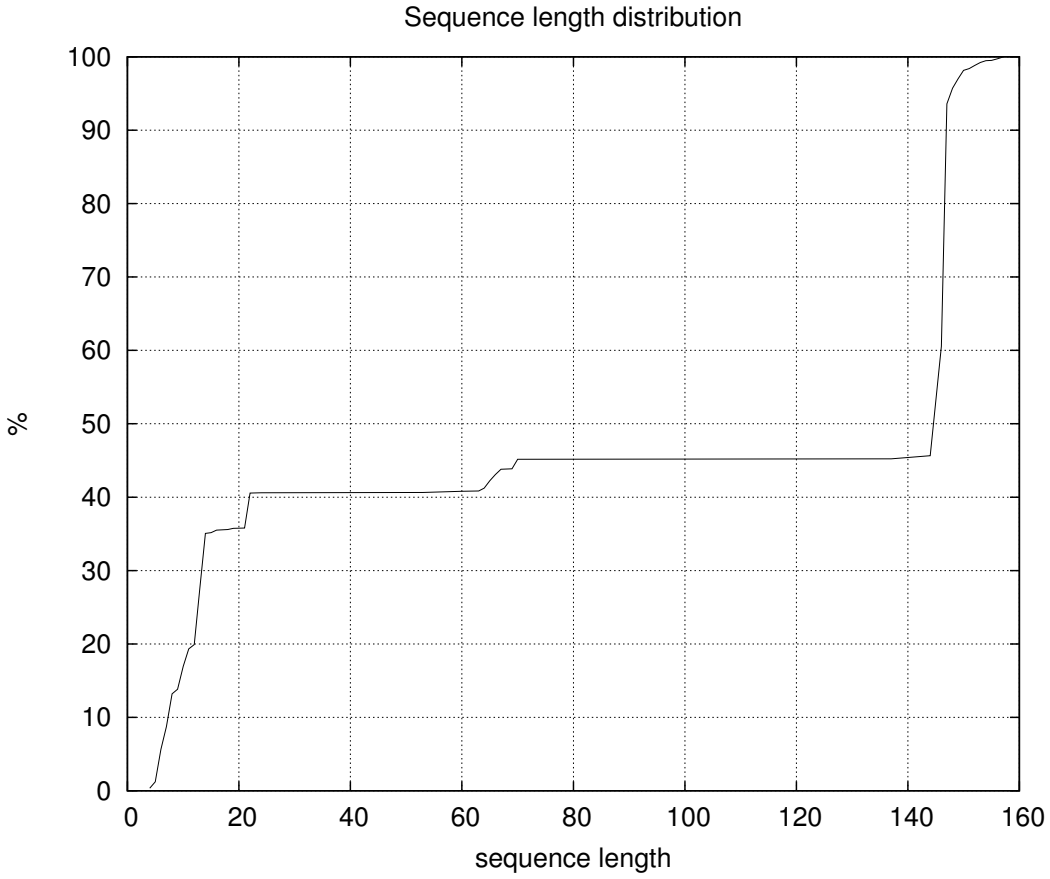


Figure 6.7: Empirical CDF of sequence lengths during the storm

bility in the Internet. This instability can be spread further by operational procedures. In my routing architecture, presented further on, the error handling mechanisms are revisited in order to provide enhanced stability compared with the current solutions.

In the next chapter, I examine how the routing tables of the DFZ in the IPv4 Internet have evolved in the decade 2001–2010. I show how practises related with Traffic Engineering have contributed to more than a third of the entries in the routing tables.

Chapter 7

Evolution of the Internet's Default Free Zone

Contents

In this chapter, I present my work on the evolution of the routing tables in the Internet's DFZ. Its growth is one of the motivations I had to develop the new routing architecture. I determine how extended the use of address fragmentation and AS_PATH Prepending is in the Internet. These two techniques are described in Section 3.4. They are identified as the basic building blocks for Traffic Engineering using BGP-4.

While the growth of the routing table in the DFZ might be a solvable issue in IPv4, it will not be so in the future IPv6 Internet. Policy document RIPE-512 [172] is very keen on stressing that Service Providers in IPv6 should concentrate on keeping the IPv6 Internet's routing table as aggregated as possible. This document has been agreed by all Regional Internet Registries and is published by all of them using their own document naming standard. For my study, I have used data coming from RIPE's Routing Repository collector rrc00, connected to RIPE's own segment at the Internet's Default Free Zone. It was the first collector established by RIPE and data from very long periods of time can be retrieved from it.

7.1 AS_PATH prepending and Address Space Fragmentation in the Internet

RIR policies for the address assignment process have assured a minimal impact on the aggregation level of the DFZ of the Internet [173]. Only when faced with the depletion of their assigned IPv4 address space, aggregation becomes less important. IANA handed out to the Regional Internet Registries the last /8 allocations of the IPv4 Internet on the 3rd of February, 2011 [174][1]. However, looking at the Internet's DFZ, disaggregation has been used massively well before the final depletion phase. In fact, some RIRs have not yet arrived to the depletion phase, but there has always been significant disaggregation in their allocated address space.

As explained in Chapter 3, influencing how traffic flows in the Internet is implemented by modifying the routing information of the different ASes. Since the AS_PATH is the only attribute that is guaranteed to be transmitted beyond the neighbouring ASes, AS_PATH Prepending is used frequently. Fragmenting the addressing space to shorter prefixes and controlling the attributes of the resulting sub-prefixes is a way to control the traffic directed to an AS in scenarios like traffic balancing.

To start with, however, the greater picture of the evolution of the Internet in the last 10 years needs to be examined the following aspects:

- How many ASes use fragmentation as part of their traffic control policies and use it in such a way that the policies disperse and are observable at topologically distant points in the Internet?
- How many ASes use AS_PATH Prepending to control the traffic directed to the prefixes they advertise in such a way that it is observable at topologically distant points in the Internet?

These two communities will be the most interesting users for my Internet architecture since they are the main cause for fragmentation and other artifacts in the Internet. Both cases refer to ASes that advertise prefixes. They are known as *leaf ASes*, because they constitute the leafs of the Internet's routing tree.

7.1.1 Address space fragmentation by leaf ASes

Address space fragmentation creates a significant overhead in terms of additional routes introduced in the Internet's DFZ routing tables. Therefore, we need to know how many ASes are actually applying fragmentation on their advertised routing space. Figures 7.1 and 7.2 show the evolution in absolute and relative terms for collectors 193.0.0.56 and 202.12.28.190 at the RIPE's routing repository rrc00 between January 2001 and December 2010. These are the only two collecting devices of rrc00 that offer uninterrupted data during this period. During this period, the amount of leaf ASes, i.e., ASes that actually advertise prefixes into the Internet, has grown almost linearly between 10,000 and 35,000. The fraction of ASes using fragmentation has never been under 20%. During the economic down-turn around 2001, the fraction of ASes using fragmentation experienced

[1]The date for the official ceremony was chosen to make it coincide with the Chinese New Year.

a reduction. It then kept at an approximately constant level until 2007. From that point onwards, a slight upward trend can be observed. Nowadays, slightly under 10,000 leaf ASes are using fragmentation These and their immediate upstream providers are the clear addressees of my alternative routing architecture.

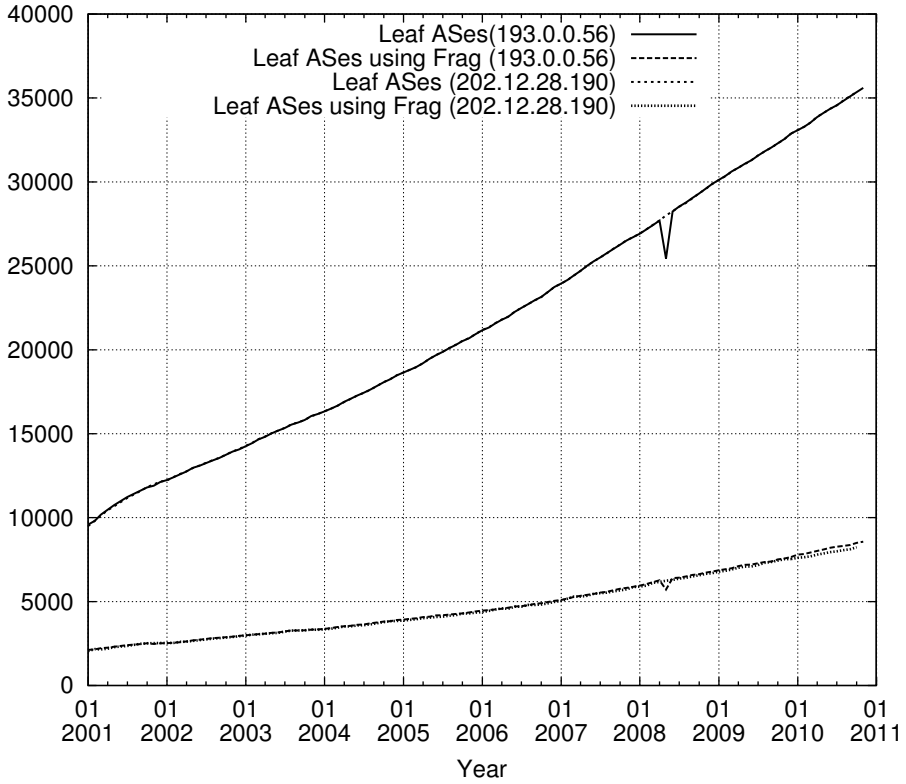


Figure 7.1: Evolution of the total number of leaf ASes and of leaf ASes using fragmentation in the Internet between 2001 and 2010

7.1.2 Use of AS_PATH Prepending in the Internet

The other trend I needed to study is the use of AS_PATH Prepending by the ASes at the edge of the Internet. I used the same collectors as in the previous subsection. I scanned the routing tables they contributed for downstream ASes that were applying AS_PATH Prepending on their prefixes.

The evolution is shown in Figure 5.1, and basically shows that over 60% of the downstream ASes observed by the collectors were using AS_PATH Prepending. This figure shows only the lower bound for the real usage of AS_PATH Prepending. As the example in Figure 3.1b on Page 20 shows, intermediate Autonomous Systems receiving two advertisements for a given prefix discard the advertisement with AS_PATH Prepending because they select the advertisement with the shorter AS_PATH. This might also happen further towards the core of the Internet.

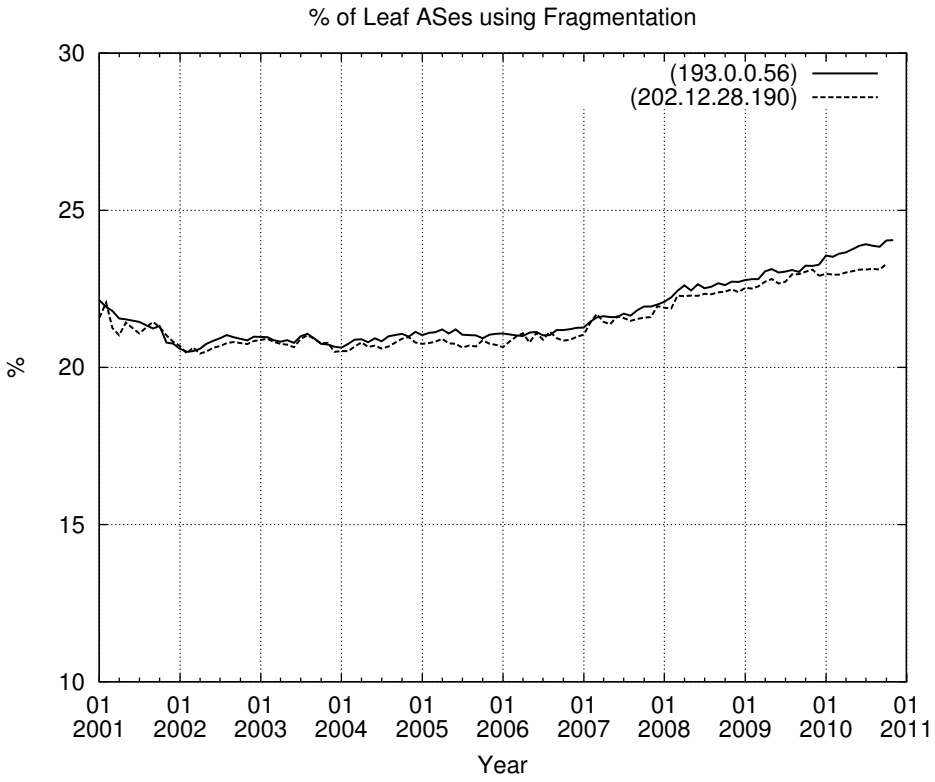


Figure 7.2: Evolution of the percentage of leaf ASes using fragmentation between 2001 and 2010

7.1.3 Behaviour of intermediate Autonomous Systems confronted to disaggregation

BGP-4 foresees the possibility for an AS to aggregate the routing information received from other ASes. This process is, however, restricted to specific conditions in RFC 4271 [2]. During my work, I studied how Internet Service Providers use fragmentation and how intermediate ASes treat the fragmented advertisements.

Table 7.1: Example of disaggregation in an Internet routing table

Prefix	AS_PATH
41.196.112.0/23	9304 4788 8452 8452 8452 24863
41.196.114.0/23	9304 4788 8452 8452 8452 24863
41.196.116.0/23	9304 4788 8452 8452 8452 24863
41.196.118.0/23	9304 4788 8452 8452 8452 24863
41.196.120.0/23	9304 4788 8452 8452 8452 24863
41.196.122.0/23	9304 4788 8452 8452 8452 24863
41.196.140.0/23	9304 4788 8452 8452 8452 24863
41.196.142.0/23	9304 4788 8452 8452 8452 24863
41.196.160.0/23	9304 4788 8452 8452 8452 24863
41.196.162.0/23	9304 4788 8452 8452 8452 24863
41.196.164.0/23	9304 4788 8452 8452 8452 24863
41.196.166.0/23	9304 4788 8452 8452 8452 24863
41.196.168.0/23	9304 4788 8452 8452 8452 24863
41.196.170.0/23	9304 4788 8452 8452 8452 24863
41.196.172.0/23	9304 4788 8452 8452 8452 24863
41.196.174.0/23	9304 4788 8452 8452 8452 24863
41.196.176.0/23	9304 4788 8452 8452 8452 24863
41.196.178.0/23	9304 4788 8452 8452 8452 24863

Table 7.1 shows an example of an intermediate AS, AS8452, that applies the same policy to all of the advertisements it receives from one of its clients, AS24863. This client is a leaf AS that advertises 18 adjacent and, at least partially, aggregate-able prefixes. This shows that disaggregation introduced at the edges of the Internet is propagated to the core, where it does not seem to be used. This only leads to increased resource usage in the DFZ routers in the form of memory to hold the RIB with all the parameters and computing power in the algorithms computing the FIB from the RIBs.

7.2 Address space fragmentation

As mentioned above, disaggregation is a popular tool in the IPv4 Internet. Houston et al. monitor this evolution and publish statistics on the CIDR Report web page both for IPv4 [175] and IPv6 [176]. However, the algorithm they use has not been published. In

order to provide a reproducible method to estimate disaggregation, I have designed and implemented my own aggregation algorithm.

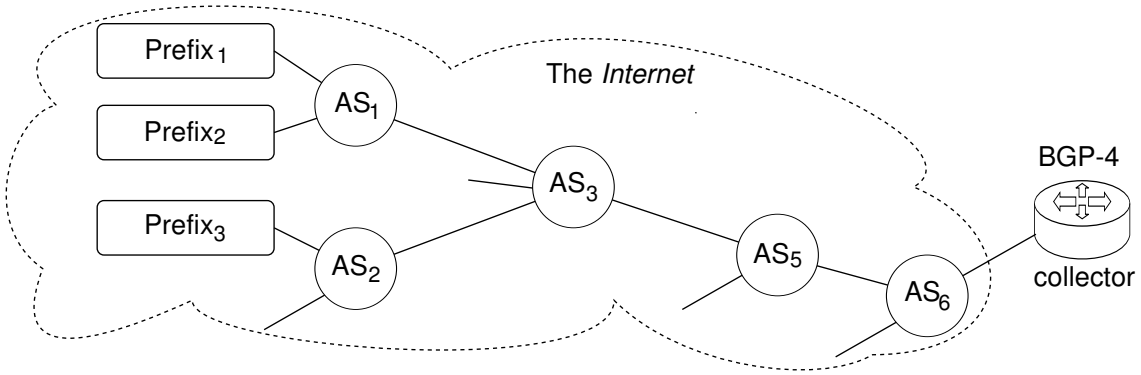


Figure 7.3: A view of the Internet from a Routing Repository

Prefixes assigned by the different Regional Internet Registries to Autonomous Systems needing public IP addressing are part the public addressing space as per RFC 1930 [14]. At any given point in time a prefix is assigned to one and only one AS, although due to the dynamics of the Internet, the assignments may differ for different points in time. In order to detect the configurations presented in Chapter 3 and eliminate sub-netting, I model the Internet routing table as a directed graph as shown in Figure 7.3. The root of the graph is the router the routing table was extracted from and the ASes present in the AS_PATH are the vertexes of the graph. The leaves of the graph are $\{\text{AS}, \text{Prefix}\}$ pairs that represent the address allocations made by the different RIRs to the ASes in their regions.

7.2.1 Estimating the Aggregation Potential in Routing Tables

In order to estimate the aggregation potential in routing tables, I propose Algorithm 3. It is not intended to be applied directly in routers, but rather helps assessing the overhead introduced by current practises to implement multi-homing over different providers, load balancing, protection against prefix-hijacking, etc. Therefore, optimisations were not sought and computational time analysis has not been performed. The algorithm is applied on an initial routing table until no new aggregations can be found. The concepts of sub- and super-netting are interpreted restrictively, in the sense that prefixes are associated to the AS that originated them and to the AS_PATH they are received through and sub- or super-netting is only allowed when both prefixes belong to the same AS and are reached through the same sequence of Autonomous Systems. The algorithm uses the following functions, which are graphically presented in Figure 7.4, to check for possible optimisations:

- `nextAggregation(prefix)` returns a prefix generated by keeping the base address and decrementing the prefix mask length by one.
- `IsFeasible(prefix)` checks whether the prefix is correct, i.e., that all bits in the prefix host field are zero.

- `Contains`($prefix_1$,$prefix_2$) checks that both prefixes are originated by the same AS and that both prefixes are reached following the same AS_PATH and that $prefix_2$ is completely contained in $prefix_1$.

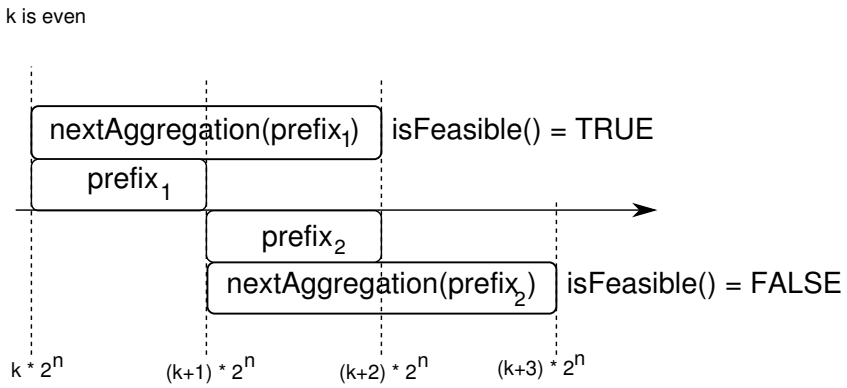


Figure 7.4: Prefix operations used in Algorithm 3

Algorithm 3 preserves the AS_PATHs from the BGP-4 collector to end hosts. The BGP-4 routing table of a router is a directed graph. The root of the graph is the router itself and each leaf contains a prefix that can be reached from the root paired with its AS. The other nodes of the graph represent the ASes traversed by a packet on its way to a given prefix. This graph has two types of edges; the regular edges connecting two nodes and the irregular edges connecting a node with itself. These edges result from policies in the ASes along the AS_PATH.

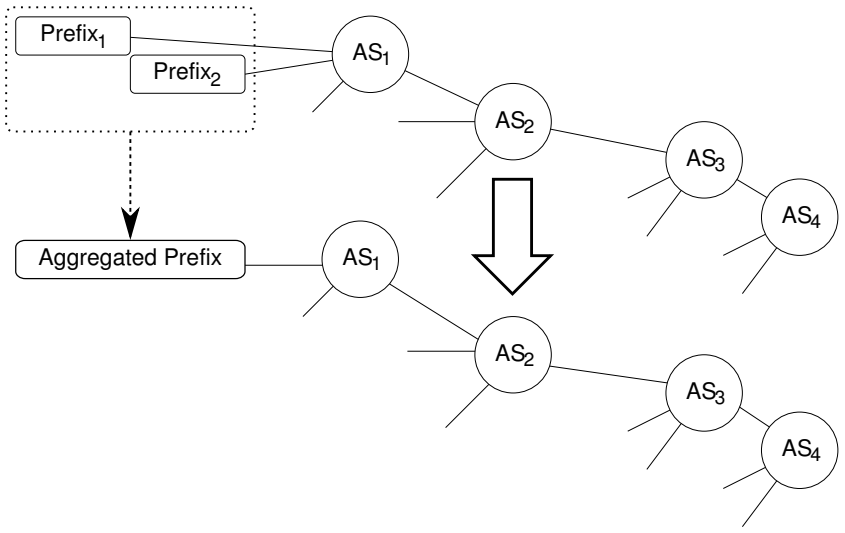


Figure 7.5: Aggregating two prefixes

Algorithm 3 will only merge two paths of the directed graph if they share all nodes expect the leafs and if the leafs refer to prefixes that are assigned to the same AS and can

be aggregated, as shown in Figure 7.5. The algorithm respects the address allocations made by the RIRs at the time the routing table was collected and the paths followed by packets at AS level and thus produces equivalent routing tables, in the sense that packets will arrive to their assigned destinations. Two different modes of comparing the AS_PATH are possible:

- *strict* comparison including the artificial edges introduced by AS_PATH Prepending artifacts, or
- *relaxed* relaxed comparison eliminating these artificial edges.

Data: InetTable, a routing table as an array of $\{prefix, AS\_PATH\}$ pairs, ordered in ascending order by the base address of the prefixes
Result: The routing table with one extra level of aggregation and a flag indicating whether the routing table was modified or not.

```
changed ← false;
foreach index = 0 to length(InetTable) − 2 do
    this_Prefix ← InetTable[index];
    next_Prefix ← InetTable[index + 1];
    aggregateThis = nextAggregation(this_Prefix);
    if IsFeasible(aggregateThis) then
        if Contains(aggregateThis,nextPrefix) then
            /* remove next_Prefix from the Internet table          */
            removeFromTable(InetTable[index + 1]);
            /* replace this_Prefix with the aggregation           */
            InetTable[index] ← aggregateThis;
            /* signal that the table has changed                  */
            changed ← true;
return changed,InetTable
```

Algorithm 3: Routing table aggregation algorithm

As a proof of concept for Algorithm 3, I took all the routing tables contributed by running IPv4 collectors to the rrc00 repository on the 1st of September, 2009 and applied Algorithm 3 on them. In total, 11 collectors were running that day. I took all the tables and applied the reduction algorithm on each. Figure 7.6 shows the evolution of the mean number of routes in the interval size and the mean number of routes aggregated in each iteration.

Significant aggregation potential could be detected. The mean initial routing table size was 287,200 routes. After eight iterations, no further aggregation was possible in any table. The resulting routing table had a size of 206,900 routes. Speaking in relative terms, the table could be reduced by 28.0% without losing connectivity. Figure 7.6 shows the reduction achieved on the first 6 iterations. It is worth to be noted that the first iteration reduces the routing table by 18.9%. The improvement is mainly obtained in the first 4 iterations, where a reduction by approx. 27.7% is achieved.

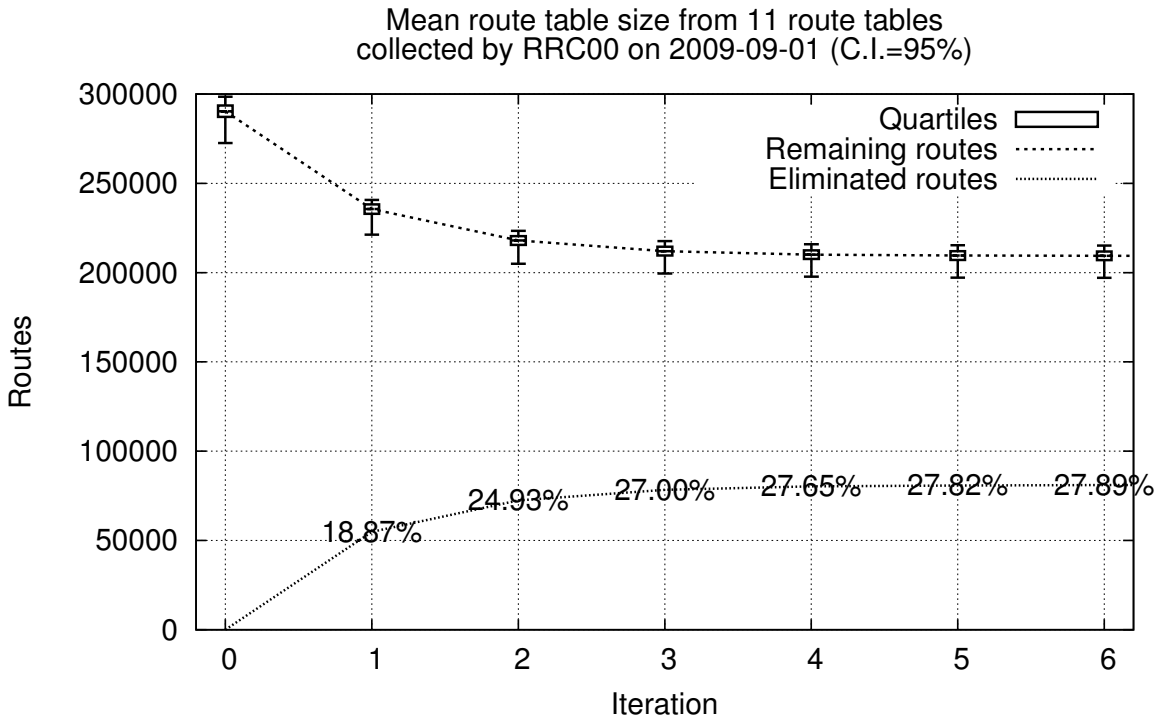


Figure 7.6: Reduction achieved on average on the first 6 iterations for all default free routing tables collected by rrc00 on the 1st of September, 2009

Table 7.2: Resulting mean table length in the first 6 iterations of the algorithm applied to the routing tables collected by rrc00 on the 1st of September, 2009 for different route matching strategies in the `Contains()` function (C.I.=95%)

Iteration	Full AS_PATH	Canonical AS_PATH	Origin only
0	2.904e+05 +/- 3223	2.904e+05 +/- 3223	2.904e+05 +/- 3223
1	2.356e+05 +/- 2626	2.346e+05 +/- 2642	2.255e+05 +/- 2385
2	2.18e+05 +/- 2474	2.165e+05 +/- 2512	2.015e+05 +/- 2035
3	2.12e+05 +/- 2442	2.103e+05 +/- 2494	1.924e+05 +/- 1889
4	2.101e+05 +/- 2432	2.082e+05 +/- 2491	1.89e+05 +/- 1832
5	2.096e+05 +/- 2429	2.076e+05 +/- 2490	1.878e+05 +/- 1811
6	2.094e+05 +/- 2429	2.074e+05 +/- 2489	1.875e+05 +/- 1806
7	2.094e+05 +/- 2429	2.074e+05 +/- 2489	1.874e+05 +/- 1804
8	2.094e+05 +/- 2429	2.074e+05 +/- 2488	1.874e+05 +/- 1804

Table 7.2 shows how Algorithm 3 performs with different implementations of the `Contains(`$prefix_1$`,`$prefix_2$`)` function. The sample is constituted by the routing tables stored by the **rrc00** collector on the 1st September, 2009. The first column indicates the iteration, the other columns show the mean routing table size at a confidence level

of 95% for different comparison strategies. The second column shows the mean routing table size when the full AS_PATH attribute is used by the aggregation algorithm. I used the Canonical AS_PATH as defined in Section 5.2.2 for the third column, and the Autonomous System Number of the originator for the fourth column. This last column represents the resulting routing table size for the Internet's DFZ if the the RIR allocations were not fragmented by the ISPs. The difference between the third and the fourth column in Table 7.2 is due to the prefixes which are advertised through connections to different ASes. The difference between the second and the third column in Table 7.2 is due to configurations compatible with Figure 3.1b where one of the links is not operational: taking the full AS_PATH into account, the prefixes will not be reduced, whereas removing the AS_PATH Prepending artifacts will make the prefixes reducible.

7.2.2 Evolution of fragmentation between 2001 and 2010

Finally, I studied the evolution of fragmentation in the Internet between the 1st of January, 2001 and the 31st of December, 2010. I used routing tables provided by collectors associated to rrc00 that were active during that whole period, based on the information provided by Table 3.2. I took the first routing table collected in each month, applied the aggregation algorithm, and used the resulting routing table sizes until the third iteration.

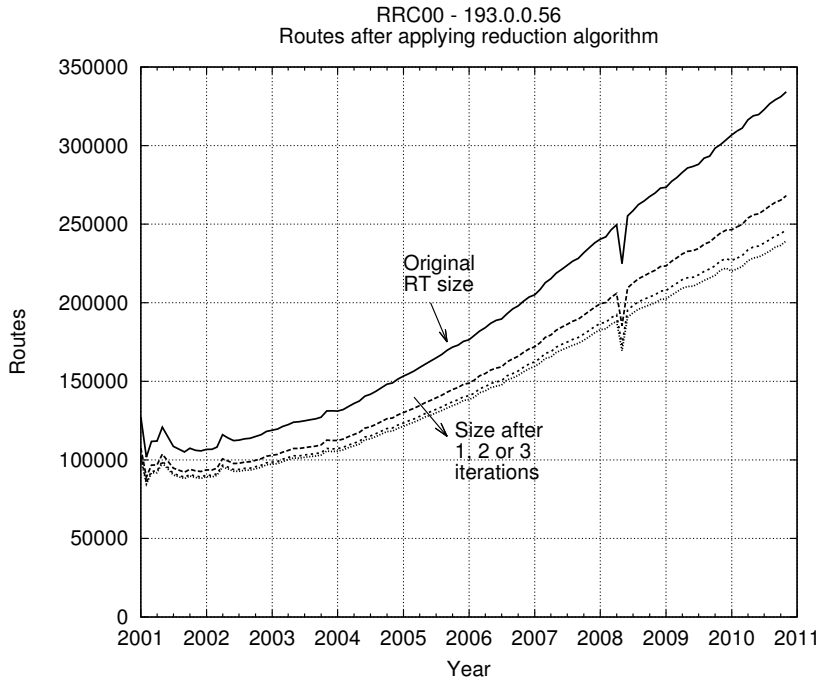
Figure 7.7 shows the reduction of the global routing table in the DFZ obtained running the first three iterations of Algorithm 3 on the routing table contributed to rrc00 between 2001 and 2010 by 193.0.0.53 (Figure 7.7a) and 202.12.28.190 (Figure 7.7b). Both figures show that the IPv4 routing table in the Internet's DFZ has not ceased to grow in the last ten years, despite the economical downturns experienced and the depletion of the IPv4 address space. Additionally, it is noteworthy that the gap between the curves widens continuously during the whole observation period.

In order to get a better view of the evolution of the gap between the iterations, I measure the disaggregation of the Internet routing tables as the ratio between the routes that were eliminated by Algorithm 3 and the initial size of the routing table:

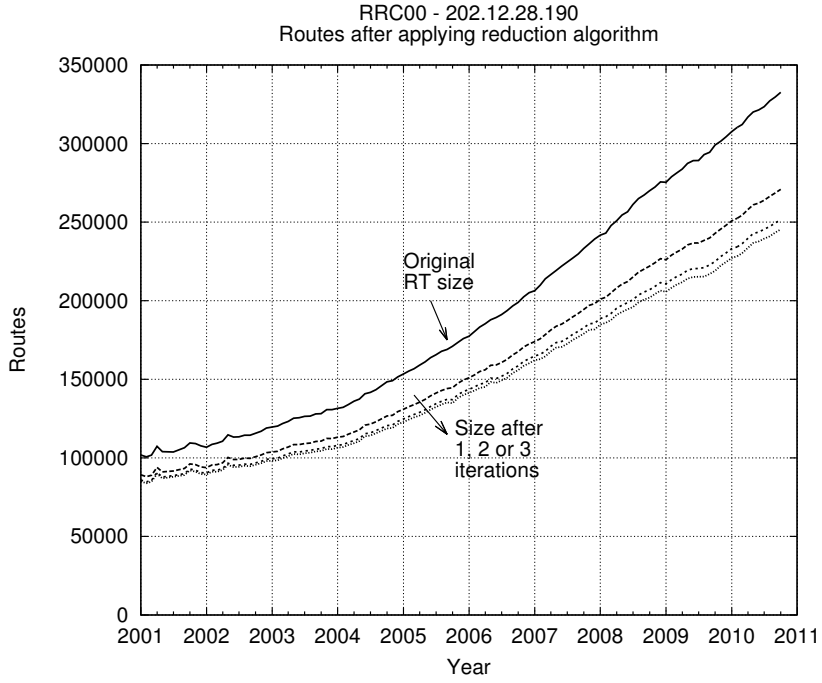
$$\rho = \frac{N_{initial} - N_{aggregated}}{N_{initial}}$$

Figure 7.8 shows the evolution of this aggregation ratio during the period of observation for the two collectors selected previously. They show that ρ grew linearly between 2002 and 2009, while during 2010, ρ remained approximately constant. Possible explanations for this behaviour are:

- *The deep economic crisis* that started around that time, *which would have slowed down the growth of the Internet.* However, no graph in Figure 7.7 suggests that this happened, because during 2010, the IPv4 routing table continued to grow. Moreover, as Figure 7.1 shows, the number of leaf ASes continued to grow during 2010 as in the previous years despite the economical downturn. Additionally, as

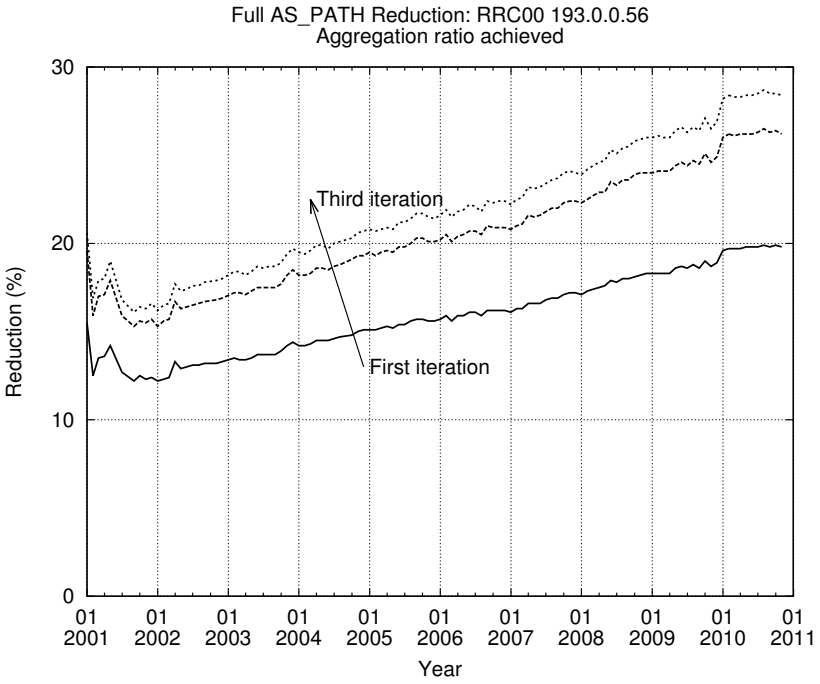


(a) Collected by 193.0.0.56

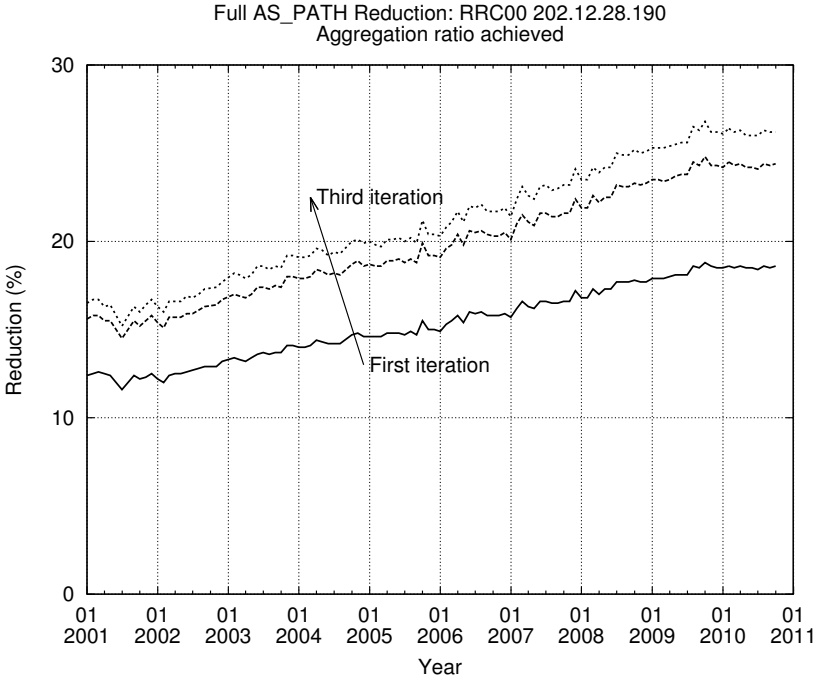


(b) Collected by 202.12.28.190

Figure 7.7: Evolution of the routing table sizes for the first three iterations of Algorithm 1 on DFZ routing tables from rrc00



(a) Collected by 193.0.0.56



(b) Collected by 202.12.28.190

Figure 7.8: Evolution of ρ for the first three iterations of Algorithm 1 on DFZ routing tables from rrc00

Figure 7.2 shows, there is a slight upward tendency in the use of address space fragmentation by leaf ASes.

- *The depletion of the IPv4 routing space that has forced the RIRs to allocate smaller prefixes to ASes.* This would translate in less possibilities to fragment the addressing space, given that the smallest prefixes that can be advertised to the Internet are /24 [177]. However, the flattening began more than one year before the last /8 prefixes were handed over to the Regional Internet Registries.
- The *transformation in the Internet's core* observed by Labovitz et al. [178]. In this recently published paper, the authors argue that the structure of the Internet has changed radically. Some of the ASes in the core of Internet have experienced traffic growth because they are the ones that host the most popular applications, sites, etc. According to the authors, the core ASes have evolved from simple traffic exchanges to traffic sources. Thus, they are no longer interested in controlling their input traffic. This task has been passed to the new consumer ASes, who are charged by the traffic volume they consume. Labovitz et al. observe the behaviour of several core Internet providers in their paper and show how there is a turning point between 2009 and 2010, when some significant ASes ceased to behave as transit ASes and started exhibiting traffic patterns typical for ASes with important traffic sources. As of writing this work, another move to consolidate the core of the Internet has happened with the merger of two major Internet players: Global Crossing and Level-3 [179]. It remains to be seen how this merger will affect the structure of the core of the Internet.

It is therefore highly plausible that the flattening of the ρ curves is linked to the last cause. However, it has to be noted that the point at which this flattening happens is 30%. Therefore, any architecture capable of significantly reducing the level of fragmentation in the IPv4 routing table may by useful in the transition phase between IPv4 and IPv6. A smaller IPv4 routing table means that the router has to devote less resources to manage it. There resources can be made available for other tasks, like handling a growing IPv6 table or implementing the IPv4/IPv6 transition mechanism of choice for any given ISP.

Conclusion

This chapter has examined the overall evolution of the Internet routing tables between 2001 and 2010. It has shown that the fragmentation of the IPv4 accounted for about a third of the routes in the Internet's Default Free Zone in 2010.

In the next chapter, I present my proposal for an alternative routing architecture that can be deployed at the edges of the Internet and will result in smaller routing tables at the core of the Internet. I explore possible implementation strategies and discuss my prototype implementation and how it would impact the overall behaviour of the Internet.

Chapter 8

An Alternative Routing Architecture Based on Parallel Routing Tables

Contents

In the previous chapters, I have analysed the short-comings of the current Internet routing architecture.

This research shows how overloading BGP-4 with additional features in order to cope with new services opens the door to a less stable Internet in the future and has motivated me to design an alternate routing architecture that simplifies the use of BGP-4 in the core of the Internet.

This work concentrates on the use of parallel routing planes and on the benefits to the routing plane in the core of the Internet when this architecture is used at the edges. BGP-4 is known to have security flaws and these are addressed elsewhere. My architecture will benefit from these security enhancements once they are implemented in BGP-4 routing daemons. However, security problems are not the main focus of my work.

This chapter is based on material published in the MONAMI 2010 [11] and MONAMI 2011 [13] conferences and in the Mobile Networks and Management Journal [12].

8.1 PaRArch: A Routing Architecture for the Internet based on Parallel Routing Tables

BGP-4 is the protocol that conveys reachability information between Autonomous Systems in the Internet. In the previous chapters, I have shown

1. that introducing new features in the BGP-4 protocol has led to routing storms due to the current error detection and response mechanism of the protocol,
2. that routing operations in one Autonomous System propagate through the Internet, and
3. that around 30% of the entries in the Internet DFZ routing tables are used for local traffic distribution purposes and can be avoided with adequate aggregation

These inefficiencies have motivated me to design and implement an alternate routing architecture for the Internet. It is backward compatible, non-mandatory and may be adopted incrementally in different regions in the Internet, especially at its edge. It contributes to stabilising the Internet in the following ways:

1. Smaller routing tables mean a less resource-intensive BGP-4 process and better debugging capabilities, avoiding the problems linked to multi-homing scenarios described in Section 3.4.2.
2. It isolates the main Internet routing table from TE-related phenomena like those presented in Chapter 5 and from instabilities due to bugs introduced by new protocol features like those described in Chapter 6.

In order to implement this new architecture, I propose to introduce a new routing plane in the Internet. In this section, I discuss the routing plane structure and the AS-level design of PaRArch and then discuss the architecture of a PaRArch enabled router.

8.1.1 Routing planes in PaRArch

I use the term "routing plane" in the sense of network realm where routes with specific semantics are treated. Looking at the current Internet, we find the following categories of routing information[1]:

- The *inter-domain* routing information needed to reach hosts outside an AS that is handled by an EGP, the de-facto standard EGP being BGP-4.
- The *intra-domain* routing information that is handled by one or more IGPs like OSPF [24], IS-IS [25], etc. within the AS.

These two routing information categories are carried by two different routing planes:

1. the *interior routing plane* with fine-grain routing information that is used to reach specific destinations within an AS
2. the *exterior* or *inter-domain routing plane* with coarser-grain routing information that is used to transfer packets between Autonomous Systems.

This use of the term "plane" is compatible with the way that GROW uses it when talking about route-reflection planes [180].

PaRArch extends the routing plane structure of the Internet by an additional routing plane. This routing plane is deployed between adjacent ASes that have a multi-homed connection as shown in Figure 3.1. ASes that are near from a topological point of view, i.e., those that are two hops away, and have multiple paths that connected them can also use this architecture (an example is shown further down in Figure 8.8). Since the routing information carried by PaRArch is exchanged between ASes, the routing protocol has to be an EGP. Therefore, the designated routing protocol for the PaRArch routing plane is BGP-4. The routing information carried by the PaRArch routing plane is kept local between the peering ASes, i.e., it will not be injected in the global inter-domain routing plane.

As shown previously with Algorithm 3, a significant portion of the Internet routing tables can be aggregated. Using this insight, I classify inter-domain routing information into two categories:

1. Best aggregations of the Internet routing table that are obtained when Algorithm 3 is not able to further aggregate the routing table.
2. De-aggregated networks: networks of the original routing table that are replaced with more aggregated networks by Algorithm 3.

PaRArch uses this classification to assign prefixes to routing planes: (i) the global inter-domain routing plane carries the best aggregations; and (ii) the new inter-domain routing plane introduced with PaRArch carries the de-aggregated routing information. This task split assures (i) that all prefixes can be reached from any point in the Internet and (ii) that peering agreements between ASes can be implemented without bloating the size of the Internet's DFZ routing table.

[1] Provider-specific implementations using MPLS may include more categories like MPLS-VPN.

8.1.2 Interaction between routing planes in PaRArch

The interior and the exterior routing planes of the Internet interact whenever there is the need to transfer routing information from one plane to another. Ideally, this interaction is minimal in order to guarantee that the inter-domain routing plane is stable and is achieved by injecting or redistributing aggregated routing information from the interior into the exterior routing plane at designated points in the AS [181]. eBGP information needs to be redistributed into the IGP in specific cases. However, the use of iBGP in the AS is preferred in these cases [62].

PaRArch defines a stricter prefix redistribution policy. As shown in Figure 8.1, the IGP can inject any level of aggregation into the PaRArch inter-domain routing plane. However, it can only inject the best aggregations to the global inter-domain routing plane. Redistributing routing information from the PaRArch or the Internet inter-domain routing plane into the intradomain routing plane is explicitly forbidden. Whenever information coming from the PaRArch routing plane is needed, the PaRArch routing plane is extended using iBGP inside theAS.

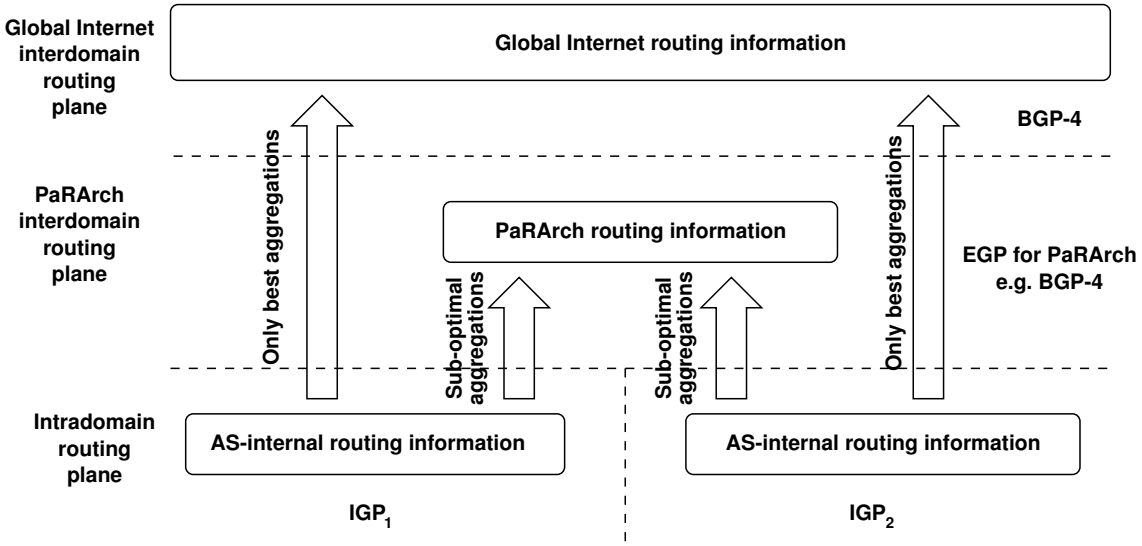


Figure 8.1: Redistribution policies between routing planes in PaRArch

Interconnecting chains of ASes using a PaRArch routing plane is possible. In the current version of the PaRArch routing architecture, this happens through independent routing realms established between pairs of peering ASes. The possibility of creating a PaRArch routing realm spanning more than two ASes is out of the scope of this work and left for future study.

8.1.3 AS-level deployment of PaRArch

PaRArch is designed to be deployed at the edge of the Internet because that is the main source of disaggregation. It has to be compatible with setups like the one shown in

Figure 3.1, where leaf ASes advertise their assigned address space using disaggregation over different links with different policies. However, the first common upstream AS consolidates the advertisements and all prefixes are advertised to the next upstream AS with the same AS_PATH. PaRArch is based on the principle that this routing information, which is needed for local purposes like traffic balancing or other traffic engineering purposes between two ASes, is exchanged exclusively between the leaf AS and the first upstream AS and kept in a routing table that holds this specific (and local) routing information and not in the global Internet routing table. Additionally, and in order to guarantee that the leaf AS is reachable from beyond the first upstream AS, the leaf AS advertises the best aggregation – usually the RIR address delegations – to the global Internet routing tables.

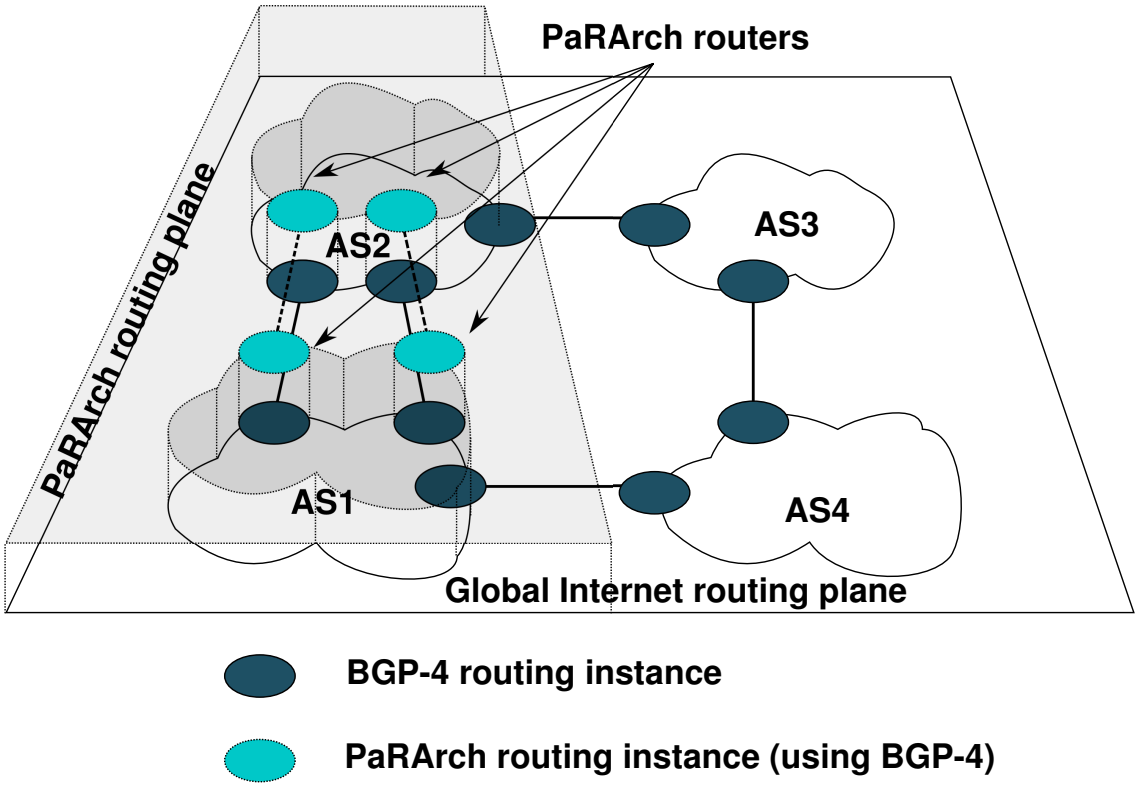


Figure 8.2: Incremental deployment of PaRArch between adjacent ASes

Figure 8.2 shows how this new architecture can be deployed between a leaf AS and its first upstream provider: AS1 and AS2 have signed a peering agreement that requires a specific distribution of traffic among the links that interconnect them. They use PaRArch the following way:

- AS1 is the client and needs to control the traffic it received from its provider. It requests the use of PaRArch from AS2 and configures the peering using the following rules:
  1. In order to assure that AS1 is globally reachable, it advertises its best aggregations on all its peering in the global Internet routing plane. This includes

the peering with AS4. Policies based on AS_PATH Prepending control how the Internet beyond its direct peers (AS3 in Figure 8.2) is going to send the traffic to it.

2. In order to comply with the traffic distribution agreed upon in the SLA, AS1 uses address space fragmentation; it advertises sub-prefixes of the address space it has been assigned by its RIR on the PaRArch routing plane.

- AS2 is the provider. It announces that it offers a connectivity service based on PaRArch. When a client requests the service, it enables a PaRArch peering to it.

This architecture can also be deployed in scenarios where PaRArch-enabled ASes are not directly connected, as shown in Section 8.3.4.1. Figure 8.3 shows how it would be accomplished.

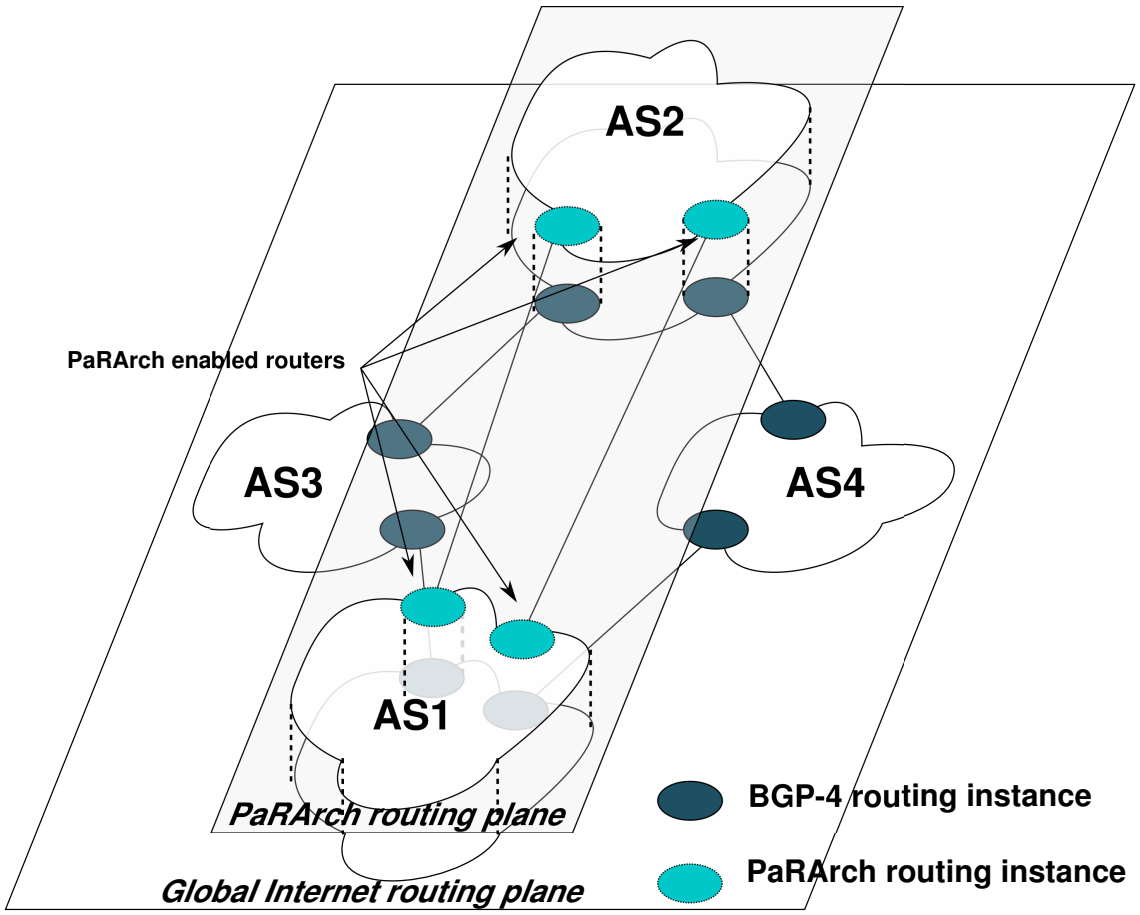


Figure 8.3: Incremental deployment of PaRArch between non-adjacent ASes

The advertising strategy is the same as above – best aggregations in the Internet routing plane and disaggregated prefixes in the PaRArch routing plane. In this case, PaRArch works like this:

- The best aggregations advertised by AS1 assure that the intermediate ASes AS3 and AS4 can reach AS1.

- The unaggregated prefixes advertised through the PaRArch routing plane allow AS2 to balance traffic directed to AS1.

8.1.4 High-level design of a PaRArch enabled router

Figure 8.4 shows a block diagram of the BGP-4 section in a PaRArch enabled router. PaRArch adds an additional BGP-4 infrastructure with a dedicated RIB to a normal BGP-4 router. The original BGP-4 routing instance handles the main RIB and is in charge of the routing information that is present in the Internet's routing table. This RIB holds the best aggregations of the PaRArch peers. The local routing setups, expressed as sub-networks of the best aggregations, are migrated to a second routing table. This second routing table is only maintained between and managed by the PaRArch peers and implemented in an additional BGP-4 routing instance[2]. The RIBs managed by both routing instances are consolidated by the Route Decision Processs of BGP-4 and the PaRArch EGP. The routes selected by these RDPs are fed back to the BGP-4 and PaRArch routing instances marked as routes that are advertised to peers and injected into the Forwarding Information Base (FIB) of the router. In case of collision, i.e., when a prefix is present in the PaRArch and the global Internet routing planes, PaRArch routes take precedence over global Internet routes.

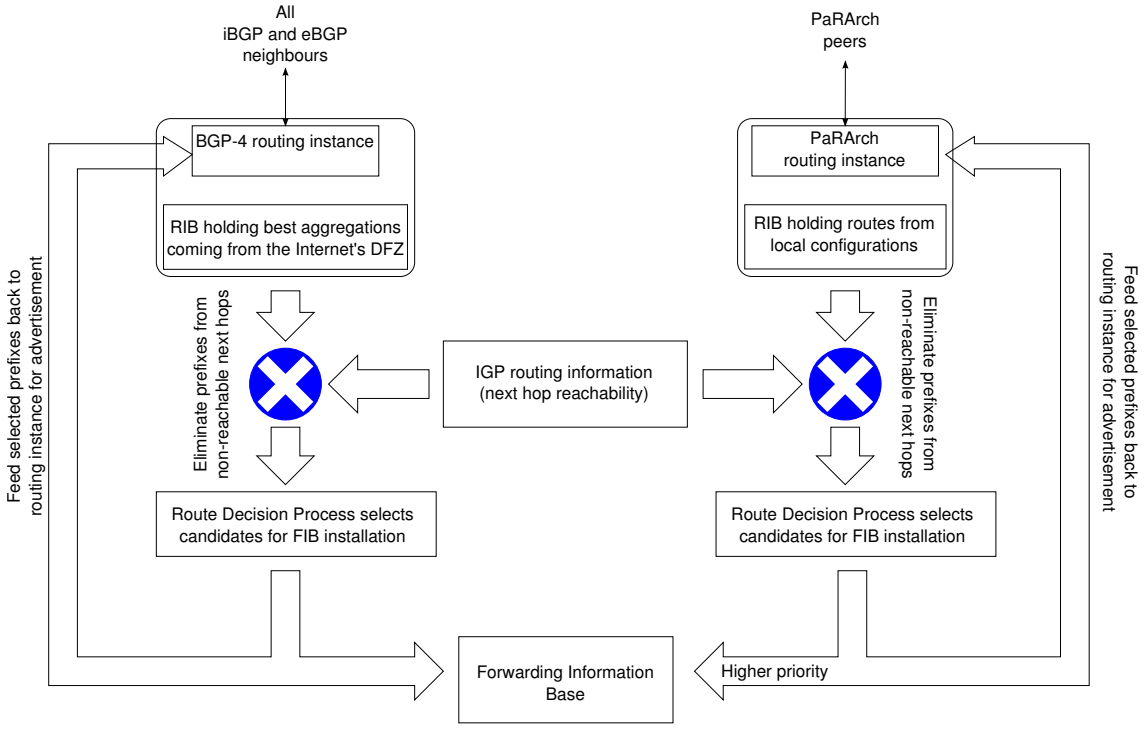


Figure 8.4: A high level view of the PaRArch routing architecture at the router level

[2]In a generalised design, any EGP could be used. Since BGP-4 is the current de-facto standard EGP, the implementation I propose uses BGP-4.

Figure 8.4 shows the relationship between the different RIBs and the FIB in a PaRArch-enabled router. The support for the BGP-4 information destined for the Internet's DFZ is provided by the left side of the figure. It represents the architecture of a state-of-the-art BGP-4 router. The right side of Figure 8.4 shows the PaRArch infrastructure of the router.

The novelty of this approach is having two isolated EGP instances running in parallel and prioritising how the entries from their respective RIBs are inserted in a common FIB in the router. In a traditional router, there is a similar process between the EGP and the IGP. Here the IGP has always less preference than the EGP. When more than one EGP instances are used, like in current MPLS-based VPN environments, the routing tables are isolated and there is no interaction between routing entries coming from different MPLS-VPN routing instances.

8.2 Prototype implementation

In the previous section I have presented the PaRArch architecture. In this setion, I present a proof of concept implementation of a PaRArch router and show how a deployment of the architecture could look like using a network emulation environment, to confirm that it is feasible and that the basic functionality works as expected.

8.2.1 Implementation alternatives

Before selecting the final environment for my proof of concept implementation of PaRArch, I pondered the advantages of using an emulated versus a simulated environment. I went for an emulated environment, because, at this stage, functional completeness was more important for me than precise timing. Another motivation for using the emulated environment was that I could reuse an open-source carrier-grade implementation of the basic router functionality as the foundation of my implementation and concentrate on implementing the distinguishing aspects of the PaRArch architecture.

8.2.1.1 Simulation environments

While the results regarding timing on a simulation environment can be very accurate, the implementation of multiple RIBs and their interactions with the FIB need to be as close as possible to those in a real router. Implementing these interactions in a simulated environment introduces additional complexities. I was confronted to them when I tried to implement a simulation based on JSim [182], a Java-based simulation environment with which I have reasonable experience. I participated in the implementation of the simulator used in the 4WARD project [183] to evaluate a BGP-4-based migration scenario [184] and there I saw that my objectives with regard to multiple routing tables where completely out of scope for JSim. It does not implement the Route Decision Process completely and has no way of determining whether the next hop is available or not.

Additionally, the simulated architecture needs then to be exported or migrated to a router code base in order to provide a practical implementation. This is possible to

some extent with simulation packages like ns2 [185] or ns3 [186]. However, having limited prior experience in either of them, I felt I would not have the necessary confidence in the results of my simulations.

8.2.1.2 Network emulation environments

Network emulation environments like Netkit [139] or GNS3 [187] are discussed in Section 4.5.1. They provide a realistic environment to test real router implementations. For my experiments I finally selected Netkit for the following reasons:

1. At the time of starting my development it was a stable environment while GNS3 had just been released and was still under heavy development.
2. I had already used both Netkit and Quagga previously and could start my development fairly quickly.
3. The main added-value of GNS3 is the possibility of using real-world router operating system images. However, PaRArch is not supported by any vendor and there is no access to the source code behind those images to modify them and integrate PaRArch functionality in them.

8.2.1.3 Open-source routing toolkits

The different source code bases for BGP-4 are discussed in Section 4.5.2. For the purpose of my experiments, XORP [147] and Quagga [83] were the best candidates available when I started the project. An additional advantage of the Quagga-on-Netkit environment was that I had prior experience in developing and testing modifications to the BGP-4 routing protocol on this environment. This experience reduced the learning curve significantly.

8.2.2 Implementation

The proof of concept for PaRArch was implemented in a prototype based on a modified Quagga daemon and tested on a Netkit environment. Routing table management is implemented directly using features provided by Linux. The Linux kernel supports multiple IPv4 and IPv6 routing tables [188] that can be assigned different priorities. These priorities are controlled with the `ip rule` command. Documented usage of this routing table prioritisation mechanism is to provide isolation between traffics coming from or directed to an interface in a Linux machine [189].

The default rules installed on a Linux router on startup are:

```
0:  from all lookup local
32766:  from all lookup main
32767:  from all lookup default
```

Routing tables are included into the lookup mechanism with the `ip rule add` command. By default, any new table is added before the main routing table. This effectively means that it has a higher priority than the main routing table when installing routes into the

FIB. In my implementation, I use routing table *10* and add it to the lookup rules with the command `ip rule add from all lookup 10`. This results in the following lookup rules:

```
0:  from all lookup local
32765:  from all lookup 10
32766:  from all lookup main
32767:  from all lookup default
```

The modifications introduced into the Quagga routing suite[3] are the following:

1. The Zebra protocol [145] used to exchange routing information between the *zebra* daemon and the different routing daemons (*bgpd*, *ripd*, *ospfd*, etc.) has been enhanced with a new optional field that holds the kernel routing table identifier. Routing table query and set operations can now address any kernel routing table.
2. The *zebra* daemon was modified to handle multiple kernel routing tables in the Netlink protocol interface. Netlink is the user-kernel communication protocol that handles routing tables in Linux equipment. It is defined in RFC 3549 [190].
3. The command line interface of the *bgd* daemon was enhanced to accept a table identifier on the command line. Some minor inconsistencies in the command line interface regarding definition of the TCP port used for the BGP-4 protocol were corrected.

8.3 Proof of Concept

For the proof of concept for the multi-table, multi-daemon architecture I selected a scenario where a leaf AS and one of its upstream providers decide to migrate the traffic control to a parallel routing table. I used an emulated testbed that mimics the layered three-tier structure of the Internet as observed in 2010 [178].

8.3.1 Use Case

The use case for the proposed architecture is that of a provider and a client AS at the border of the Internet. Both are interconnected by two geographically independent links. They have established an interconnection agreement that mandates that the downstream traffic (from provider to client) has to be balanced between the two links and never exceed 50% of the total traffic. This ensures that the surviving link can take over all the downstream traffic in case of failure of either of the links. The scenario is sketched in Figure 8.2 and the test cases are explained in Section 8.3.2. The use case also shows how an incremental deployment at the edges of the Internet of PaRArch would look like.

[3]See Section 4.5.2 for a description of the Zebra and Quagga routing protocol suites.

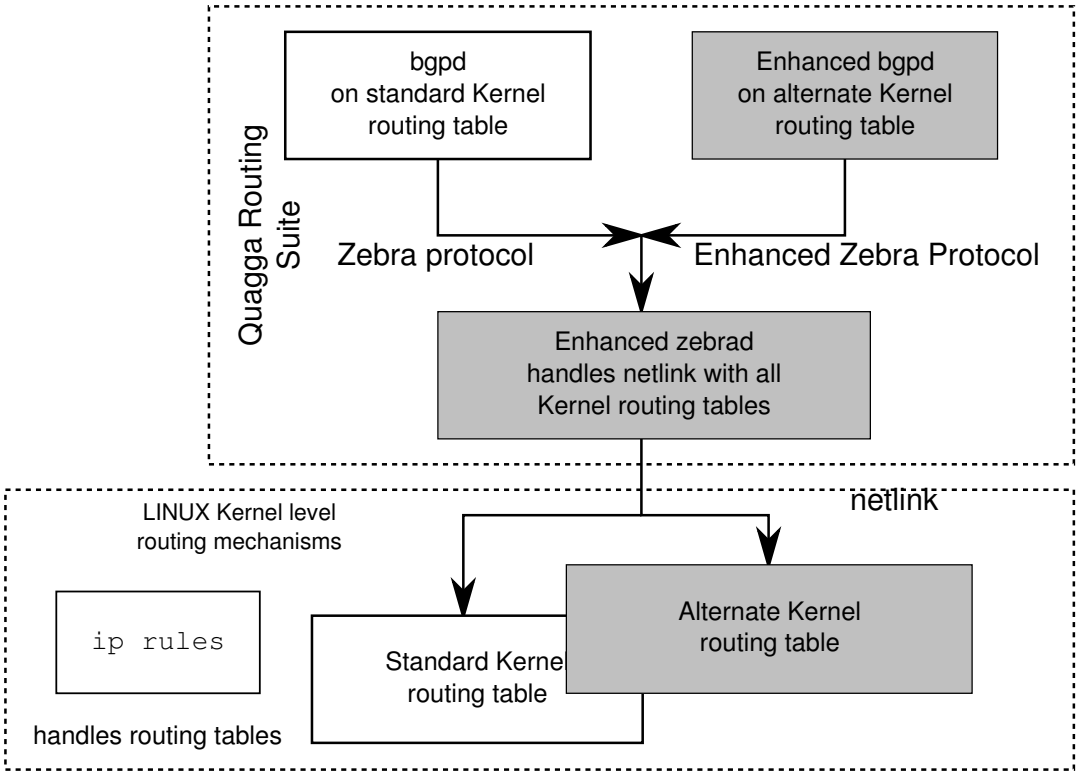


Figure 8.5: High-Level View of the Prototype implementation on Linux

8.3.2 Test cases for a PaRArch-enabled router

The objective of the PaRArch architecture is to provide flexibility in the interconnection of a leaf AS by keeping the possibility of locally using routing configurations implemented using address space fragmentation, and, at the same time, isolate the Internet DFZ from these configurations. This results in two generic use cases. In the following, I refer to the proof of concept topology shown in Figure 8.6.

Traffic Engineering in the local routing plane

This test case shows the basic functionality of PaRArch. A pair of neighbouring ASes decides to setup a parallel routing plane in order to control the traffic they exchange. In this case, we need to test the following functionalities:

1. **Setting up PaRArch:** it should be possible to establish a parallel routing plane between AS1012 and AS1002.
2. **Traffic should flow as intended:** the traffic distribution between the two links should be controllable in such manner that traffic directed to either of the two traffic sinks h_6402 and h_6502 should flow using either interconnection link depending on the locally programmed policy.

Stable global routing plane

This test case shows how the rest of the world perceives a PaRArch region. From the outside world, this region should be seen as a stable region of the Internet that advertises clients with the best aggregation possible. The main feature that needs to be tested is that:

1. **local routes should not leak into the global routing plane**. Independent from any local policy, the other Autonomous Systems should receive a stable routing advertisement with the best aggregation only.

8.3.3 Testbed Implementation

The development and tests of the modified Quagga and a proof of concept were implemented in a Netkit [139, 191] environment. The topology is shown in Figure 8.6. It follows the general principle of the layered three-tier topology proposed by Labovitz et al. [178]. All Autonomous Systems are implemented using one router only. The central core layer is implemented by four fully meshed ASes, r_100 to r_103. The second layer is implemented with r_1000, r_1001, r_1002 and r_1003. r_1002 implements the upstream PaRArch Autonomous System. The third layer is implemented again by r_1010, r_1011 and r_1012.

This last router is the core of the downstream PaRArch Autonomous System AS1002, which includes two hosts with active probes. These hosts are connected to the infrastructure through two different Local Area Networks (LANs) that have prefixes that are different but can be aggregated: 6.0.4.0/24 and 6.0.5.0/24. In order to show that traffic control at the link level is made possible by PaRArch with no routes leaking from the PaRArch routing plane to the main routing plane, I advertise these prefixes on the links in the PaRArch routing plane and the aggregate prefix (6.0.4.0/23) on the main routing plane. I use a simple routing policy deployment script that changes the AS_PATH Prepending of the two prefixes advertised by AS1012 in the PaRArch routing plane. Since this routing plane is controlled by specific BGP-4 daemons, the control scripts connect to the PaRArch daemon. As shown in Listing 8.1, the script is implemented using expect [192] and the Quagga CLI language. This is equivalent to current practises in real networks using Cisco routers.

In order to test the effect on the path followed by packets, I use active tomography probes (implemented with traceroute [193]) on the test hosts. Figure 8.7 shows the graphical representation of the result of an experiment that lasted 8 hours. During this time, the path was probed every 15 seconds and the policy was changed every five minutes (as shown in Listing 8.2). The figure shows the IP addresses of the interfaces and paths detected by the traceroute probes. The table shows the mean round-trip time measured by the probe as well as number of probing instances that detected each host. The interfaces in the second hop where not traversed by the same number of probes because the measurement process and the route changer script where launched independently.

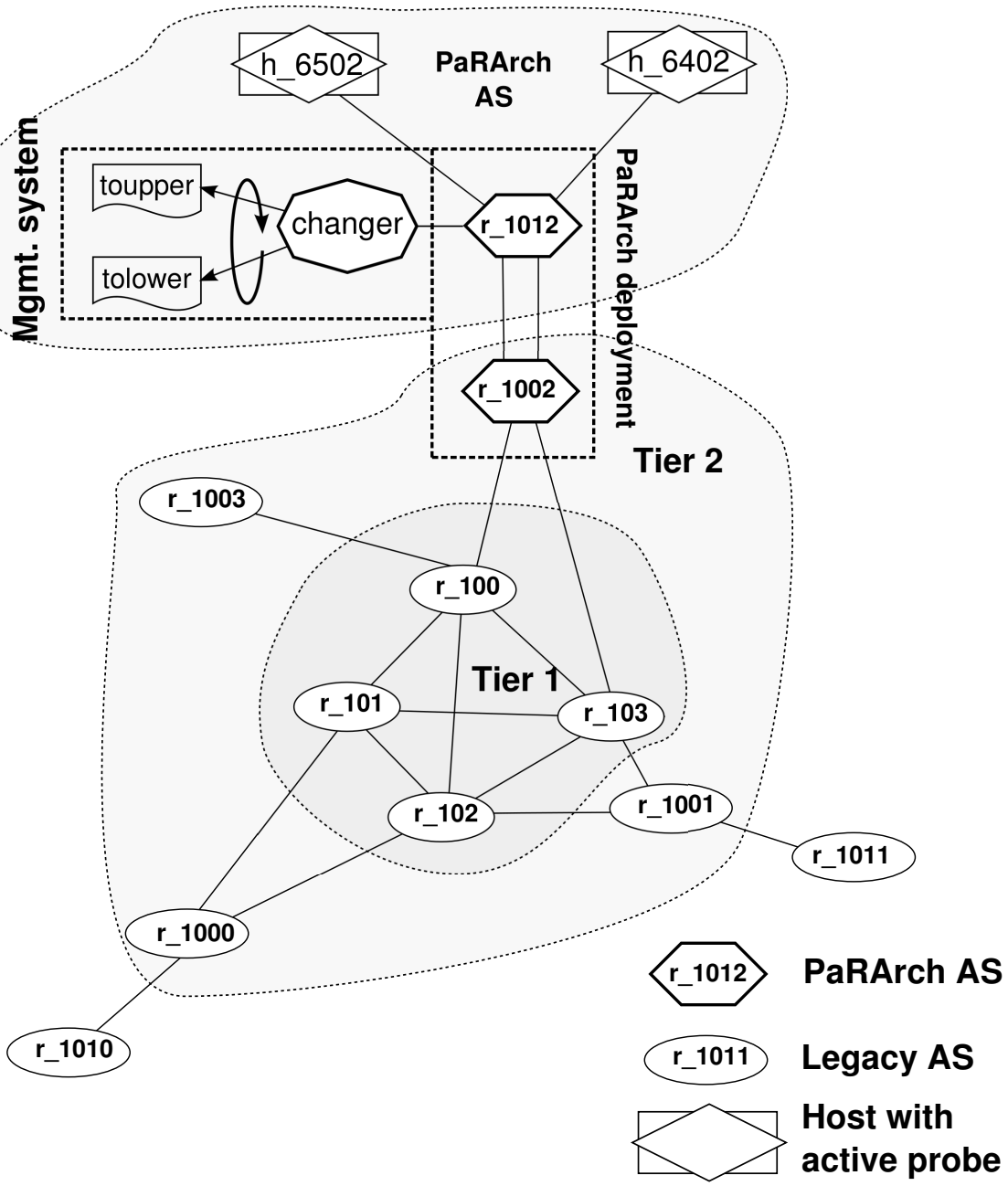
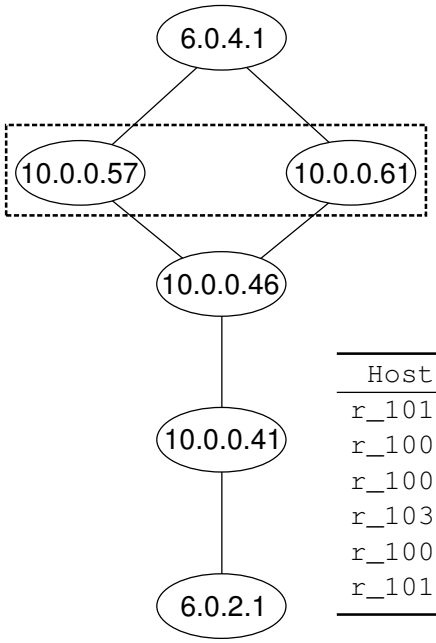


Figure 8.6: Proof of concept topology

8.3.4 Evaluation of the Traffic Balancing Use Case

The proof of concept network emulation environment was used to compare different traffic balancing techniques that can be considered current practises. Two different situations were examined:

1. Stub AS with first upstream (Tier 1) AS
2. Stub AS with Tier 0 AS



Host	IP	N	RTT
r_1012	6.0.4.1	1910	0.436 ms
r_1002	10.0.0.61	954	0.892 ms
r_1002	10.0.0.57	956	0.897 ms
r_103	10.0.0.46	1910	1.361 ms
r_1001	10.0.0.41	1910	1.854 ms
r_1011	6.0.2.1	1910	2.100 ms

Figure 8.7: Graphical output of the tomography tool

The stub AS case has been used in other instances [70]. Taking into account the consolidation process in the Internet, the number of Tier2 ASes connected to only one provider will grow. These ASes should be connected by more than one link and will need to use disaggregation in order to implement traffic balancing. In these cases, PaRArch in its current IPv4 implementation or in a future version supporting IPv6 depending on the evolution of this protocol, could provide a valuable tool to avoid further growth of the Internet routing tables.

During my tests, I compared PaRArch with the following, well-known techniques to differentiate between the two upstream links of AS1012:

1. *Marking using well-known communities*: The well-known NO_ADVERTISE community is used to direct the traffic to the two sub-networks through the different links.
2. *Marking using Multi-Exit Discriminators*: The MED attribute is used to signal preference for the incoming traffic.
3. *AS_PATH Prepending*: Different lengths in the AS_PATH attribute signal the preference of the two links.

These techniques were compared with the PaRArch architecture. The criteria used for this comparison were whether the sub-nets are advertised in the Internet's DFZ or not, whether during this process they keep the metric information for use further upstream and whether operation and maintenance procedures may result in accidental leakage of prefixes. I used in all cases a correct configuration that implemented the intended traffic flows.

Table 8.1: Comparison between different BGP-4 control techniques

	Subnets advertised	Subnets keep metric	Risk of leak	Impact on DFZ's routing table size
Community NO_ADVERTISE	No	N/A	Yes	Leaked subnets progress to the Internet
MED	Yes	No	N/A	N/A
AS_PATH Prepend	Yes	Yes	N/A	The subnets progress to the Internet
PaRArch	No	N/A	No	The subnets do not progress to the Internet

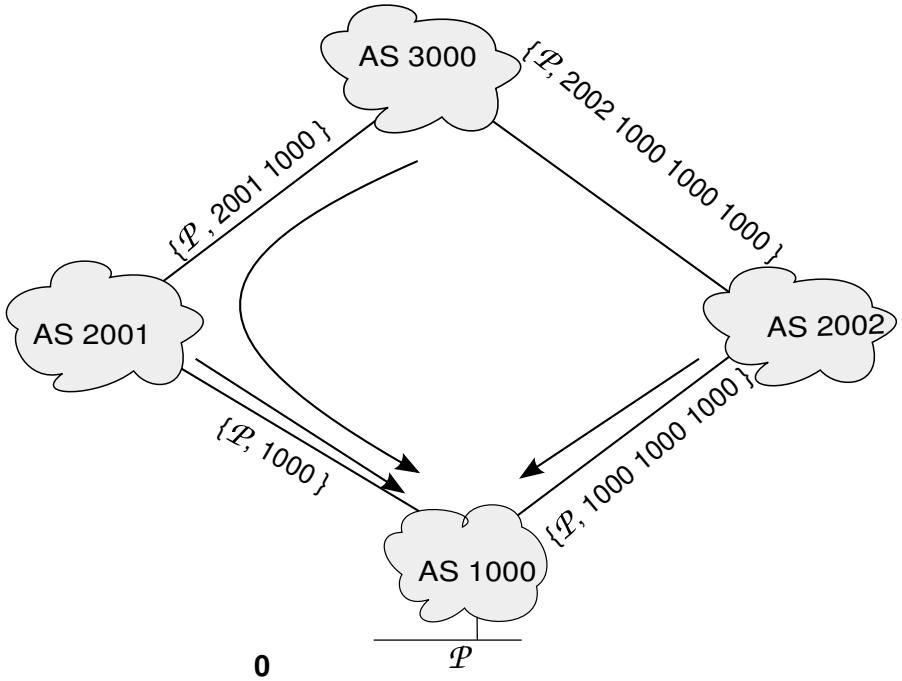
This comparison is shown in Table 8.1. It can be argued what *quality criteria* to use in this classification. I prefer either not to advertise at all, or making sure that once a prefix is advertised, the routing information is correctly mapped to the preferences for inbound traffic defined by the AS that advertises them. In this sense, the Multi-Exit Discriminator or the NO_ADVERTISE community provide a valid solution. However, they have their drawbacks: MEDs are known to oscillate [194] and any community-based marking can be easily interfered with by a router misconfiguration. Regarding routing table size growth in the Default Free Zone, AS_PATH Prepending performs worse than the proposed architecture. The same applies for the Multi-Exit Discriminator and the NO_ADVERTISE community in case of misconfiguration.

8.3.4.1 Interaction with a Tier-1 provider

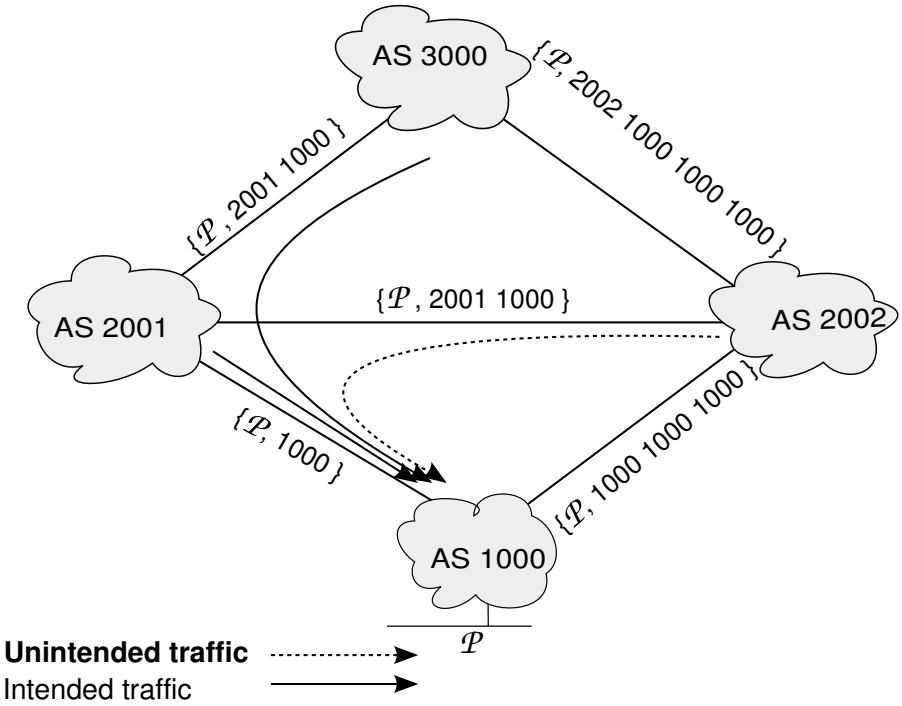
The four marking techniques shown in Table 8.1 only apply between directly connected ASes. If a leaf AS like AS1012 needs to interact with a Tier-1 AS, only PaRArch or AS_PATH Prepending can be used. In this case, the leaf AS needs to establish the BGP-4 sessions for PaRArch using multi-hop and the ASN of the Tier-1 AS. The Tier-1 AS also needs to know the peering information for the leaf AS that is requesting the connection.

Figure 8.8 shows an example of a routing configuration with unintended side effects that can be avoided by PaRArch. Listing 8.4 shows how the additional link between AS2001 and AS2002 creates unintended routing effects on current BGP-4 implementations. Listing 8.3 shows that this does not happen with PaRArch. AS1000 advertises more specific prefixes to AS3000 in order to control the main traffic over the PaRArch routing plane. AS2001 and AS2002 receive the aggregated prefixes without AS_PATH Prepending. AS2001 and AS2002 therefore always compute better reachability through the direct links and traffic diversions like the one presented in Figure 8.8 do not happen.

Listing 8.5 shows the main routing table at AS3000 when the link between AS2001 and AS2002 is down. Listing 8.6 shows the main routing table at AS3000 when the



(a) Original TE configuration



(b) Unintended traffic flows due to AS_PATH Prepending

Figure 8.8: Unintended side effects of Traffic Engineering using AS_PATH Prepending

link between AS2001 and AS2002 is up. When comparing it to Listing 8.4, it can be easily seen that in the absence of AS_PATH Prepending, BGP-4 has chosen the

shortest paths between the Autonomous Systems. Additionally, the main routing table hold information of the most aggregated prefixes, i.e., 192.168.0.0/23 for AS1000 and 192.168.4.0/23 for AS3000.

Listing 8.7 shows that the routes followed by packets to AS1000 when link is up and down are the same. The policies are applied in the PaRArch routing plane of r_1000 using the configuration shown in Listing 8.8. This configuration needs no readjustment depending on the state of the connection between AS2001 and AS2002.

Listing 8.1: Policy enforcement scripts

```
#!/usr/bin/expect
if {[llength $argv] == 0} {
    set site "localhost"
} else {
    set site [lindex $argv 0]
}
if {[llength $argv] == 2} {
        set port [lindex $argv 1]
} else {
        set port "bgpd"
}
spawn telnet $site $port
expect "Password: "
send "zebra\r"
expect "AS1012> "
send "ena\r"
expect "AS1012# "
send "conf t\r"
expect "(config)# "
send "router bgp 1012\r"
expect "(config-router)# "
send "neighbor 10.0.0.57 route-map upper out\r"
expect "(config-router)# "
send "neighbor 10.0.0.61 route-map lower out\r"
expect "(config-router)# "
send "end\r"
expect "AS1012# "
send "clear ip bgp * soft out\r"
expect "AS1012# "
send "exit\r"
```

Listing 8.2: Script that triggers a policy change

```
#!/bin/bash
if [ -z "$*" ]; then
    router="localhost"
else
        router=$1
fi
echo Router is $router
while [ true ]; do
        /hostlab/toupper $router
        date "+%R %F"
        sleep 300
        /hostlab/tolower $router
        date "+%R %F"
        sleep 300
done
```

Listing 8.3: Intended routing configuration before the link is activated

```
BGP table version is 0, local router ID is 192.168.3.1
BGP table ID is 0
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop    Metric LocPrf Weight Path
*  192.168.0.0      10.0.0.14                      0 3000 2001 1000 i
*>                  10.0.0.5         0             0 1000 1000 1000 i
*> 192.168.1.0      10.0.0.5         0             0 1000 i
```

```
*  192.168.2.0      10.0.0.14                      0 3000 2001 i
*>                  10.0.0.5                       0 1000 2001 i
*> 192.168.3.0      0.0.0.0          0         32768 i
*  192.168.4.0/23   10.0.0.5                       0 1000 2001 3000 i
*>                  10.0.0.14        0             0 3000 i

Total number of prefixes 5
```

Listing 8.4: Unintended routing configuration after the link is activated

```
BGP table version is 0, local router ID is 192.168.3.1
BGP table ID is 0
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop    Metric LocPrf Weight Path
*> 192.168.0.0      10.0.1.1                       0 2001 1000 i
*                   10.0.0.14                      0 3000 2001 1000 i
*                   10.0.0.5         0             0 1000 1000 1000 i
*> 192.168.1.0      10.0.0.5         0             0 1000 i
*> 192.168.2.0      10.0.1.1         0             0 2001 i
*                   10.0.0.14                      0 3000 2001 i
*                   10.0.0.5                       0 1000 2001 i
*> 192.168.3.0      0.0.0.0          0         32768 i
*  192.168.4.0/23   10.0.1.1                       0 2001 3000 i
*                   10.0.0.5                       0 1000 2001 3000 i
*>                  10.0.0.14        0             0 3000 i

Total number of prefixes 5
```

Listing 8.5: BGP-4 table at r_3000 when link between AS2001 and AS 2002 is down

```
BGP table version is 0, local router ID is 192.168.5.1
BGP table ID is 0
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  192.168.0.0/23   10.0.0.13                              0 2002 1000 i
*>                  10.0.0.9                               0 2001 1000 i
*  192.168.2.0      10.0.0.13                              0 2002 1000 2001 i
*>                  10.0.0.9                 0             0 2001 i
*  192.168.3.0      10.0.0.9                               0 2001 2002 i
*>                  10.0.0.13                0             0 2002 i
*> 192.168.4.0/23   0.0.0.0                  0         32768 i

Total number of prefixes 4
```

Listing 8.6: BGP-4 table at r_3000 when link between AS2001 and AS2002 is up

```
BGP table version is 0, local router ID is 192.168.5.1
BGP table ID is 0
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop      Metric LocPrf Weight Path
*  192.168.0.0/23   10.0.0.13                        0 2002 1000 i
*>                  10.0.0.9                         0 2001 1000 i
```

```
*  192.168.2.0      10.0.0.13                      0 2002 2001 i
*>                  10.0.0.9           0           0 2001 i
*  192.168.3.0      10.0.0.9                       0 2001 2002 i
*>                  10.0.0.13          0           0 2002 i
*> 192.168.4.0/23   0.0.0.0            0       32768 i

Total number of prefixes 4
```

Listing 8.7: traceroute from r_1000 to r_3000

```
Link between AS2001 and AS2002 down
 1  10.0.0.2 (10.0.0.2)  1 ms  1 ms  0 ms
 2  192.168.4.1 (192.168.4.1)  1 ms  1 ms  1 ms

 1  10.0.0.6 (10.0.0.6)  1 ms  1 ms  0 ms
 2  192.168.5.1 (192.168.5.1)  1 ms  1 ms  1 ms

Link between AS2001 and AS2002 up

 1  10.0.0.2 (10.0.0.2)  1 ms  1 ms  0 ms
 2  192.168.4.1 (192.168.4.1)  1 ms  1 ms  1 ms

 1  10.0.0.6 (10.0.0.6)  1 ms  1 ms  0 ms
 2  192.168.5.1 (192.168.5.1)  1 ms  1 ms  1 ms
```

Listing 8.8: PaRArch configuration for AS 1000

```
hostname AS1000
password zebra
!
router bgp 1000
 bgp router-id 192.168.1.1
 network 192.168.0.0/24
 network 192.168.1.0/24
 neighbor 10.0.0.10 remote-as 3000
 neighbor 10.0.0.10 port 2609
 neighbor 10.0.0.10 ebgp-multihop 255
 neighbor 10.0.0.10 update-source eth0
 neighbor 10.0.0.10 route-map upper out
 neighbor 10.0.0.14 remote-as 3000
 neighbor 10.0.0.14 port 2609
 neighbor 10.0.0.14 ebgp-multihop 255
 neighbor 10.0.0.14 update-source eth1
 neighbor 10.0.0.14 route-map lower out
!
access-list 1 permit 192.168.0.0 0.0.0.255
access-list 2 permit 192.168.1.0 0.0.0.255
!
route-map upper permit 10
 match ip address 2
 set as-path prepend 1000
!
route-map upper permit 20
!
route-map lower permit 10
 match ip address 1
 set as-path prepend 1000
!
route-map lower permit 20
!
```

8.4 Benefits of PaRArch

In this section, I describe the three main benefits of PaRArch, namely (i) it simplifies route management, thereby reducing the Operational Expenses (OPEX) of ISPs (ii) it reduces the size requirements for the FIB, which is one of the most expensive components in a router, and (iii) it increases the overall stability of the Internet.

8.4.1 Simplifications in route management

Currently, route management in the Internet involves three players under normal circumstances. (1) The leaf AS that advertises a set of routes covering the address space the Internet Routing Registry (IRR) has delegated it, (2) the first upstream provider AS that may apply a first filtering, and (3) the rest of the ASes that receive the routes. In this context, route management actions are mainly performed by the AS originating the prefix and its first upstream provider. The leaf AS has to advertise the routes complying to the address space delegations and restrictions imposed by RIR policies and the first upstream provider normally applies some filters in order to prevent buggy

advertisements corrupting its routing tables. These filters have to take RIR policies into account as well.

With PaRArch, the leaf AS advertises its best aggregations to the global routing plane. Filtering in this plane is restricted to the best aggregations. All prefixes used for traffic control prefixes are advertised in the PaRArch routing plane and will not progress further than the first provider AS into the Internet. This implies that route filtering is simplified for the first provider AS. The burden for the leaf AS is approximately the same. It will continue advertising the same amount of prefixes, although distributed between the global and the PaRArch routing plane.

When a misconfiguration occurs in an AS in the Internet and traffic from the leaf AS is diverted, the current situation is more complex and the root-cause analysis phase might take some time. We can distinguish two situations:

(1) The prefix that is involved in the incident is not being used by the leaf AS that "owns" the address space where it lies.
(2) The leaf AS is using the same prefix that is involved in the incident either for TE purposes or to prevent attacks by other ASes.

In the first case, the leaf AS can ask the provider AS where the wrong advertisement originated and ask the AS that is responsible for the incident to withdraw it. In the second case, the provider AS will not consolidate the wrong prefix: the AS_PATH attribute will be shorter for the leaf AS than for the attacking AS. The attacked AS will have to start asking other ASes that have experienced a route change for assistance. An example for this is the YouTube hijacking incident [120].

With PaRArch, the overall situation is equivalent to the first case and the provider AS can assist the leaf AS in restoring lawful routing.

8.4.2 Reduction of the FIB size

Additionally, this architecture will allow a significant reduction of the amount of routes in the global routing tables. This has a direct impact on the design of core routers: the RIBs are smaller and consume less memory. Additionally, any FIB size reduction algorithm implemented by the router will have to process smaller RIBs and require less resources to achieve the same result.

Reversing the current trend towards fragmentation of the Default Free Zone should not have any impact on the ability of transit ISPs to implement traffic balancing based on their clients' advertisements. Upstream ISPs are expected to have enough clients to arrive at a near optimum traffic distribution using their best aggregations. This is also consistent with the behaviour of intermediate ASes shown in Section 7.1.3.

8.4.3 Overall Stability and Robustness

PaRArch uses parallel BGP-4 sessions. This provides the best isolation between the different routing tables. In case of a routing storm due to implementation deficiencies in one of the routing daemons, the other session will remain unaffected. Since the Internet

DFZ routing table is handled by a plain and unmodified BGP-4 routing daemon, implementation deficiencies are less likely to appear here. This behaviour is more desirable than emulating an MP-BGP session managing several routing tables and other proposals brought forward to the IETF to provide multi-protocol support and provides a better isolation between the different Address Families transported by an MP-BGP session. In my proof of concept setup, the "Internet" routing table information is managed by a BGP-4 daemon using the standard ports, while the *local* routing information is managed by a BGP-4 daemon using non-standard ports. Both daemons have independent RIBs stored in different Linux kernel routing tables. These routing tables have no means of interaction. PaRArch intentionally does not include any route redistribution mechanism between the DFZ routing table and the local routing table to avoid cross-pollution.

Additionally, PaRArch does not need to deploy any routing policies in the main Internet routing table. This implies that the Internet DFZ routing plane would cease to suffer from negative effects of routing policies on BGP-4 [156, 71] in the case of global adoption across the whole Internet.

8.5 Conclusion

In this chapter I have shown how introducing some modifications into the current Internet routing architecture introduces better route management and a more robust routing infrastructure. It has lower resource consumption with regards to routing table management and is less prone to routing storms.

In the next and final chapter, I give an outlook on future work on this architecture. The certain need for IPv6 and the uncertainties that still surround the migration process open opportunities for this work to be continued and possibly be deployed in the Internet.

Chapter 9

Conclusions and future research

Contents

9.1 Conclusions

BGP-4 is the routing protocol between Autonomous Systems in the Internet and at the same time it is also used for traffic engineering purposes. Traffic engineering is implemented by means of policies. These policies make BGP-4 yield unintended results.

In my work, I have started by showing that routing policies are used by a majority of the ASes at the edges of the Internet. I have proposed and implemented an algorithm to study different phenomena related with traffic engineering. Using this algorithm, I have looked at situations where the network was oscillating. I could show with BGP-4 traffic samples that providers try to route "around" oscillations and how they do it. In order to better understand the behaviour of different providers, I propose to use the empirical Cumulative Distribution Function of the arrival times of different BGP-4 packets at the RIPE's Routing Repository. Then I showed different instances of instability in the network protocol that can be traced back to ambiguities in the definition of extensions to the BGP-4 protocol. In this case, I show how a quick fix revealed the internal structure of an AS Confederation and how a distant AS was affected by a routing storm.

These precedents led me to introduce PaRArch, a new routing architecture for the Internet that overcomes the problems detected by separating the main routing plane of the Internet from the local traffic engineering configurations. I have implemented this architecture on the open-sourced routing daemon Quagga. To test my modified version of this daemon, I propose use cases where traffic engineering is involved. I demonstrate that reachibility is never lost and that the routing policies work as intended and compare

it with current practises using BGP-4 only, where side effects produce routing results that are not intended. PaRArch is backwards compatible and can be incrementally deployed and should provide a much stabler Internet. This factor is another advantage in the current situation, where the migration to IPv6 is in its starting phase and a stable and reliable IPv4 base network that does not to interfere with the new IPv6 deployments is needed.

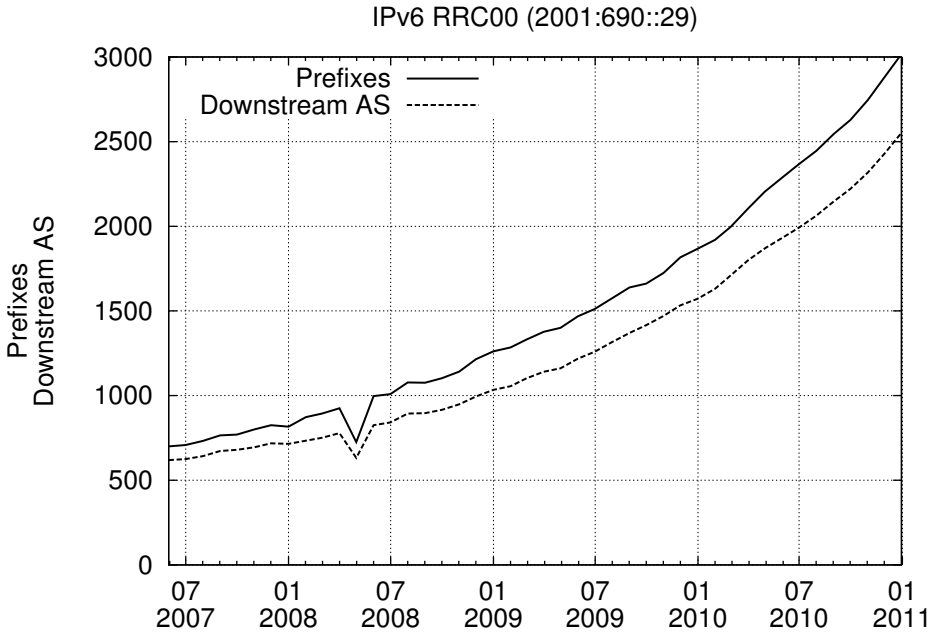
9.2 Using PaRArch in the new IPv6 Internet

One of the current technical challenges for the Internet community is the address space depletion and success of the migration to IPv6. The RIPE RR has consistent information regarding the evolution of the IPv6 DFZ and its routing tables only since 2007.

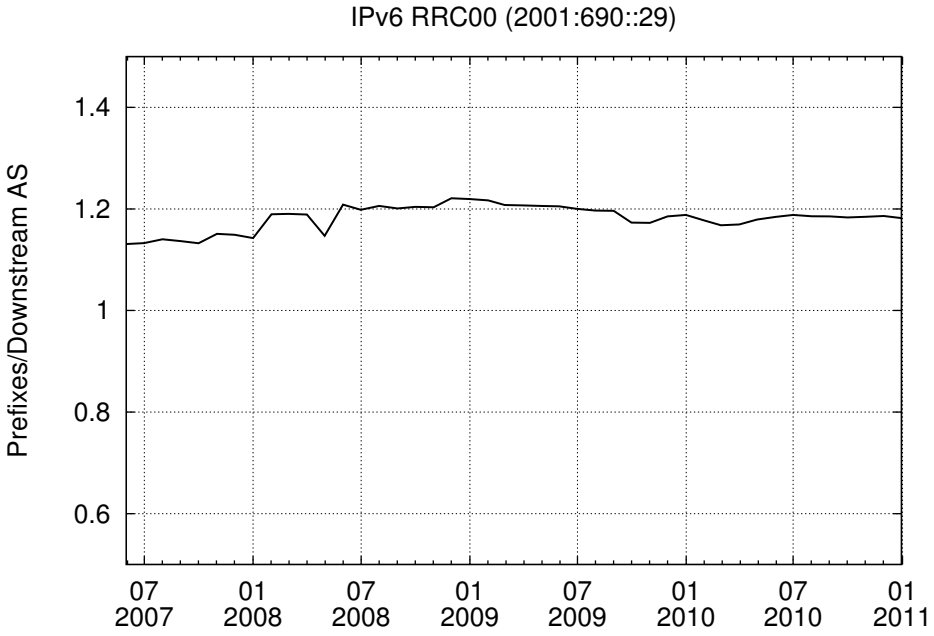
Figure 9.1a shows evolution of the IPv6 routing table size in the Internet's DFZ, the number of ASes and the average number of prefixes advertised by one AS. It is worth noting that this number is greater than 1, which indicates that some ASes are advertising more than one prefix. Whether these ASes have more than one prefix assigned or this evolution reveals the first signs of disaggregation in the IPv6 Internet remains to be studied.

Adopting the proposed architecture at this point in time also has benefits for the transition to an IPv6-based Internet and during its lifetime. It would allow the reopening of the problem of AS-level multi-homing in the IPv6 Internet, which is currently closed at the IETF level and only partially resolved by the RIRs. As far as the IETF is concerned, RFC 4177 [195] presents the architectural elements of multi-homing in IPv6, concentrates on site-level multi-homing configurations and favours the hierarchical properties of the IPv6 addressing allocation procedure over well established and documented Autonomous System level multi-homing techniques used in today's IPv4 Internet. Das provides a broader discussion of multi-homing in IPv6 and notes that one of the major discouraging facts for reusing the multi-homing techniques of the IPv4 Internet in the IPv6 Internet is routing table size growth [196]. This argument is understandable in the light of Figure 7.7.

In 2009, RIPE and other RIRs have opened the door to IPv6 multi-homing by introducing the PI addressing space in their IPv6 Allocation Policies [197]. While Provider Independent (PI) addressing space is essential for multi-homed ASes, it imposes important restrictions to who can access it, how it is handled and how much space is assigned. There are, however, no restrictions imposed on advertisement practises. Although it recommends that fragmentation should be avoided, it does not specifically prohibit it. The architecture proposed in this paper could be useful in bringing together the TE needs of multi-homed ASes and a controlled growth of the IPv6 routing table. This philosophy has been maintained in the last revision of the RIPE IPv6 allocation policy document [172] in 2011.



(a) Number of prefixes and leaf ASes



(b) Prefixes advertised per leaf AS

Figure 9.1: Evolution of the IPv6 DFZ

9.3 Future work

In addition to adapting PaRArch to IPv6 scenarios, there are other aspects of this work that will be continued in the future. The current version of the MRT binary format parsing tools is prepared for IPv4 and IPv6. However, since the work I have presented here is more related to IPv4 (e.g., examining the evolution of the routing tables, examining 4-byte ASN violations and IPv4 prefix update sequences) than to IPv6, most of the analysis programs rely on the IPv4 routines, which are hence better tested. IPv6 support needs to be further tested. IPv6 routing table analysis is a task for the near future and will provide the necessary test cases with real-world data.

Another aspect of Internet dynamics that needs to be more investigated is the characterisation of ISP behaviour with regards to the time slots used to perform routing infrastructure maintenance in a broader sense. This includes investigating when and how an ISP modifies the attributes of the prefixes it advertises. This is an important factor for the possible success of the proposed architecture. An architecture isolating TE from the Internet will makes sense, whether ISPs limit or not their operation time window to minimise the impact of TE operations on the Internet. However, by isolating TE, the pressure to constrain operations to a given time window disappears. In general, studies comparing the early adoption phases of IPv4 and IPv6 have been recently published [133] and the tools I have developed for this work can be also used in the future to conduct these and related studies.

Another recent evolution of the MRT binary format that needs to be explored is the possibility to store geo-positioning data of the collecting devices as proposed in RFC 6397 [198]. Geo-positioning data in conjunction with the time information will give an extra level of reassurance to the operations time-window detection procedure described in Section 5.3. Currently, BGP-4 traffic is stored in RIPE's Routing Repository are stored using the timestamp of the arrival time. Geo-positioning will allow to translate these timestamps to the local time of the collecting devices. This will allow identify the moment of the day – morning, afternoon, etc. – when the events where actually generated.

Regarding the IPv4 proposal for PaRArch, the main future work is to transfer it to the industry. In order to do so, some of the concepts will need to go through the IETF standardisation process. This is a lengthy process that just has been started.

Bibliography

[1] J. Postel, "Internet Protocol." RFC 791 (Standard), Sept. 1981. Updated by RFC 1349.

[2] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271 (Draft Standard), Jan. 2006. Updated by RFCs 6286, 6608.

[3] RIPE, "RIPE Routing Information Service." `http://www.ripe.net/data-tools/stats/ris/routing-information-service`, oct 1999. Last visit, 20-Nov-2011.

[4] A. Antony and H. Uijterwaaly, "Routing Information Service," Design note RIPE-200, RIPE, October 1999.

[5] J. Cowie, "Staring Into The Gorge: Router Exploits." `http://www.renesys.com/blog/2009/08/staring-into-the-gorge.shtml`, August 2009. Last visit, 12-Oct-2009.

[6] Watanabe, "AS_CONFED_SEQUENCE in AS4_PATH attribute." `http://irs.ietf.to/past/docs_20090218/IRS19_watanabe_AS_CENFED_SEQ_in_AS4PATH.pdf`, February 2009. Last visit, 18-Aug-2009.

[7] P. Aranda Gutiérrez, "Detection of Trial and Error Traffic Engineering with BGP-4," in *The Fifth International Conference on Networking and Services; ICNS 2009*, IARIA, April 2009.

[8] P. Aranda-Gutiérrez, "On the use of Trial and Error Traffic Engineering techniques in the Internet," in *Sixth International Conference on Broadband Communications, Networks and Systems; Broadnets 2009*, ICST, September 2009.

[9] P. Aranda Gutiérrez, "Simple Statistical Analysis Method for the Behaviour of Autonomous Systems," in *The Sixth International Conference on Networking and Services;ICNS 2010*, ICST, May 2010.

[10] P. Aranda Gutiérrez, "Using RFC4893 violations to reveal the topology of AS Confederations," in *The Sixth International Conference on Networking and Services;ICNS 2010*, ICST, May 2010.

[11] P. Aranda Gutiérrez, "Revisiting the Impact of Traffic Engineering Techniques on the Internet's Routing Table," in *MonAMI'10*, Sep 2010.

[12] P. Aranda Gutiérrez, "A simplified internet routing architecture: Removing traffic engineering and security artifacts from the internet's default free zone," in Pentikousis *et al.* [199], pp. 433–445.

[13] P. Aranda Gutiérrez, "Flexible Routing with Maximum Aggregation in the Internet," in *MonAMI'11*, Sep 2011.

[14] J. Hawkinson and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)." RFC 1930 (Best Current Practice), Mar. 1996.

[15] K. Hubbard, M. Kosters, D. Conrad, D. Karrenberg, and J. Postel, "Internet Registry IP Allocation Guidelines." RFC 2050 (Best Current Practice), Nov. 1996.

[16] "Policies for IPv4 address space management in the Asia Pacific region: 3.- Delegations from the APNIC IPv4 address pool." `http://www.apnic.net/policy/add-manage-policy#9.10`, May 2011. Last visit, 11-May-2011.

[17] G. Huston, "The 16-bit AS Number Report." `http://www.potaroo.net/tools/asn16/`, nov 2011. Last visit, 20-Nov-2011.

[18] Q. Vohra and E. Chen, "BGP Support for Four-octet AS Number Space." RFC 4893 (Proposed Standard), May 2007.

[19] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets." RFC 1918 (Best Current Practice), Feb. 1996.

[20] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses." RFC 3927 (Proposed Standard), May 2005.

[21] M. Cotton and L. Vegoda, "Special Use IPv4 Addresses." RFC 5735 (Best Current Practice), Jan. 2010. Updated by RFC 6598.

[22] R. Hinden and B. Haberman, "Unique Local IPv6 Unicast Addresses." RFC 4193 (Proposed Standard), Oct. 2005.

[23] H. Gredler and W. Goralski, *The Complete IS-IS Routing Protocol*. Computer Science, Springer London, 2005. ISBN 978-1-85233-822-0.

[24] J. Moy, "OSPF Version 2." RFC 2328 (Standard), Apr. 1998. Updated by RFCs 5709, 6549.

[25] T. Li, H. Smit, and T. Przygienda, "Domain-Wide Prefix Distribution with Two-Level IS-IS." RFC 5302 (Proposed Standard), Oct. 2008.

[26] P. Peymani and M. Kolon, *Juniper Networks Router Security; Best Common Practices for Hardening the Infrastructure*. Juniper Networks, Inc., May 2002.

[27] A. Ebalard, "IPv6 Type 0 Routing Header," *IETF Journal*, vol. 3, no. 2, 2007.

[28] P. Biondi and A. Ebalard, "IPv6 Type 0 Routing Header Security," in *CanSecWest 2007*, 2007.

[29] A. Reitzel, "Deprecation of Source Routing Options in IPv4," Internet-Draft draft-reitzel-ipv4-source-routing-is-evil-00, Internet Engineering Task Force, Sept. 2007. Expired.

[30] J. Abley, "Deprecation of Type 0 Routing Headers in IPv6," Internet-Draft draft-jabley-ipv6-rh0-is-evil-00, Internet Engineering Task Force, May 2007. Expired.

[31] J. Abley, "Deprecation of Type 0 Routing Headers in IPv6," Internet-Draft draft-ietf-ipv6-deprecate-rh0-01, Internet Engineering Task Force, June 2007. Now RFC 5095.

[32] M. Gibson, "Achieving Assured Service Levels through Source Routed MPLS," Internet-Draft draft-gibson-mpls-srcroute-00, Internet Engineering Task Force, Feb. 2001. Expired.

[33] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4." RFC 4728 (Experimental), Feb. 2007.

[34] T. Bates, E. Chen, and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)." RFC 4456 (Draft Standard), Apr. 2006.

[35] Cisco Press, *BGP Best Path Selection Algorithm*, may 2006. Last visit: 20-Nov-2011.

[36] S. Halabi, *Internet Routing Architectures*. Cisco Press, 2nd edition ed., 2000.

[37] D. Thaler and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection." RFC 2991 (Informational), Nov. 2000.

[38] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm." RFC 2992 (Informational), Nov. 2000.

[39] O. Bonaventure, "Interdomain routing with BGP4; Part 4/4." `http://www.info.ucl.ac.be/~obo/pres/BGP-2003/pdf/BGP-4.sxi.pdf`, May 2003. Last visit, 23-Dec-2009.

[40] Cisco Systems Inc., *Interworking technology handbook: BGP-4*, dec 2009. Last visit: 20-Nov-2011.

[41] Juniper Networks, *Examine BGP Routes and Route Selection in Juniper routers*, 2009. Last visit, 20-Nov-2011.

[42] W. J. Goralski, ed., *Juniper and Cisco Routing: Policy and Protocols for Multi-vendor IP Networks*. Wiley, 2004.

[43] T. Li and Y. Rekhter, "A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)." RFC 2430 (Informational), Oct. 1998.

[44] J. Babiarz, K. Chan, and F. Baker, "Configuration Guidelines for DiffServ Service Classes." RFC 4594 (Informational), Aug. 2006. Updated by RFC 5865.

[45] G. Huston, "Interconnection, Peering and Settlements-Part I." `http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_2-1/peering_and_settlements.html`, Mar 1999.

[46] G. Huston, "Interconnection, Peering and Settlements-Part II." `http://www.cisco.com/web/about/ac123/ac147/ac174/ac200/about_cisco_ipj_archive_article09186a00800c8900.html`, Jun 1999.

[47] R. van der Berg, "How the Net works: an introduction to peering and transit." `http://arstechnica.com/old/content/2008/09/peering-and-transit.ars`, Sep 2008. Last visit: 5-Jul-2010.

[48] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, 2001.

[49] "BGP Looking Glasses for IPv4/IPv6, Traceroute & BGP Route Servers." `http://www.bgp4.as/looking-glasses`, 2002. Last visit 20-Nov-2011.

[50] ATT, "AT&T Managed Internet Service (MIS)." `http://new.serviceguide.att.com/mis.htm/`, 2007. Last visit, 05-Oct-2009.

[51] NTT, "NTT Communications Global IP Network - Service Level Agreements." `http://www.us.ntt.net/support/sla/`, 2007. Last visit, 05-Oct-2009.

[52] W. B. Norton, "Tier 1 ISP." `http://drpeering.net/white-papers/Ecosystems/Tier-1-ISP.html`, 2010. Last visit: 20-Nov-2011.

[53] "Free Online Dictionary Of Computing: upstream." `http://foldoc.org/upstream`, May 1999. Last visit, 05-Jul-2010.

[54] T. G. Griffin and G. Wilfong, "An analysis of bgp convergence properties," in *In Proc. of SIGCOMM'99*, pp. 277–288, ACM Press, 1999.

[55] S. Uhlig and B. Quoitin, "Tweak-it: BGP-Based Interdomain Traffic Engineering for Transit ASes," in *Proc. Next Gen. Internet Networks*, pp. 75–82, 2005.

[56] S. Uhlig, "A Multiple-objectives Evolutionary Perspective to Interdomain Traffic Engineering in the Internet," in *Workshop on Nature Inspired Approaches to Networks and Telecommunications*, 2004.

[57] S. Uhlig, O. Bonaventure, and B. Quoitin, "Interdomain Traffic Engineering with minimal BGP configurations," 2003.

[58] B. Wire, "Internap Closes netVmg Acquisition; Announces Definitive Agreement to Acquire Second Internet Route Control Company, Sockeye Networks.." `http://www.highbeam.com/doc/1G1-108539181.html`, Oct 2003. Last visit, 01-Sep-2010.

[59] C. Panigl, J. Schmitz, P. Smith, and C. Vistoli, "Recommendations for Coordinated Route-flap Damping Parameters," RIPE Routing-WG Document ripe-228, RIPE, October 2001.

[60] E. Chen, "Route Refresh Capability for BGP-4." RFC 2918 (Proposed Standard), Sept. 2000.

[61] Cisco Press, *BGP Soft Reset Enhancement*, jan 2003. Last visit: 20-Nov-2011.

[62] S. Halabi, "BGP4 Case Studies/Tutorial." `http://www.ittc.ku.edu/EECS/EECS_800.ira/bgp_tutorial/`, January 1996.

[63] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute." RFC 1997 (Proposed Standard), Aug. 1996.

[64] B. Donnet and O. Bonaventure, "On BGP communities," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 55–59, 2008.

[65] B. Quoitin and O. Bonaventure, "A survey of the utilization of the BGP community attribute," Internet-Draft draft-quoitin-bgp-comm-survey-00, Internet Engineering Task Force, Mar. 2002. Expired.

[66] B. Quoitin, S. Uhlig, and O. Bonaventure, "Using Redistribution Communities for Interdomain Traffic Engineering," in *QofIS*, pp. 125–134, 2002.

[67] B. Quoitin, S. Tandel, S. Uhlig, and O. Bonaventure, "Interdomain traffic engineering with redistribution communities," *Comput. Commun.*, vol. 27, no. 4, pp. 355–363, 2004.

[68] G. Huston, "POTAROO Web site." `http://www.potaroo.net/`, oct 2011. Last visit, 20-Nov-2011.

[69] Barry Raveendran Green and Phillip Smith, *CISCO - ISP Essentials*. Cisco Press, 9 2002.

[70] S. Uhlig and O. Bonaventure, "Designing BGP-based outbound traffic engineering techniques for stub ASes," *Comput. Commun. Rev.*, vol. 34, 2004.

[71] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The Stable Paths Problem and Interdomain Routing," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 232–243, 2002.

[72] D. Obradovic, "Real-time model and convergence time of bgp," in *Proceedings of IEEE Infocom*, 2002.

[73] M. Roughan, J. Li, R. Bush, Mao, and T. Griffin, "Is BGP Update Storm a Sign of Trouble: Observing the Internet Control and Data Planes During Internet Worms," in *Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Calgary, Canada*, July 2006.

[74] M. Suchara, A. Fabrikant, and J. Rexford, *BGP safety with spurious updates*, pp. 2966–2974. Proceedings IEEE INFOCOM, IEEE, 2011.

[75] R. Kuhn, K. Sriram, and D. Montgomery, "Border Gateway Protocol Security," Recommendations of the National Institute of Standards and Technology Special Publication 800-54, NIST, Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930, 2007.

[76] "The Oregon RouteViews Project." `http://www.routeviews.org`, 2001.

[77] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1." RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785, 6266, 6585.

[78] "The Multi-threaded Routing Toolkit MRT." `http://www.mrtd.net`, 2003. Discontinued, domain did not exist the 20-Nov-2011.

[79] L. Blunk, M. Karir, and C. Labovitz, "Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format." RFC 6396 (Proposed Standard), Oct. 2011.

[80] RIPE, "RIS Raw Data." `http://www.ripe.net/data-tools/stats/ris/ris-raw-data`, 1999. Last visit, 20-Nov-2011.

[81] H. Kong, "Consistency Verification of Zebra BGP Data Collection." Technical report @ RIPE-46, September 2003.

[82] K. Ishiguro, "GNU Zebra." `http://www.zebra.org`, 2003. Last visit, 09-Apr-2011.

[83] "Quagga Routing Suite." `http://www.quagga.net`, Dec 2009. Last visit, 09-Apr-2011.

[84] P. Cheng, X. Zhao, B. Zhang, and L. Zhang, "Longitudinal study of BGP monitor session failures," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 34–42, April 2010.

[85] P. Cheng and X. Zhao, "BGP Reset: Records of BGP Monitoring Session Failures." `http://bgpreset.cs.arizona.edu`, 2009. Last visit: 02-May-2011.

[86] P. Smith, "Aggregation (?) Effect of business practices on the Internet today." `www.sanog.org/resources/sanog6/pfs-routing-aggregration.pdf`, July 2005. SANOG 6. Thimphu, Bhutan. 16 – 23 Jul 2005.

[87] K. Fall, P. B. Godfrey, G. Iannaccone, and S. Ratnasamy, "Routing Tables: Is Smaller Really Much Better?," in *Eighth ACM Workshop on Hot Topics in Networks*, ACM, oct 2009.

[88] R. Draves, C. King, S. Venkatachary, and B. D. Zill, "Constructing Optimal IP Routing Tables," in *In Proc. IEEE INFOCOM*, pp. 88–97, 1999.

[89] H. Ballani, P. Francis, and T. Cao, "ViAggre: Making Routers Last Longer!," in *Seventh ACM Workshop on Hot Topics in Networks*, ACM, nov 2008.

[90] S. Suri, T. Sandholm, T. S, and P. Warkhede, "Compressing Two-Dimensional Routing Tables," *Algorithmica*, vol. 25, pp. 287–300, 2003.

[91] Z. A. Uzmi, S. Jawad, A. Tariq, and P. Francis, "IP Prefix Aggregation with SMALTA," tech. rep., Max Planck Institute for Software Systems, Jul 2010. Last visit 30-Apr-2011.

[92] Z. Uzmi, A. Tariq, and P. Francis, "FIB Aggregation with SMALTA," Internet-Draft draft-uzmi-smalta-01, Internet Engineering Task Force, Jan. 2011. Expired.

[93] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. V. D. Merwe, "The Case for Separating Routing from Routers," in *In ACM SIGCOMM Workshop on Future Directions in Network Architecture*, ACM Press, 2004.

[94] A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture." RFC 4655 (Informational), Aug. 2006.

[95] R. Zhang and J.-P. Vasseur, "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements." RFC 4216 (Informational), Nov. 2005.

[96] D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2." RFC 3630 (Proposed Standard), Sept. 2003. Updated by RFCs 4203, 5786.

[97] T. Li and H. Smit, "IS-IS Extensions for Traffic Engineering." RFC 5305 (Proposed Standard), Oct. 2008. Updated by RFC 5307.

[98] R. Cohen and A. Shochot, "The Global-ISP Paradigm," *Computer Networks*, vol. 51, pp. 1908 – 1921, June 2007.

[99] X. Zhang, P. Francis, J. Wang, and K. Yoshida, "Scaling IP Routing with the Core Router-Integrated Overlay," in *Proceedings of the ICNP*, 2006.

[100] P. Francis, X. Xu, H. Ballani, D. Jen, R. Raszuk, and L. Zhang, "FIB Suppression with Virtual Aggregation," Internet-Draft draft-ietf-grow-va-05, Internet Engineering Task Force, June 2011. Work in progress.

[101] P. Francis, X. Xu, H. Ballani, D. Jen, R. Raszuk, and L. Zhang, "Auto-Configuration in Virtual Aggregation," Internet-Draft draft-ietf-grow-va-auto-04, Internet Engineering Task Force, June 2011. Work in progress.

[102] H. Ballani, P. Francis, D. Jen, X. Xu, and L. Zhang, "Performance of Virtual Aggregation," Internet-Draft draft-ietf-grow-va-perf-00, Internet Engineering Task Force, July 2009. Expired.

[103] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic Routing Encapsulation (GRE)." RFC 2784 (Proposed Standard), Mar. 2000. Updated by RFC 2890.

[104] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)." RFC 4364 (Proposed Standard), Feb. 2006. Updated by RFCs 4577, 4684, 5462.

[105] X. Yang, "NIRA: A New Internet Routing Architecture," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Internet measurment* (ACM, ed.), aug 2003.

[106] B. Zhang, V. Kambhampati, D. Massey, R. Oliveira, D. Pei, L. Wang, and L. Zhang, "Internet Routing: Separating Customers from Providers," sep 2006.

[107] J. Postel, "Domain Name System Structure and Delegation." RFC 1591 (Informational), Mar. 1994.

[108] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core," in *Seventh ACM Workshop on Hot Topics in Networks*, ACM, nov 2008.

[109] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP)," Internet-Draft draft-farinacci-lisp-12, Internet Engineering Task Force, Mar. 2009. Expired.

[110] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture." RFC 4423 (Informational), May 2006.

[111] D. Meyer and D. Lewis, "Architectural Implications of Locator/ID Separation," Internet-Draft draft-meyer-loc-id-implications-01, Internet Engineering Task Force, Jan. 2009. Expired.

[112] S. Uhlig and O. Bonaventure, "IST Project ATRIUM- Report I4.2 Analysis of Interdomain Traffic," tech. rep., ATRIUM Project, 2001.

[113] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig, "A performance evaluation of BGP-based traffic engineering," *Intl. Journal of Network Management*, vol. 15, pp. 177–191, 2004.

[114] "Internap Internet Access." Internet Site, 2009. Last visit, 20-Nov-2011.

[115] "Internap Internet Access, Improved Network Performance through Optimization - Flow Control Platform Components." Internet Site, 2010. Last visit, 20-Nov-2011.

[116] "Review: RouteScience's PathControl." `http://www.networkworld.com/reviews/2002/0415rev.html`, Apr 2002. Last visit: 01-Sep-2010.

[117] J. Seedorf and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement." RFC 5693 (Informational), Oct. 2009.

[118] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez, "Deep diving into bittorrent locality," in *Co-Next Student Workshop '09: Proceedings of the 5th international student workshop on Emerging networking experiments and technologies*, (New York, NY, USA), pp. 7–8, ACM, 2009.

[119] C.-C. Lin, M. Caesar, and K. van der Merwe, "Toward interactive debugging for isp networks," in *Eighth ACM Workshop on Hot Topics in Networks*, ACM, oct 2009.

[120] "YouTube Hijacking: A RIPE NCC RIS case study." http://www.ripe.net/news/study-youtube-hijacking.html, mar 2008.

[121] The IST Intersection Consortium, "INTERSECTION (INfrastructure for heTErogeneous, Resilient, SEcure, Complex, Tightly Inter-Operating Networks)." `http://www.intersection-project.eu/`, Jan 2008. Last visit 25-Jun-2010.

[122] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, " PHAS: A Prefix Hijack Alert System ," 2006.

[123] N. Feamster, H. Balakrishnan, and J. Rexford, "Some Foundational Problems in Interdomain Routing," in *Third ACM Workshop on Hot Topics in Networks*, ACM, nov 2004.

[124] P. Traina, D. McPherson, and J. Scudder, "Autonomous System Confederations for BGP." RFC 3065 (Proposed Standard), Feb. 2001. Obsoleted by RFC 5065.

[125] Q. Vohra and E. Chen, "BGP Support for Four-octet AS Number Space." RFC 4893bis (Internet draft), Oct. 2009.

[126] J. Scudder, E. Chen, P. Mohapatra, and K. Patel, "Revised Error Handling for BGP UPDATE Messages," Internet-Draft draft-ietf-idr-optional-transitive-04, Internet Engineering Task Force, Nov. 2011. Work in progress, replaced by draft-ietf-idr-error-handling.

[127] G. Nalawade, "BGPv4 Soft-Notification Message," Internet-Draft draft-nalawade-bgp-soft-notify-01, Internet Engineering Task Force, July 2005. Expired.

[128] J. Scudder, C. Appanna, and I. Varlashkin, "Multisession BGP," Internet-Draft draft-ietf-idr-bgp-multisession-06, Internet Engineering Task Force, Mar. 2011. Expired.

[129] D. Dolev, S. Jamin, O. Mokrync, and Y. Shavitt, "Internet resiliency to attacks and failures under BGP policy routing," *Computer Networks*, vol. 50, no. 16, pp. 3183–3196, 2006.

[130] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, "Public Review for An Internet Routing Forensics Framework for Discovering Rules of Abnormal BGP Events," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 55–66, 2005.

[131] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz, "BGP Routing dynamics revisited," in *SIGCOMM Computer Communication Review*, vol. Volume 37 Issue 2, ACM, 2007.

[132] Y. Huang, N. Feamster, A. Lakhina, and J. Xu, "Diagnosing network disruptions with network-wide analysis," *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS 07*, vol. 35, no. 1, p. 61, 2007.

[133] G. Zhang, B. Quoitin, and S. Zhou, "Phase changes in the evolution of the ipv4 and ipv6 as-level internet topologies," *Computer Communications*, vol. 34, no. 5, pp. 649–657, 2011.

[134] R. Khosla, S. Fahmy, and Y. C. Hu, *BGP molecules: Understanding and predicting prefix failures*, pp. 146–150. IEEE, 2011.

[135] "The Internet Intelligence Authority." `http://www.renesys.com`, 2000.

[136] E. Zmijewski, "Reckless Driving on the Internet." `http://www.renesys.com/blog/2009/02/the-flap-heard-around-the-worl.shtml`, Feb 2009.

[137] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, (Boston, MA), pp. 255–270, USENIX Association, Dec. 2002.

[138] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, "German-Lab Experimental Facility," in 3rd *Future Internet Symposium 2010*, (Berlin, Germany), Sept. 2010.

[139] Università Roma Tre; Computer Network Laboratory, "Netkit: The poor man's system to experiment computer networking." http://wiki.netkit.org/index.php/Main_Page, Dec 2009.

[140] J. Dike, "The User-mode Linux Kernel Home Page." http://user-mode-linux.sourceforge.net/, mar 2008. Last visit, 20-Nov-2011.

[141] F. Bellard, "QEMU: Open Source Processor Emulator." http://wiki.qemu.org/Main_Page, nov 2011. Last visit, 20-Nov-2011.

[142] Oracle, "VirtualBox." https://www.virtualbox.org/, 2007. Last visit, 20-Nov-2011.

[143] C. Fillot, "Cisco 7200 Simulator." http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator, aug 2007. Last visit, 20-Nov-2011.

[144] G. Malkin, "RIP Version 2." RFC 2453 (Standard), Nov. 1998. Updated by RFC 4822.

[145] K. Ishiguro, "The Zebra Protocol." http://www.quagga.net/docs/quagga.html#SEC148, jul 2006. Last visit, 20-Nov-2011.

[146] The OpenBSD Project, "OpenBGPD." http://www.openbgpd.org, November 2009.

[147] "eXtensible Open Router Platform." http://www.xorp.org/, 2010.

[148] Juniper Networks, Inc., *Junos 11.4 OS: CLI User Guide*, Nov 2011.

[149] "The Bird Internet Routing Daemon." http://bird.network.cz/, Jan 2012.

[150] "RIPE RIS - libbgpdump." http://www.ris.ripe.net/source/bgpdump, 2002. Last visit 20-Nov-2011.

[151] M. Rossi and G. Huston, "MRT dump file manipulation toolkit (MDFMT) - version 0.2." http://caia.swin.edu.au/reports/090730B/CAIA-TR-090730B.pdf, Jul 2009. Last Visit: 07-May-2011.

[152] Jon Oberheide, "pybgpdump." http://jon.oberheide.org/pybgpdump/, Jan 2007. Last visit 07-May-2011.

[153] W. R. Parkhurst, *Cisco BGP-4 Command and Configuration Handbook (CCIE Professional Development)*. Cisco Press, 2001.

[154] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On Selfish Routing in Internet-Like Environments," in *ACM/IEEE Transactions on Networking, selected papers from SIGCOMM 2004*, August 2006.

[155] R. R. Dakdouk, S. Salihoglu, H. Wang, H. Xie, and Y. R. Yang, "Interdomain Routing as Social Choice," in *Proceedings of Incentive-Based Computing (IBC), Lisboa, Portugal*, July 2006.

[156] T. Griffin and G. Huston, "BGP Wedgies." RFC 4264 (Informational), Nov. 2005.

[157] R. Bush, T. Griffin, Z. M. Mao, and R. B. (iij, "Route Flap Damping: Harmful?," 2002.

[158] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz, "Route flap damping exacerbates Internet routing convergence," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 221–233, ACM, 2002.

[159] P. Smith and C. Panigl, "Recommendations on Route-flap Damping," RIPE Routing Working Group Document ripe-378, RIPE, May 2006.

[160] C. I. T. Project, "TRAMMS IP Traffic report no 2," tech. rep., CELTIC, June 2008.

[161] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[162] P. Aranda Gutiérrez, "BGP-4 Protocol Patterns and Their Impact on QoS Behaviour," in *Inter-domain Performance and Simulation Workshop, Budapest*, 2004.

[163] "IBM WebSphere Telecom Web Services Server, Version 7.2." `http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp?topic=/com.ibm.twss.primitives.doc/sla_r.html`, 2007.

[164] "August 2011: Echovault 5.0 introduces real-time operational intelligence, sla templates and interactive reporting." `http://www.creanord.com/echonews/Real-Time-Operational-Intelligence-Interactive-Reporting.html`, aug 2011.

[165] L. Daigle, "WHOIS Protocol Specification." RFC 3912 (Draft Standard), Sept. 2004.

[166] Networks and More!, Inc., "Networks and More!, Inc.." `http://www.andmore.com`, 2007. Last visit, 20-Nov-2011.

[167] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4." RFC 4760 (Draft Standard), Jan. 2007.

[168] P. Marques and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing." RFC 2545 (Proposed Standard), Mar. 1999.

[169] P. Traina, "Autonomous System Confederations for BGP." RFC 1965 (Experimental), June 1996. Obsoleted by RFC 3065.

[170] P. Traina, D. McPherson, and J. Scudder, "Autonomous System Confederations for BGP." RFC 5065 (Draft Standard), Aug. 2007.

[171] "GNUPlot homepage." `http://www.gnuplot.info/`, mar 2011. Last visit, 20-Nov-2011.

[172] B. Carr, O. Sury, J. P. Martinez, A. Davidson, R. Evans, F. Yilmaz, and I. Wijte, "IPv6 Address Allocation and Assignment Policy," RIPE Address Policy Working Group Document ripe-512, RIPE, Feb 2011.

[173] APNIC, "Policies for IPv4 address space management in the Asia Pacific region," Tech. Rep. APNIC-124, APNIC, may 2011.

[174] "Iana ipv4 address space registry," 2011.

[175] T. Bates, P. Smith, and G. Huston, "The IPv4 CIDR report." `http://www.cidr-report.org/as2.0/`, 2011. Last visit: 02-May-2011.

[176] G. Huston, "The ipv6 cidr report." `http://www.cidr-report.org/v6/as2.0/`, 2011. Last visit: 02-May-2011.

[177] R. Bush, B. Carr, D. Karrenberg, N. O'Reilly, O. Sury, N. Titley, F. Yilmaz, and I. Wijte, "IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region," RIPE Address Policy Working Group Document ripe-492, RIPE, Feb 2010.

[178] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *SIGCOMM'10*, pp. 75–86, 2010.

[179] M. Daneman, "Global Crossing sold for 3 Billion Dollars to Level 3 Communications." `http://www.democratandchronicle.com/article/20110412/BUSINESS/104120311/0/PODCAST07/Global-Crossing-sold-3B-Level-3-Communications?odyssey=nav|head`, Apr 2011.

[180] R. Raszuk, R. Fernando, K. Patel, D. McPherson, and K. Kumaki, "Distribution of diverse BGP paths.," Internet-Draft draft-ietf-grow-diverse-bgp-path-dist-06, Internet Engineering Task Force, Nov. 2011. Work in progress.

[181] B. R. Greene and P. Smith, *Cisco ISP Essentials*. Cisco Press, 2002.

[182] J. Hou, "J-Sim Official." `http://sites.google.com/site/jsimofficial/`, 2005. Last visit, 20-Nov-2011.

[183] The 4WARD Consortium, "The FP7 4WARD Project." `http://www.4ward-project.eu/`, 2008. Last visit, 01-Nov-2011.

[184] P. Aranda-Gutiérrez, P. Pöyhönen, L. Izaguirre Gamir, and F. Huertas Ferrer, "Using bgp-4 to migrate to a future internet," in Pentikousis *et al.* [199], pp. 14–25.

[185] K. Fall and K. Varadhan, "The Network Simulator - ns-2." `http://isi.edu/nsnam/ns/`, jun 2010. Last visit, 20-Nov-2011.

[186] T. Henderson, S. R. S. Floyd, and G. Riley, "ns-3." `http://www.nsnam.org/`, nov 2011. Last visit, 20-Nov-2011.

[187] J. Grossmann, B. Marsili, X. Alt, and A. Eromenko, "Graphic Network Simulator." `http://www.gns3.net`, 2007. Last visit, 20-Nov-2011.

[188] M. A. Brown, "Guide to IP Layer Network Administration with Linux." `http://linux-ip.net/html/routing-tables.html`, 2007. Last visit, 16-Mar-2011.

[189] B. Hubert, "Linux Advanced Routing & Traffic Control HOWTO." `http://www.lartc.org/lartc.html`, may 2012.

[190] J. Salim, H. Khosravi, A. Kleen, and A. Kuznetsov, "Linux Netlink as an IP Services Protocol." RFC 3549 (Informational), July 2003.

[191] M. Pizzonia and M. Rimondini, "Netkit: easy emulation of complex networks on inexpensive hardware," in *TRIDENTCOM* (M. P. de Leon, ed.), p. 7, ICST, 2008.

[192] N. I. of Standards and Technology, "Expect." `http://www.nist.gov/el/msid/expect.cfm`, jan 2010.

[193] R. Steenbergen, "A Practical Guide to (Correctly) Troubleshooting with Traceroute." `http://www.nanog.org/meetings/nanog47/presentations/Sunday/RAS_Traceroute_N47_Sun.pdf`, oct 2009.

[194] T. Griffin and G. T. Wilfong, "Analysis of the med oscillation problem in bgp," in *ICNP*, pp. 90–99, IEEE Computer Society, 2002.

[195] G. Huston, "Architectural Approaches to Multi-homing for IPv6." RFC 4177 (Informational), Sept. 2005.

[196] K. Das, "IPv6 Multihoming." `http://ipv6.com/articles/general/IPv6_Multihoming.htm`, 2008. Last visit 23-Oct-2010.

[197] B. Carr, O. Sury, J. P. Martinez, A. Davidson, R. Evans, F. Yilmaz, and I. Wijte, "IPv6 Address Allocation and Assignment Policy," tech. rep., RIPE, Sept. 2009. Obsoleted by ripe-512.

[198] T. Manderson, "Multi-Threaded Routing Toolkit (MRT) Border Gateway Protocol (BGP) Routing Information Export Format with Geo-Location Extensions." RFC 6397 (Proposed Standard), Oct. 2011.

[199] K. Pentikousis, O. Blume, R. A. Calvo, and S. Papavassiliou, eds., *Mobile Networks and Management - Second International Conference, MONAMI 2010, Santander, Spain, September 22-24, 2010. Revised Selected Papers*, vol. 16 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, 2011.